MOLARIS: Version 9.15

Reference Manual

Release date: September 28, 2015

Contents

Chapter 1

Installation and quickstart

In this section we present a simple outline of MOLARIS, in order to make the user familiar with the execution procedure. First we describe the way MOLARIS should be installed in your system and then we present the first demo. All the "chemistry" and the methodology behind this first calculation will be explained in further chapters.

1.1 MOLARIS references

MOLARIS is a computer simulation package based on the POLARIS and ENZYMIX programs. A publication describing the new improvements and additions present in the new version of MOLARIS is in preparation. The recommended reference when reporting the use of the package MOLARIS is now in preparation (ref. 2 below), while ref. 1 below is appropriate for citing some of the developments within the program:

1. Microscopic and Semimicroscopic Calculations of Electrostatic Energies in Proteins by the POLARIS and ENZYMIX Programs, F. S. Lee, Z. T. Chu, and A. Warshel, J. Comp. Chem. 14,161 (1993).

2. MOLARIS version VERSION. Z. T. Chu, J. Villà, M. Strajbl, C. N. Schutz, A. Shurki, A. Warshel. University of Southern California.

1.2 Installing and running MOLARIS

1.2.1 Installation procedure

MOLARIS is distributed in the form of a gzipped tar file containing all the binaries, libraries and demos. In order to install MOLARIS, the user needs to gunzip the file using

```
$ gunzip Molaris_<version>.tar.gz
```

and then untar the resulting file using:

```
$ tar xvf Molaris_<version>.tar
```

Alternatively, some versions of the tar program accept the following:

```
$ tar xvf Molaris_<version>.tar
$ tar zxvf Molaris_<version>.tar.gz
```

Either of those procedures will create a directory called MOLARIS, whose tree structure is the following:

```
$ tar xvf Molaris_<version>.tar
molaris/
molaris/bin
molaris/doc
molaris/lib
molaris/demo/pdb
molaris/demo/ez_relax
molaris/demo/pl_pka_pdld
molaris/demo/demo3
(...)
```

From now on we will refer to the newly generated MOLARIS directory as $MOLARIS_PATH directory. The directory $MOLARIS_PATH/demo contains all the demos distributed in the current version and scripts for running them. All the cordinates files needed for those demo calculations are included in the directory $MOLARIS_PATH/demo/pdb and the directory $MOLARIS_PATH/lib contains the libraries used by MOLARIS. This manual is located in the $MOLARIS_PATH/doc directory. Finally, the binary is located in the $MOLARIS_PATH/bin directory.

1.2.2 Running MOLARIS

The first step in running the program is to add the path where the binary file is in the .login file. In csh or tcsh this is done by adding the lines:

```
setenv MOLARIS_PATH ``<where_molaris_has_been_installed>''
setenv PATH ``\$MOLARIS\_PATH/bin:\$PATH''
```

If the shell is the Bourne shell, the following lines are needed

```
MOLARIS_PATH=''<where_molaris_has_been_installed>''
export $MOLARIS_PATH
PATH=$MOLARIS_PATH/bin:$PATH
export $PATH
```

The program can be run interactively, answering the questions in a menu driven form, or alternatively the user can create the keywords file needed for the run. Additionally, when running interactively, a file called mola<version>.runMM_DD_YY:HH (where MM is for month, DD is for day, YY is for year and HH is for hour; for example mola<version>.run01_28_00:1 is generated. This file records all the inputs the user has typed on the screen during the interactive run and can be used as a template for future non-interactive runs. The keywords are described in section 2 of this manual, and section **??** gives a brief introduction on how the program works by running three of the demos and providing a quickstart.

In order for the program to run it must know where the libraries are. This information is read in the form of environmental variables. In particular, MOLARIS will look for a file

called .molaris_rc in the $MOLARIS_PATH/bin directory that will contain the definition of these environmental variables. Then, the program will source a file with the same name (if it exists) in the $HOME directory. Finally, MOLARIS will source the ./.molaris_rc file (this is in the current working directory). The description of the .molaris_rc file is given in section 2.1.

1.2.3 Supported platforms

The program has been tested on the following platforms:

version	Release date	supported platforms
MOLARIS 9.05beta	June 2002	Linux (Redhat 6.2) HP-UX OSF1 ALPHA SGI (IRIX 6.5)
MOLARIS 9.04	June 2002	Linux (Redhat 6.2) SGI (IRIX 6.5)
MOLARIS 7.1	August 2000	Linux (Redhat 6.2) IBM (AIX) SGI (IRIX 6.5)
MOLARIS 6.7	May 2000	Linux (Redhat 6.2) IBM (AIX) SGI (IRIX 6.5)

1.3 Quickstart

In order to familiarize the user with the program we show here how to perform some short runs, where the main aspects of the procedure used are emphasized.

To run these short demos, go to the directory $MOLARIS_PATH/demo. Here you will find a script called shrunall. In order to run all the demos (the three described here plus the rest of the demos described in section 3.4) the user must type

```
$ shrunall
```

We recommend to run all the demos in this way in order to be sure the program was properly installed and in order to report all the possible bugs in the demos. Note, however, that this requires a significant number of processors. If, instead, we just want to run a particular demo (X) we will type:

```
$ run demoX
```

See section 3.4 for a complete description of the demos and section 3.5 on how to contact us in order to report bugs in the program.

By now, however, it is more instructive to run the first three demos interactively (these demos can also be run in the background by using the *.inp files in the ez_relax, pl_pka_pdld and demo3 directories in $MOLARIS_PATH/demo respectively). These three demos cor-

respond to a protein relaxation, a pKa calculation and an empirical valence bond (EVB) run. For interactively running a particular demo just go to the corresponding directory and run MOLARIS.

1.3.1 Demo 1: protein relaxation

The first demo performs in a relaxation of the protein BPTI and generates a PDB file with the final structure. Let us go to the $MOLARIS_PATH/demo/ez_relax directory, where all the needed files for the calculation are present and then type

```
$ molaris
```

After this, the program will prompt

```
$Usage:
 For interactive run, please press the Enter key.
 For using input file on command line, please press the Enter key,
 type quit, then type on the command line:
 molaris $<$ input\_file\_name
 or:
 molaris input\_file\_name
```

This message informs the user about the different possibilities of running MOLARIS. The user can run the program interactively, as we will do now, or prepare an input file with all the appropriate commands for running the calculation in the background. In this case, the file bpti_relax.inp in the directory $MOLARIS_PATH/demo/ez_relax contains all the commands needed for the present run (see also section 3.4.1). However, as we want to use the interactive option, we press ¿.

```
 ===========================================
 MOLARIS version 6.7
 Job starting time: Tue May 2 10:45:24 2000
 ===========================================

 Checking the file path and name for ...

 file path and name for AMINO\_LIB -$>$
 ../../lib/amino98.lib

 file path and name for PARM\_LIB -$>$
 ../../lib/parm.lib

 file path and name for EVB\_LIB -$>$
 ../../lib/evb.lib

 file path and name for SOLVENT\_OPT -$>$
 ../../lib/solvent.opt

 directory path \& name for OUT\_DIR -$>$
 /tmp/demo

 All your output files will be written and put in the directory:
 /tmp/demo/bpti\_relax/

 reading data from: ../../lib/amino98.lib

Input the coordinate file name:
```

We can check if the information in the .molaris_rc file has been read properly (for a detailed description of this file see section 2.1). At this point we write the name of the coordinates file of the system we are interested in. MOLARIS accepts both PDB and Mol2 formats, or a combination of them. In this case we type ../pdb/bpti.pdb.

Initially the program checks the coordinates file and look for possible errors in it. If the file is OK then the program will proceed by comparing the residues of the coordinates file with the residues in the topology library, provided with the program and called amino98.lib in the current version. If the file contains a residue that is not in the library a new entry is automatically added to this library.

After checking and writing the topology in a special file called $OUT_DIR/bpti.top (which corresponds to the topology file in previous versions of POLARIS and ENZYMIX) the user is asked what task he/she wants to perform. In this case we will choose ENZYMIX and the program prompts the following table:

```
        Table of the Keywords for the Enzymix Level
        ...........................................

keyword   modifier     example
-------   --------     -------
pre_enz     no         pre_enz
relax       no         relax
ac          no         ac
evb         no         evb
evb2        no         evb2
evb_ab      no         evb_ab
adiab_pot   no         adiab_pot
adiab_tem   no         adiab_tem
end         no         end
help        yes        help <keyword1> <keyword2> ...
help        yes        help all
help        no         help
exit        no         exit
-------------------------------------------------------------------------

enzymix>
```

Here you start to see that the MOLARIS package works as nested tasks, where every keyword follow a hierarchy of execution. In this way, every time we finish a particular task we must write an end statement if we want to save the changes made or exit if we did some mistake and we want to quit without saving. In this particular demo we want to perform a relaxation of the protein, so we select relax. The following table appears:

```
------ The default parameter values of MD to be used in this run ------

     keyword             default value
-----------------  --------------------------
nsteps              500
temperature (k)     300.00
tolerance_temp(k)  3000.00
stepsize (ps)       0.0020000
nbupdate            30
region2a_r          18.0
water_r             18.0
langevin_r          20.0
induce              0
indforce            10
constraint_1        0.030
constraint_2        0.030
```

```
constraint_pair     0
constraint_post     0
constraint_ang      0
cutpp               10.0
cutpw               10.0
cutww               10.0
cutaep              20.0
cutaew              20.0
movie_co            0

-------------------------------------------------------------------------


        Table of the Keywords RELAX Level
        .................................

keyword   modifier   example
-------   --------   ------
md_parm      no      md_parm
rest_in      yes     rest_in rest.in
rest_out     yes     rest_out rest.1
energy_out   yes     energy_out gap.out
end          no      end
help         yes     help <keyword1> <keyword2> ...
help         yes     help all
help         no      help
exit         no      exit
-------------------------------------------------------------------------

relax>
```

Here we have several choices to make. If we just quit the level with end, the program will perform a relaxation taking the default parameters for the MD calculation. Let us change those parameters before quitting the relax level. When typing md_parm we enter in the next hierarchy level and we have all the possible choices in the following table:

```
relax> md_parm
        Table of the Keywords MD_PARM Level
        ...................................


keyword         modifier   example
-------         --------   -------
nsteps            yes      nsteps 500
temperature(K)    yes      temperature 300.0
tolerance_temp    yes      tolerance_temp 3000.0
stepsize (ps)     yes      stepsize 0.002
nbupdate          yes      nbupdate 30
gas_phase         yes      gas_phase 0
region2a_r        yes      region2a_r 18.0
water_r           yes      water_r 18.0
langevin_r        yes      langevin_r 20.0
ex_w_center       yes      ex_w_center 3.0 4.5 2.34
induce            yes      induce 0
indforce          yes      indforce 0
constraint_1      yes      constraint_1 0.03
constraint_2      yes      constraint_2 0.03
constraint_w      yes      constraint_w 30.0
constraint_pair   yes      constraint_pair 5 9 10.0 1.3
constraint_post   yes      constraint_post 10 10. 10. 10. 3.4 -4.6 4.7
constraint_r      yes      constraint_r 5 10.0 50.0 2.0 4.6 7.3
constraint_ang    yes      constraint_ang 10 34 35 10.0 120.
h_constraint      yes      h_constraint 0
movie_co          yes      movie_co rg1
movie_fq          yes      movie_fq 10
```

```
pmf               no       pmf
ub_sampling       no       ub_sampling
fix_region        yes      fix_region 1
fix_atom          yes      fix_atom 8
dist_atoms        yes      dist_atoms 2 5
dist_write_fq     yes      dist_write_fq 10
opt_his           yes      opt_his 1
steep_mini        yes      steep_mini 1
df_mini           yes      df_mini 1 0.0001
log_detail        yes      log_detail 1
help              yes      help <keyword1> <keyword2> ...
help              yes      help all
help              no       help
end               no       end
exit              no       exit
-------------------------------------------------------------------------
```

All the parameters have their default value, but let's say that we want a shorter run and we want to change the temperature and the stepsize. To do so we type:

```
md_parm> nsteps 300
md_parm> temperature 200.0
md_parm> stepsize 0.0002
md_parm> end
```

and we obtain the following new table of parameters:

```
Induced dipoles are not used

      Current Parameters for MD Run:
      ------------------------------

         keyword            value
         -------            -----
         nsteps               300
         temp               200.0
         tolerance_temp    3000.0
         stepsize        0.000200
         nbupdate              30
         gas_phase              0
         region2a_r          18.0
         water_r             18.0
         langevin_r          20.0
         induce                 0
         indforce              10
         constraint_1       0.030
         constraint_2       0.030
         constraint_w       0.000
         constraint_pair        0
         constraint_post        0
         constraint_ang         0
         h_constraint           0
         fdft                   0
         qmmm                   0
         qcff                   0
         pmf                    0
         ub_sampling            0
         fix_region             0
         fix_atom               0
         movie_co               0
         movie_fq              10
         cutpp               10.0 <- protein-protein
         cutpw               10.0 <- protein-water
         cutww               10.0 <- water-water
         cutaep              20.0 <- ac/evb-protein
```

```
          cutaew              20.0 <- ac/evb-water

 total time for this MD run in ps  =    0.060


 relax>
```

Closing the relax level with an additional end keyword will start the run. At the beginning the relevant information is printed (different radii, coordinates for the center, number of solvent molecules generated...) and then the actual MD calculation starts, giving the values of the energies at intervals of 10 steps:

```
(...)
 In dynamics:   Istep=    293   Temp=      201.15      Target=    200.00
 In dynamics:   Istep=    294   Temp=      201.12      Target=    200.00
 In dynamics:   Istep=    295   Temp=      201.10      Target=    200.00
 In dynamics:   Istep=    296   Temp=      201.07      Target=    200.00
 In dynamics:   Istep=    297   Temp=      201.05      Target=    200.00
 In dynamics:   Istep=    298   Temp=      201.02      Target=    200.00
 In dynamics:   Istep=    299   Temp=      200.99      Target=    200.00
 In dynamics:   Istep=    300   Temp=      200.96      Target=    200.00

 rms of all protein heavy atoms       for (x_average-x0) =          0.14
 rms of all protein heavy atoms       for (x_current-x0) =          0.24


 Energies for the system at step    300:
-------------------------------------------------------------------------
 protein - ebond    :    182.97 ethet    :    441.29
           ephi     :     64.85 eitor    :     10.62
           evdw     :    541.47 emumu    :  -1131.45
           ehb_pp   :   -139.03

 water   - ebond    :     67.76 ethet    :     78.94
           evdw     :    183.54 emumu    :  -3588.84
           ehb_ww   :      0.00

 pro-wat - evdw     :   -143.03 emumu    :   -319.59
           ehb_pw   :      0.00

 long    - elong    :    -10.28

 ac      - evd_acp  :      0.00 emumuacp :      0.00
           evd_acw  :      0.00 emumuacw :      0.00
           ehb_acp  :      0.00
           ehb_acw  :      0.00

 evb     - ebond    :      0.00 ethet    :      0.00 ephi     :      0.00
           evdw     :      0.00 emumu    :      0.00 eoff     :      0.00
           egpshift :      0.00

 induce  - eind     :      0.00 eindw    :      0.00

 const.  - ewatc    :    104.52 eproc    :     13.13 edist    :      0.00

 langevin- elgvn    :    -31.80 evdw_lgv :     36.00 eborn    :     -3.93

 system  - epot     :  -3642.85 ekin     :   1410.76 etot     :  -2232.09
 ------------------------------------------------------------------------


 Constraint energy on region I:       0.00
```

Once here, we quit the enzymix level and we enter the analyze level in order to get the

coordinates in PDB format after the relaxation.

```
enzymix> end


task> analyze

        Table of the Keywords for the Analysis Level
        ............................................

keyword   modifier   example
-------   --------   -------
rest_in     yes      rest_in rest.in
rest_to_pdb yes      rest_to_pdb rest.pdb
allres      no       allres
restype     yes      restype ASP
resatom     yes      resatom 1
resbond     yes      resbond 1
resang      yes      resang  1
restor      yes      restor  1
resitor     yes      resitor 1
distatom    yes      distatom 2 5
chkbond     yes      chkbond 50.0
electro     yes      electro 1 18.0 4
center_s    no       center_s
center_r    yes      center_r 5 12
sphereion   yes      sphereion 12.50 3.64 -6.28 10.63
sphereres   yes      sphereres 12.50 3.64 -6.28 10.63
sphereatm   yes      sphereres 12.50 3.64 -6.28 10.63
addbond     yes      addbond 2 5 9 10 18
mutate_res  yes      mutate_res 2 SER
rotate_h    yes      rotate_h 5 12
viewmovie   no       viewmovie
viewpot     no       viewpot
vdwsurf     no       vdwsurf
makepdb     no       makepdb
makelib1    no       makelib1
dock        no       dock
end         no       end
help        yes      help <keyword1> <keyword2> ...
help        yes      help all
help        no       help
exit        no       exit
-------------------------------------------------------------------------


analyze>
```

We choose makepdb:

```
analyze> makepdb

        Table of the Keywords makepdb Level
        ...................................

keyword   modifier   example
-------   --------   ------
residue     yes      residue 2
file_nm     yes      file_nm file.pdb
end         no       end
help        yes      help <keyword1> <keyword2> ...
help        no       help
exit        no       exit
-------------------------------------------------------------------------

makepdb>
```

Then we select the right options and quit the makepdb level:

```
makepdb> residue all

makepdb> file_nm bpti_relaxed.pdb

makepdb> end

a pdb file: bpti_relaxed.pdb


is created


analyze>
```

The program executes the requested commands and it is ready to be quit by double typing end.

At this point it is important to note the use of the non-interactive way of running the program, which allows one to redirect the output. Try, for example

```
$ molaris < bpti_relax.inp > bpti_relax.out
```

which puts the output in a file called solv_pdld.out in the current directory or

```
$ molaris bpti_relax.inp bpti_relax.out
```

that puts the bpti_relax.out file in the $OUT_DIR/bpti_relax directory as described in section 2.1.

1.3.2 Demo 2: pKa calculation

The second demo consists in the calculation of the pKa shift for the aspartic residue number 3 in BPTI. The complete input file for this demo is given in section 3.4.2. Change to the directory $MOLARIS_PATH/demo/pl_pka_pdld and type molaris. After this, as in the previous run, input the name of the PDB filename (../pdb/bpti.pdb).

Now, we will choose the polaris task and the program will prompt us with a table of the options for polaris:

```
task> polaris

        Table of the Keywords for the Enzymix Level
        ...........................................

keyword   modifier     example
-------   --------     -------
pre_pol     no         pre_pol
solv_pdld   no         solv_pdld
solv_fep    no         solv_fep
ai_pdld     no         ai_pdld
bind_pdld   no         bind_pdld
bind_fep    no         bind_fep
pka_pdld    no         pka_pdld
pka_fep     no         pka_fep
redox_pdld  no         redox_pdld
redox_fep   no         redox_fep
logp        no         logp
```

```
titra_ph_0  no         titra_ph_0
titra_ph    no         titra_ph
pka_multi   no         pka_multi
evb_pdld    no         evb_pdld
end         no         end
help        yes        help <keyword1> <keyword2> ...
help        yes        help all
help        no         help
exit        no         exit
-------------------------------------------------------------------------


polaris>
```

We will choose pKa_pdld, and the program will prompt:

```
polaris> pka_pdld

        Table of the Keywords for the pKa_pdld Level
        ........................................

keyword   modifier   example
-------   --------   -------
reg1_res    yes      reg1_res 2
pka_w       yes      pka_w     3.0
pdld_fn     yes      pdld_fn asp.pdld
regII_r     yes      regII_r 16.0
config      yes      config 0 5
md_parm_r   no       md_parm_r
md_parm_w   no       md_parm_w
md_parm_p   no       md_parm_p
help        yes      help <keyword1> <keyword2> ...
help        no       help
end         no       end
exit        no       exit
-------------------------------------------------------------------------

pka_pdld>
```

Next we will change the value of the intrisic pKa for residue 3, which will be called our region I. We will also set the number of configurations to run and the characteristics of the dynamics. Thus, we will tell the program to run the calculation on the initial protein structure and on 2 more conformations which will be generated automatically by MD runs. Also we will restrict the MD runs to 300 steps for this test run:

```
pka_pdld> reg1_res 3

atoms added to region I:
atom#  charg_a  charg_b
-----  -------  -------
   43    0.000    0.000
   44    0.000    0.000
   45    0.000    0.000
   46    0.450    0.000
   47   -0.725    0.000
   48   -0.725    0.000

pka_pdld> pka_w 3.92
pka_pdld> config 1 2
pka_pdld> md_parm_r
pka_pdld> 300
pka_pdld> end
pka_pdld> md_parm_w
pka_pdld> 300
```

```
pka_pdld> end
pka_pdld> md_parm_p
pka_pdld> 300
pka_pdld> end
```

In order to run the program we will just end the level and the calculation will proceed. The explanation of the output will be given in the corresponding section of the next chapters, but here we just show the final result of the pKa calculations of this very simple (and of course unreliable because of the short run) test:

```
          PDLD SEMI-MACROSCOPIC ESTIMATE FOR pKa
          ......................................


effective dielectric       2       4       6       8      20      40      80
epsilon_p(e_p)

pKa_intr for str. 1       5.86    4.88    4.55    4.39    4.09    3.99    3.95

pKa_intr for str. 2       6.64    5.27    4.81    4.59    4.18    4.04    3.97

-------------------------------------------------------------------------------
aver pKa_int              6.25    5.08    4.68    4.49    4.13    4.02    3.96


estimated apparent pKa            4.55
```

1.3.3 Demo 3: EVB calculation

This demo provides an example of an EVB calculation for the first step of the reaction catalyzed by subtilisin. The details of the calculation are given in the following chapters, but here we want the user to realize the simplicity of running EVB calculations on a given enzymatic reaction. First it is important to realize that in order to study an enzymatic reaction we should compare the reaction mechanism in the protein and the corresponding reaction in water. Thus, most of the times we are interested in comparing the free energy profiles in both environments. We will outline here the procedure for the water run. As can be seen from the input files in the mentioned directories the structure of the run is almost identical, differing just in the numbering of the EVB atoms. The input files are printed in section 3.4.3.

As before, we start by changing to the demo directory for the water run and we type molaris. After this we are prompted for the PDB file, which in this case is ../../pdb/subwat.pdb. We can relax the initial system by typing something like:

```
enzymix
 relax
    rest_out subrelax.res
    energy_out subrelax.gap
    md_parm
       nsteps 100
       temperature 30
       stepsize 0.0002
       region2a_r 16
       water_r 16
       langevin_r 18
    end
 end
```

```
end
```

The next step is to get the numbering of the atoms that will define the EVB region. This procedure is described in detail in the following chapters. First, we get some information of the system with the analyze task

```
analyze> resatom 64


atom list for residue:      HIS

number  name  type      x          y          z      charge      atoms bonded(name)
------  ----  ----   -------    -------    -------   ------   -------------------------
 882    N      N3     19.400     19.300     23.700   -0.400   C    HN   CA
 883    HN     H2     18.532     19.478     24.164    0.400   N
 884    CA     C4     20.300     20.300     23.400   -0.097   N    HA   CB   C
 885    HA     H1     20.711     20.152     22.501    0.097   CA
 886    CB     C4     19.400     21.600     23.600   -0.194   CA   HB1  HB2  CG
 887    HB1    H1     19.358     21.707     24.593    0.097   CB
 888    HB2    H1     18.569     21.393     23.084    0.097   CB
 889    CG     C3     19.800     23.000     23.100    0.035   CB   ND1  CD2
 890    ND1    N3     20.100     24.000     23.800    0.013   CG   HD1  CE1
 891    HD1    H1     20.235     24.014     24.791    0.187   ND1
 892    CE1    C3     20.200     25.000     23.000    0.110   ND1  HE1  NE2
 893    HE1    H1     20.402     25.941     23.271    0.077   CE1
 894    NE2    N2     20.000     24.600     21.800   -0.550   CE1  CD2
 895    CD2    C3     19.500     23.300     21.900    0.058   CG   NE2  HD2
 896    HD2    H1     19.030     22.749     21.210    0.070   CD2
 897    C      C3     21.400     20.400     24.300    0.550   CA   O    N
 898    O      O1     22.600     20.900     24.000   -0.550   C

Total charge of this residue:      0.000
```

We do the same for Ser221, Tyr276 and Gly277. With this information in mind (especially the numbering of the relevant atoms) we can enter the enzymix task again to perform the EVB run.

```
enzymix> evb

reading evb data from:
/home/jvilla/treball/molaris/MOLARIS_NG/lib/evb.lib

EVB library is read successfully


The default MD parameters for this run are as the following, use md_parm
to change them if you want to

(...)


        Table of the Keywords EVB Level
        ...............................

  keyword    modifier  example
-----------  --------  -------
evb_simp        yes    evb_simp 1 18.0 6
evb_parm        no     evb_parm
evb_state       yes    evb_state 2 1.0 0.0
evb_atm         yes    evb_atm 2 0.2 C+ 0.5 C0
evb_bnd         yes    evb_bnd 1 2 5
evb_d_con       yes    evb_d_con 2 5 1.5 50 1
evb_a_con       yes    evb_a_con 2 5 7 90 50 1
evb_p_con       yes    evb_p_con 2 10.0 10.0 10.0 6.363 2.010 -0.367 1
```

```
 pentacoord      yes    pentacoord 235 2 233 237
 md_parm         no     md_parm
 evb_entropy     yes    evb_entropy 1
 ave_elec        yes    ave_elec 500
 map_pf          yes    map_pf  11 1 2
 gas_dg          yes    gas_dg 2 -5.0
 Hij             yes    Hij 1 2 6 9 -120 0.6
 Hr              yes    Hr 1 2 6 4 9 -120 0.6
 H3              yes    H3 1 2 5 7 8 300 0.6
 R4              yes    R4 1 2 1.44
 induce_evb      yes    induce_evb 1
 exponential     yes    ne 1 2 2.5
 vdw_pair        yes    vdw_pair 2 5
                        vdw_pair 2 5 60 760
 add_ang         yes    add_ang  2 5 8 200 120 2
 read_evb        yes    read_evb evb.in
 rest_in         yes    rest_in rest.in
 rest_out        yes    rest_out rest.out
 rest_fq         yes    rest_fq 100
 rest_step       yes    rest_step 10 200 320
 ad_restout      yes    ad_restout 1 2 5.0 20 30
 rest_constr     yes    rest_constr 1
 energy_out      yes    energy_out gap.out
 centroid        no     centroid
 check_f         yes    check_f 1
 end             no     end
 help            yes    help <keyword1> <keyword2> ...
 help            yes    help all
 help            no     help
 exit            no     exit
-------------------------------------------------------------------------

 evb>
```

We will start by defining the number of resonance states and determine the sequence of mapping steps in the free energy perturbation (FEP) procedure. In this case we will generate 11 frames going from state 1 to state 2.

```
 evb> evb_state 3 1.0 0.0 0.0 1

 evb> map_pf 11 1 2

 evb>
```

After this we can define the charges of the EVB atoms and their bonding pattern in the different VB configurations and finally change the default values of the MD run. Additionally, we give the program information about the EVB run that is explained in the following sections:

```
      evb_atm   889   0.0350   C+   0.1640   C+   0.1640   C+
      evb_atm   890   0.0150   N+  -0.1610   N+  -0.1610   N+
      evb_atm   891   0.1870   H0   0.1870   H0   0.1870   H0
      evb_atm   892   0.1100   C+   0.5450   C+   0.5450   C+
      evb_atm   893   0.0750   H0   0.0750   H0   0.0750   H0
      evb_atm   894  -0.5200   N+  -0.1610   N+  -0.1610   N+
      evb_atm   895   0.0280   C+   0.0960   C+   0.0960   C+
      evb_atm   896   0.0700   H0   0.0680   H0   0.0680   H0
      evb_atm  3026  -0.4270   O0  -1.0000   O-  -0.2000   O0
      evb_atm  3027   0.4270   H0   0.1870   H0   0.1870   H0
      evb_atm  3854   0.3920   C+   0.3920   C+   0.2000   C0
      evb_atm  3855  -0.3920   O0  -0.3920   O0  -1.0000   O-
      evb_atm  3837  -0.0970   C0  -0.0970   C0  -0.0970   C0
      evb_atm  3838   0.0970   H0   0.0970   H0   0.0970   H0
      evb_atm  3856  -0.4000   N+  -0.4000   N+  -0.4000   N+
```

```
evb_atm  3857   0.4000   H0   0.4000   H0   0.4000   H0
evb_bnd 0 889 895
evb_bnd 0 889 890
evb_bnd 0 890 891
evb_bnd 0 890 892
evb_bnd 0 892 893
evb_bnd 0 892 894
evb_bnd 0 894 895
evb_bnd 0 895 896
evb_bnd 1 3026 3027
evb_bnd 2 894 3027
evb_bnd 3 894 3027
evb_bnd 3 3026 3854
evb_bnd 0 3854 3855
evb_bnd 0 3838 3837
evb_bnd 0 3837 3854
evb_bnd 0 3854 3856
evb_bnd 0 3856 3857
gas_dg 1   0.0
gas_dg 2 115.0
gas_dg 3  50.0
evb_parm
   ifglag_r4 0
end
rest_out evb_sub-rx.res # store the relaxed structure
                        # into a restart file
md_parm
   temperature 300
   ss 0.0001            # using a small time-step
   nsteps 30000
   region2a_r 16
   water_r 16
   langevin_r 18
   constraint_pair    894 3026 3.0 3.0
   constraint_pair   3026 3854 3.0 3.0
   constraint_post   3026 3.0 3.0 3.0 21.4 27.0 20.6
end
```

After the end command, the program will start the FEP protocol, according to the settings above. The final result of the program is a bunch of *.map files, each of them corresponding to every frame. These data will be collected and processed by the mapping procedure described below.

Chapter 2

Reference manual

The first thing the user must do in order to run MOLARIS is to include the path of the $MOLARIS_PATH/bin directory in his $HOME/.login file as described in section 1.2.1. Then just type

```
$ molaris
```

and follow interactively the execution of the program. Once the user is familiarized with the program he/she is encouraged to write the command files using his/her favorite editor and run the program using option A

```
$ molaris < input_file.extension > output_file.extension
```

or option B

```
$ molaris input_file.extension output_file.extension
```

The difference between the latter two options is the location where the program will write the output file. When using the first option the output file is written in the current directory while when using the second option it is written in the $OUT_DIR/input_file directory as described below. Also, when running the program interactively an input file called polaris.run<date> is generated in order to get a template useful for future runs.

2.1 The .molaris_rc file

The .molaris_rc file contains the definition of the environmental variables used by the program. This file looks like:

```
setenv AMINO_LIB $MOLARIS_PATH/lib/amino98.lib
setenv PARM_LIB $MOLARIS_PATH/lib/parm.lib
setenv EVB_LIB $MOLARIS_PATH/lib/evb.lib
setenv SOLVENT_OPT $MOLARIS_PATH/lib/solvent.opt
setenv OUT_DIR ./output
setenv PDB_CONECT $MOLARIS_PATH/lib/pdb_dictionary
```

The meaning of these environmental variables is:

- AMINO_LIB: location of the amino acid library containing the topology and charges of the most common residues. If our system contains some residue that is not yet in the library it adds it automatically. This is described in section 3.2.1. The atom names in the PDB file are translated into atom types using the information of this library.
- PARM_LIB: location of the library containing the ENZYMIX force field. The program obtains the atom types from the atom names in the PDB file through the use of the amino acid library. This file is described in section 3.2.2.
- EVB_LIB: location of the library containing the parameters to be used for region I atoms (EVB atoms) in case an EVB calculation is requested. This file is described in section .
- SOLVENT_OPT: location of the parameters file containing some default settings of the program. This file is scheduled to disappear in a future release of the program, being incorporated instead in the form of keywords.
- OUT_DIR: `MOLARIS` writes all the output files in this directory, unless the option B of running the program (described at the beginning of chapter 2) is used. In that case, the program automatically generate the $OUT_DIR/input_file directory and place there all the output files.

When running `MOLARIS`, the program sources the .molaris_rc file located in the $MOLARIS_PATH/bin directory. After this, the program sources the file $HOME/.molaris_rc if it exists. Finally, the program also sources the file ./.molaris_rc. In this way we ensure that all the files are sourced in the appropriate order, and the user is able to set different environmental variables depending on his/her needs.

2.2 A sample `MOLARIS` input file

This is a simple `MOLARIS` input file that can be used for obtaining the pKa of a particular residue of the system (for a more complete version of this input file see section **??**).

```
../pdb/bpti.pdb
polaris                       # enter the POLARIS module
   pKa_pdld                   # enter the pKa calculation
      reg1_res 1              # identify region I. Do pKa of residue 1
      config 1 2              # configurations to be done
   end
end
end
```

The format for the input file in `MOLARIS` is free, and the user is allowed to add as many comment lines (beginning with the # sign) and as many blank lines as he/she wants. Also, the comments can be included after the keywords, in order to make the file self-explanatory. Essentially, `MOLARIS` does not care about any character in a line after it finds the # sign.

The input file is divided into several sections. The first non-comment and non-blank line must be the name of the PDB file of the system the user is working on. After this, one or more of the tasks in `MOLARIS` can be especified (ANALYZE, POLARIS and/or

ENZYMIX) with their corresponding subkeywords.

There are four kinds of keywords:

1. list keywords: may enclose one or more keywords, which in turn may be also list keywords. In the example above, POLARIS, ANALYZE, solv_pdld and pol_parm are list keywords.

2. modifier keywords: used for assigning a given variable (config and nsteps in the example)

3. information keywords: give some information on the system (*e.g.* , resatom)

4. exit keywords: they are used for exiting the current level.

This division is, however, just for clarification. It has a very little effect on how the input file for the program is written. In the next sections we will list and explain all the current keywords in MOLARIS. First, however, a global explanation of the flow chart of the program is needed.

2.3 Flowchart of MOLARIS

In Figure we show the general structure of program MOLARIS.

It is composed of three main modules

1. ANALYZE: this module contains all the utilities for obtaining information on the structure of the system under study (numbering of atoms and residues, measuring of geometric parameters,...)

2. POLARIS: uses the PDLD approach in order to study properties of macromolecules (*e.g.* pKa, REDOX, binding, activation energies, etc)

3. ENZYMIX: Simulates properties of proteins using the surface constrained all-atom protein/solvent model. It is the main tool for running the EVB approach, but it can be also used for more general MD and FEP calculations.

The use of one of these options or tasks makes the program call one of the corresponding modules within MOLARIS. The next step is to specify the particular keywords in order to set the run. Thus, each of the keywords in figure 2.1 are list keywords and must finish with an end statement as we explained above.

Figure 2.1: Flow chart for the MOLARIS program, with an indication of all the list keywords and subkeywords. The program is divided into three main modules, marked in black. Keywords that are given in gray are still under development or being implemented in older versions. As seen from the figure, most of the tasks use molecular dynamics as a tool, and the list md_parm controls how this MD are done.

2.4 MOLARIS keywords

In the following sections we present all the keywords available in the current version of MOLARIS. Sections **??**, **??**, **??**, and **??** include the keywords as they appear in the menu-driven execution.

Rows with gray background indicate features currently under development.

2.4.1 general keywords

In adition to the keywords in the next sections, in all the levels it is also possible to use the special keywords:

keyword	example	more info
end	end	page **??**
help	help <keyword1> <keyword2> ...	page **??**
help	help all	page **??**
help	help	page **??**
exit	exit	page **??**

Also, it is also possible to call the shell by typing:

```
!<unix shell command>
```

For example:

```
!ls -l
```

2.4.2 analyze task keywords

keyword	example	more info
rest_in	rest_in rest.in	page **??**
rest_to_pdb	rest_to_pdb rest.pdb	page **??**
allres	allres	page **??**
restype	restype ASP	page **??**
resatom	resatom 1	page **??**
resbond	resbond 1	page **??**
resang	resang 1	page **??**
restor	restor 1	page **??**
resitor	resitor 1	page **??**
distatom	distatom 2 5	page **??**
chkbond	chkbond 50.0	page **??**
electro	electro 1 18.0 4	page **??**
center_s	center_s	page **??**
center_r	center_r 5 12	page **??**
sphereion	sphereion 12.50 3.64 -6.28 10.63	page **??**
sphereres	sphereres 12.50 3.64 -6.28 10.63	page **??**
sphereatm	sphereres 12.50 3.64 -6.28 10.63	page **??**
addbond	addbond 2 5 9 10 18	page **??**
mutate_res	mutate_res 2 SER	page **??**
rotate_h	rotate_h 5 12	page **??**
viewmovie	viewmovie	page **??**
viewpot	viewpot	page **??**
vdwsurf	vdwsurf	page **??**
makepdb	makepdb	page **??**
makelib1	makelib1	page **??**
dock	dock	page **??**

viewmovie level keywords

keyword	example	more info
file_nm	file_nm movie.datA1	page ??
view_fq	view_fq 20	page ??
vwall	vwall	page ??
vwres	vwres 2	page ??
form	form pdb	page ??
frame	frame 40	page ??

viewpot level keywords

keyword	example	more info
file_nm	file_nm map_pot.dat	page ??
pot	pot ap2	page ??
dif	dif aw1 ap1	page ??
ave	ave ap1 ap2	page ??

makepdb level keywords

keyword	example	more info
residue	residue 2	page ??
file_nm	file_nm file.pdb	page ??

makelib1 level keywords

keyword	example	more info
file_nm	file_nm mol.dat	page ??
connect	connect 2 9	page ??

dock level keywords

keyword	example	more info
pdb2	pdb2 mole2.pdb	page ??
res_in_1	res_in_1 3	page ??
res_in_2	res_in_2 4	page ??
f_dock	f_dock new2.pdb	page ??

2.4.3 polaris task keywords

keyword	example	more info
pre_pol	pre_pol	page ??
solv_pdld	solv_pdld	page ??
solv_fep	solv_fep	page ??
ai_pdld	ai_pdld	page ??
bind_pdld	bind_pdld	page ??
bind_fep	bind_fep	page ??
pka_pdld	pka_pdld	page ??
pka_fep	pka_fep	page ??
redox_pdld	redox_pdld	page ??
redox_fep	redox_fep	page ??
logp	logp	page ??
titra_ph_0	titra_ph_0	page ??
titra_ph	titra_ph	page ??
pka_multi	pka_multi	page ??

evb_pdld	evb_pdld	page **??**

pre_pol level keyword

keyword	example	more info
ionres	ionres 3	page **??**
unionres	unionres 3	page **??**
iontyp	iontyp ASP	page **??**
setcrg	setcrg 1 0.50	page **??**
setcrg0	setcrg0 2	page **??**
electro	electro 1 18.0 4	page **??**
set_opt	set_opt	page **??**
rest_in	rest_in rest.in	page **??**

keyword	example	more info
rg	rg 18.5	page **??**
outer	outer 3.0	page **??**
inner	inner 1.0	page **??**
rdcutl	rdcutl 6.0	page **??**
rdcutp	rdcutp 10.0	page **??**
micro	micro 0	page **??**
log	log 0	page **??**
lgvn	lgvn 0	page **??**
ndxp	ndxp 30	page **??**
ncenter	ncenter 1	page **??**
itl	itl 20	page **??**
itp	itp 3	page **??**
field	field 0	page **??**
potent	potent 0	page **??**
icut	icut 1	page **??**
iref	iref 1	page **??**
ionic	ionic 0 0.5	page **??**
pdld_atom	pdld_atom 20	page **??**
rp	rp 40 2.30	page **??**
ionic	ionic 1 0.5	page **??**

set_opt sublevel keywords Table of rp's:

solv_pdld level keywords

keyword	example	more info
reg1_res	reg1_res 3	page **??**
reg1_atm	reg1_atm 2 to 50	page **??**
ab_crg	ab_crg 20 0.50 0.02	page **??**
config	config 1 5	page **??**
pdld_fn	pdld_fn asp.pdld	page **??**
pdld_center	pdld_center 12.5 2.3 8,9	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

md_parm sublevel keywords In this level all the keywords are modifiers (M) and can be used to change the default parameters for the molecular dynamics simulations. Note that in the PDLD calculations this keyword has the suffix "_r", "_w" or "_p" depending of the type of calculation on which we want to change the parameters.

keyword	example	more info
nsteps	nsteps 500	page ??
temperature(K)	temperature 300.0	page ??
tolerance_temp	tolerance_temp 3000.0	page ??
stepsize (ps)	stepsize 0.002	page ??
nbupdate	nbupdate 30	page ??
gas_phase	gas_phase 0	page ??
region2a_r	region2a_r 18.0	page ??
water_r	water_r 18.0	page ??
langevin_r	langevin_r 20.0	page ??
ex_w_center	ex_w_center 3.0 4.5 2.34	page ??
induce	induce 0	page ??
indforce	indforce 0	page ??
constraint_1	constraint_1 0.03	page ??
constraint_2	constraint_2 0.03	page ??
constraint_w	constraint_w 30.0	page ??
constraint_pair	constraint_pair 5 9 10.0 1.3	page ??
constraint_post	constraint_post 10 10. 10. 10. 3.4 -4.6 4.7	page ??
constraint_ang	constraint_ang 10 34 35 10.0 120.	page ??
h_constraint	h_constraint 0	page ??
movie_co	movie_co rg1	page ??
movie_fq	movie_fq 10	page ??
fix_region	fix_region 1	page ??
fix_atom	fix_atom 8	page ??

solv_fep level keywords

keyword	example	more info
reg1_res	reg1_res 3	page ??
reg1_atm	reg1_atm 2 to 50	page ??
ab_crg	ab_crg 20 0.50 0.02	page ??
map_pf	map_pf 11	page ??
pdld_fn	pdld_fn asp.pdld	page ??
pdld_center	pdld_center 12.5 2.3 8,9	page ??
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

ai_pdld level keywords

keyword	example	more info
pre_ab	pre_ab	page ??
config	config 1 5	page ??
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

pre_ab sublevel keywords *under construction*

bind_pdld level keywords

keyword	example	more info
reg1_res	reg1_res 3	page ??
reg1_atm	reg1_atm 2 to 50	page ??
ab_crg	ab_crg 20 0.50 0.02	page ??
config	config 1 5	page ??
pdld_fn	pdld_fn asp.pdld	page ??

pdld_center	pdld_center 12.5 2.3 8,9	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

bind_fep level keywords

keyword	example	more info
reg1_res	reg1_res 3	page **??**
reg1_atm	reg1_atm 2 to 50	page **??**
ab_crg	ab_crg 20 0.50 0.02	page **??**
map_pf	map_pf 11	page **??**
pdld_fn	pdld_fn asp.pdld	page **??**
pdld_center	pdld_center 12.5 2.3 8,9	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

pka_pdld level keywords

keyword	example	more info
reg1_res	reg1_res 2	page **??**
pka_w	pka_w 3.0	page **??**
pdld_fn	pdld_fn asp.pdld	page **??**
regII_r	regII_r 16.0	page **??**
config	config 0 5	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

pka_fep level keywords

keyword	example	more info
reg1_res	reg1_res 3	page **??**
reg1_atm	reg1_atm 2 to 50	page **??**
ab_crg	ab_crg 20 0.50 0.02	page **??**
map_pf	map_pf 11	page **??**
pdld_fn	pdld_fn asp.pdld	page **??**
pka_w	pka_w 3.90	page **??**
pdld_center	pdld_center 12.5 2.3 8,9	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

redox_pdld level keywords

keyword	example	more info
reg1_res	reg1_res 3	page **??**
reg1_atm	reg1_atm 2 to 50	page **??**
ab_crg	ab_crg 20 0.50 0.02	page **??**
config	config 1 5	page **??**
pdld_fn	pdld_fn asp.pdld	page **??**
redox_w	redox_w 200	page **??**
pdld_center	pdld_center 12.5 2.3 8,9	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

redox_fep level keywords

keyword	example	more info
reg1_res	reg1_res 3	page **??**
reg1_atm	reg1_atm 2 to 50	page **??**
ab_crg	ab_crg 20 0.50 0.02	page **??**
map_pf	map_pf 11	page **??**
pdld_fn	pdld_fn asp.pdld	page **??**
redox_w	redox_w 200	page **??**
pdld_center	pdld_center 12.5 2.3 8,9	page **??**
md_parm_r	md_parm_r	page 26
md_parm_w	md_parm_w	page 26
md_parm_p	md_parm_p	page 26

titra_ph_0 level keywords

keyword	example	more info
respka	respka 10 14.5	page **??**
typpka	typpka SER 14.5	page **??**
calpka	calpka	page **??**
titra_r	titra_r 4 20.0	page **??**
titra_n	titra_n 10 13 17	page **??**
monte_r	monte_r 10.0	page **??**
dielect	dielect 40.0	page **??**
ph_focus	ph_focus 1 4 0.2	page **??**
temperature	temp 200.0	page **??**
nsteps	nsteps 40	page **??**

evb_pdld level keywords

To run evb_pdld calculation, you must have already performed at least 2 EVB runs using keywords enzymix/evb for reactant and transition state, if you do not have 2 restart files from these 2 EVB runs, please exit here.

Sometimes you may also need to run enzymix and evb energy mapping for the whole reaction to find out the densities at the transition state.

To run evb_pdld calculation, you must have already performed at least 2 EVB runs using keywords enzymix/evb for reactant and transition state, if you do not have 2 restart files from these 2 EVB runs, please exit here.

Sometimes you may also need to run enzymix and evb energy mapping for the whole reaction to find out the densities at the transition state.

keyword	example	more info
read_evb yes read_evb file.evb	page **??**	
restart1 yes restart1 restart1.rest	page **??**	
restart2 yes restart2 restart2.rest	page **??**	

2.4.4 enzymix task keywords

keyword	example	more info
pre_enz	pre_enz	page **??**
relax	relax	page **??**
ac	ac	page **??**
evb	evb	page **??**
evb2	evb2	page **??**
evb_ab	evb_ab	page **??**

adiab_pot	adiab_pot	page **??**
adiab_tem	adiab_tem	page **??**

pre_enz level keywords

keyword	example	more info
ionres	ionres 3	page **??**
unionres	unionres 3	page **??**
iontyp	iontyp ASP	page **??**
setcrg	setcrg 1 0.50	page **??**
setcrg0	setcrg0 2	page **??**
electro	electro 1 18.0 4	page **??**
rest_in	rest_in rest.in	page **??**
rest_constr	rest_constr 1	page **??**

evb level keywords

keyword	example	more info
evb_simp	evb_simp 1 18.0 6	page **??**
evb_parm	evb_parm	page **??**
evb_state	evb_state 2 1.0 0.0	page **??**
evb_atm	evb_atm 2 0.2 C+ 0.5 C0	page **??**
evb_bnd	evb_bnd 1 2 5	page **??**
evb_d_con	evb_d_con 2 5 1.5 50 1	page **??**
evb_a_con	evb_a_con 2 5 7 90 50 1	page **??**
evb_p_con	evb_p_con 2 10.0 10.0 10.0 6.363 2.010 -0.367 1	page **??**
pentacoord	pentacoord 235 2 233 237	page **??**
md_parm	md_parm	page 26
evb_entropy	evb_entropy 1	page **??**
ave_elec	ave_elec 500	page **??**
map_pf	map_pf 11 1 2	page **??**
gas_dg	gas_dg 2 -5.0	page **??**
Hij	Hij 1 2 6 9 -120 0.6	page **??**
Hr	Hr 1 2 6 4 9 -120 0.6	page **??**
H3	H3 1 2 5 7 8 300 0.6	page **??**
R4	R4 1 2 1.44	page **??**
induce_evb	induce_evb 1	page **??**
exponential	ne 1 2 2.5	page **??**
vdw_pair	vdw_pair 2 5	page **??**
add_ang	add_ang 2 5 8 200 120 2	page **??**
read_evb	read_evb evb.in	page **??**
rest_in	rest_in rest.in	page **??**
rest_out	rest_out rest.out	page **??**
rest_fq	rest_fq 100	page **??**
rest_step	rest_step 10 200 320	page **??**
ad_restout	ad_restout 1 2 5.0 20 30	page **??**
rest_constr	rest_constr 1	page **??**
energy_out	energy_out gap.out	page **??**
centroid	centroid	page **??**
check_f	check_f 1	page **??**

evb_entropy sublevel keywords

keyword	example	more info
init_1	init_1 20.0	page **??**
finl_1	finl_1 0.0	page **??**
mass_1	mass_1 50.0	page **??**
ee_map	ee_map 1.0	page **??**

2.4.5 evb2 level keywords

This is a modification of the evb level in which the definition of the EVB region can be updated. It is identical to the evb level, with the additional keywords:

keyword	example	more info
evb_set	evb_set 3	page ??
evb_simp	evb_simp 1 18.0 6	page ??
evb_parm	evb_parm	page ??
evb_state	evb_state 2 1.0 0.0	page ??
evb_atm	evb_atm 2 0.2 C+ 0.5 C0	page ??
evb_bnd	evb_bnd 1 2 5	page ??
evb_d_con	evb_d_con 2 5 1.5 50 1	page ??
evb_a_con	evb_a_con 2 5 7 90 50 1	page ??
evb_p_con	evb_p_con 2 10.0 10.0 10.0 6.363 2.010 -0.367 1	page ??
pentacoord	pentacoord 235 2 233 237	page ??
md_parm	md_parm	page 26
evb_entropy	evb_entropy 1	page ??
ave_elec	ave_elec 500	page ??
map_pf	map_pf 11 1 2	page ??
gas_dg	gas_dg 2 -5.0	page ??
Hij	Hij 1 2 6 9 -120 0.6	page ??
Hr	Hr 1 2 6 4 9 -120 0.6	page ??
H3	H3 1 2 5 7 8 300 0.6	page ??
R4	R4 1 2 1.44	page ??
induce_evb	induce_evb 1	page ??
exponential	ne 1 2 2.5	page ??
vdw_pair	vdw_pair 2 5	page ??
add_ang	add_ang 2 5 8 200 120 2	page ??
read_evb	read_evb evb.in	page ??
rest_in	rest_in rest.in	page ??
rest_out	rest_out rest.out	page ??
rest_fq	rest_fq 100	page ??
rest_step	rest_step 10 200 320	page ??
ad_restout	ad_restout 1 2 5.0 20 30	page ??
rest_constr	rest_constr 1	page ??
energy_out	energy_out gap.out	page ??
centroid	centroid	page ??
check_f	check_f 1	page ??

2.4.6 ac level keywords

keyword	example	more info
reg1_res	reg1_res 2 3	page ??
reg1_atm	reg1_atm 2 to 20	page ??
ab_crg	ab_crg 20 0.50 0.02	page ??
ab_vdw	ab_vdw 20 H1 DH	page ??
md_parm	md_parm	page 26
read_ac	read_ac heme.ac	page ??
map_lambda	map_lambda 1.0	page ??
map_pf	map_pf 11 1 2	page ??
entropy	entropy	page ??
rest_in	rest_in heme.rest	page ??
rest_out	rest_out heme.rest	page ??
rest_fq	rest_fq 100	page ??
rest_constr	rest_constr 1	page ??
energy_out	energy_out gap.out	page ??
check_f	check_f 1	page ??

2.4.7 relax level keywords

keyword	example	more info
md_parm	md_parm	page 26
rest_in	rest_in rest.in	page **??**
rest_out	rest_out rest.1	page **??**
energy_out	energy_out gap.out	page **??**

2.4.8 adiab_pot, adiab_tem, levels keywords

These levels share the same keywords. The difference between them is in the internal settings and we use several levels for them for conceptual simplicity (see definition of keywords in **??**).

keyword	example	more info
evb_simp	evb_simp 1 18.0 6	page **??**
evb_parm	evb_parm	page **??**
evb_state	evb_state 2 1.0 0.0	page **??**
evb_atm	evb_atm 2 0.2 C+ 0.5 C0	page **??**
evb_bnd	evb_bnd 1 2 5	page **??**
evb_d_con	evb_d_con 2 5 1.5 50 1	page **??**
evb_a_con	evb_a_con 2 5 7 90 50 1	page **??**
evb_p_con	evb_p_con 2 10.0 10.0 10.0 6.363 2.010 -0.367 1	page **??**
pentacoord	pentacoord 235 2 233 237	page **??**
md_parm	md_parm	page 26
evb_entropy	evb_entropy 1	page **??**
ave_elec	ave_elec 500	page **??**
gas_dg	gas_dg 2 -5.0	page **??**
Hij	Hij 1 2 6 9 -120 0.6	page **??**
Hr	Hr 1 2 6 4 9 -120 0.6	page **??**
H3	H3 1 2 5 7 8 300 0.6	page **??**
R4	R4 1 2 1.44	page **??**
induce_evb	induce_evb 1	page **??**
exponential	ne 1 2 2.5	page **??**
vdw_pair	vdw_pair 2 5	page **??**
add_ang	add_ang 2 5 8 200 120 2	page **??**
read_evb	read_evb evb.in	page **??**
rest_in	rest_in rest.in	page **??**
rest_out	rest_out rest.out	page **??**
rest_fq	rest_fq 100	page **??**
rest_step	rest_step 10 200 320	page **??**
ad_restout	ad_restout 1 2 5.0 20 30	page **??**
rest_constr	rest_constr 1	page **??**
energy_out	energy_out gap.out	page **??**
centroid	centroid	page **??**
check_f	check_f 1	page **??**

2.4.9 Alphabetical list of keywords

In this section all the keywords from `MOLARIS` are listed alphabetically and a short explanation is given for each of them.

- EVB_SET

  USAGE: evb_set 3
  specifies # of different (e.g. 3) sets of definitions for evb states, atoms and bonds, ...

- EVB_PARM

USAGE: evb_parm
changes EVB parameters, overwriting parameters from evb.lib for the current evb run.

- EVB_STATE

  USAGE: evb_state 2 1.0 0.0 OR evb_state 3 0.0 1.0 0.0 1 OR evb_state 3 0.0 1.0 0.0 2 / specifies the # of the resonance states (e.g. 2 or 3) and the density for each state. In case there are more than 2 states, the last number means that the mapping is for states 1->2 (e.g. 1 in the 2nd example) or for states 2->3 (e.g. 2 in the 3rd example)

- EVB_ATM

  USAGE: evb_atm 2 0.2 C+ 0.5 C0
  specifies an EVB atom with charges and atom types for all the resonance states (e.g. atom 2 with a charge of 0.2 and type C+ for the 1st state, 0.5 and C0 for the 2nd state).

- EVB_BND

  USAGE: evb_bnd 1 2 5 OR evb_bnd 1 2 5
  specifies an EVB bond for a specified resonance state (e.g. atoms 2 and 5 form an EVB bond in the resonance state 1) If the first number is zero, the atoms form a bond in ALL RSs.

- EVB_D_CON

  USAGE: evb_d_con 2 5 1.5 50 1 specifies the distance constraints for evb atoms (e.g. atoms 2 and 5, distanc=1.5A, force=50, constraint energy is added to the energy of evb state 1 and energy output file).

- EVB_A_CON

  USAGE: evb_a_con 2 5 7 90 50 1 specifies the angle constraints for evb atoms (e.g. atoms 2,5 & 7, angle=90, force=50, constraint energy is added to the energy of evb state 1 and energy output file).

- PENTACOORD

  USAGE: pentacoord 235 2 233 237
  atom 235 is pentacoordinated in RS 2. In that RS, atoms 233, 235 and 237 form an straight line.

- MD_PARM

  USAGE: md_parm OR md_parm_r OR md_parm_w OR md_parm_p
  changes the MD parameters for the current run. In POLARIS, _r, _w, _p change parameters specifically for relaxation, water and protein runs, respectively.

- ENTROPY

  USAGE: entropy
  Calls the entropy calculation. Under development.

- MAP_PF

  USAGE: map_pf 1 OR map_pf 11 1 2
  specifies the # of the steps for MD simulations to map free energy profile. in the first example, which is the default, only one MD run, no free energy profile. in the second example, use mapping interval of 0.1, generating 11 energy gap files for free energy profile from state 1 to 2. When used within POLARIS, the state A and B are not needed, only the number of frames When used within MAPPING program the number of frames is not read, because only the identity of the resonance structures is of interest. Thus, in a MAPPING run the first number is irrelevant.

- GAS_DG

  USAGE: gas_dg 2 -5.0
  specifies the gas phase delta G for the given evb state (e.g. -5.0 for state 2).

- HIJ

  USAGE: Hij 1 2 6 9 -120 0.6
  specifies 2-atom off-diagonal elements (e.g. i=1, j=2, n1=6, n2=9, A=-120, mu=0.6 for the formula: Hij=A*exp(-mu*R_n1,n2)

- HR

  USAGE: Hr 1 2 6 4 9 -120 0.6
  specifies 3-atom off-diagonal elements (e.g. i=1, j=2, n1=6, n2=4,n3=9, A=-120, mu=0.6 for the formula: Hij=A*exp(-mu*(r_n1,n2-r_n2,n3))

- H3

  USAGE: H3 1 2 5 7 8 300 0.6
  specifies the 3-evbatom (Borgis) off-diagonal (e.g. i=1, j=2, l=5, m=7, n=8, A=300.0, alfa=0.6, beta=0.1 gama=90 angle0=180, for the formula: Hij=A*exp(-alfa*r_ln-beta*r_m_ln**2-gama*(angle-angle0)**2) where l,m,n are atom #s and r_m_lk is the distance between m and the center of l and n (i and j are resonance states) angle is formed by atoms l,m,n.

- R4

  USAGE: R4 1 2 1.44
  specifies the inductive interaction (e.g. i=1, j=2, alfa=1.44) for the formula: -166.0*alfa/(R_ij)*
  where i, j are atom #s.

- INDUCE_EVB

  USAGE: induce_evb 0
  0: no induce for evb atoms 1: induce for evb atoms, which activates the full iterative inductive treatment of the EVB system

- EXPONENTIAL

USAGE: exponential 1 2 2.5
specifies the exponential coefficient in nonbond formula (e.g. i=1, j=2, beta=2.5, for D*exp(-beta*R_ij) where i, j are atom #s, default beta=2.0

- VDW_PAIR

  USAGE: vdw_pair 2 5 OR vdw_pair 2 5 60. 760.
  specifies the evb atom #s to calculate nonbond interaction between them (e.g. atom 2 and 5) by the VDW formula (the default one is the exponential formula) in the second example: A(2)*A(5)=60.0, B(2)*B(5)=760.0 for the formula: A(2)*A(5)/r**12-B(2)*B(5)/r**6

- EL_CORREC

  USAGE: el_correc 2 5 3.0
  specifies special evb-evb nonbond electrostatic correction term for atoms 2 and 5 (e.g. i=2, j=5, mu=3.0 for formula: Q(i)*Q(j)*exp(-mu*R_ij**2)/R_ij)

- READ_EVB

  USAGE: read_evb evb.in
  reads evb atoms information from a specified file (e.g. evb.in) instead of using the keywords evb_atm and evb_bnd

- REST_IN

  USAGE: rest_in rest.in
  reads MD data from a specified restart file (e.g. rest.in)

- REST_OUT

  USAGE: rest_out rest.out
  writes MD data to a specified restart file (e.g. file is rest.out, the default restart file is rest.scratch) If EVB is being done, for example, the name specified by the keyword will be use as the root for all the restart files

- REST_STEP

  USAGE: rest_step 10 200 320
  writes MD data at specified steps (e.g. at step 10, 200 and 320) to restart files (e.g. rest_step.10, rest_step.200 and rest_step.320)

- AD_RESTOUT

  USAGE: ad_restout 1 2 5.0 20 30
  creates a series of restart files for subsequent downhill evb runs by specifying the evb states, E_alfa and # of files (e.g. 5.0 kcal/mol for state_a=1 and state_b=2, makes 20 restart files, maximum # of restart files is 20, # of MD steps between 2 restart files is at least 30). Restart files are created when E_alfa-1.0 < H(a,a)-H(b,b) < E_alfa+1.0

- REST_CONSTR

  USAGE: rest_constr 1
  use coordinates from restart for constraints (default is 0), should not be used for map_pf>1

- ENERGY_OUT

  USAGE: energy_out gap.out
  in EVB runs, writes energy data every 10 steps in a specified file (e.g. gap.out, this is the default energy output filename)

- CENTROID

  USAGE: centroid
  invokes the centroid level, which controls the QCP calculation used to add quantum corrections to the classical free energy by means of a path integral calculation.

- READ_LINK

  USAGE: read_link ab_link.dat
  specifies filename for quantum atoms and link atoms for an ab initio coupled EVB run

- CHECK_F

  USAGE: check_f 1
  calculates and writes out numerical and analytical force on evb or ac atoms, (default is 0, no numerical force calculation and writeout) this numerical and analytical forces are written out every 50 steps

- REG1_RES

  USAGE: reg1_res 2 or reg1_res 2 to 4 or reg1_res 2 4 reg1_res all
  assigns the given residue (e.g. the 2nd residue) to region I. (in the first example) assigns the given residue range (e.g. from residue 2 through 4) (in the 2nd example) assigns several individual residues (e.g. residue 2 and 4) (in the 3rd example) assigns all residues (in the 4th example)

- PKA_W

  USAGE: pka_w 3.0
  assigns for the specified residue the pKa in water, which overwrites the default value.

- PDLD_FN

  USAGE: pdld_fn asp.pdld
  specifies user-defined pdld input file name (e.g. asp.pdld), instead reg1_res to assign region I atoms

- REG1_ATM

  USAGE: reg1_atm 10 to 20
  specifies region I atoms (e.g. atom 10 through 20).

- AB_CRG

  USAGE: ab_crg 10 0.50 0.0
  assigns charges of 0.5 for the state A and 0.0 for the state B for atom 20 in ac calculation.

- REG2_R

  USAGE: reg2_r 16.0
  specifies the radius for region II. In PDLD the default is 18A. Do not use this keyword for normal calculation. this keyword is only for experienced users.

- RS_NUM

  USAGE: rs_num 2 5 8
  specifies the residue #'s (e.g. the residues 2, 5 and 8) for the intrinsic pKa calculations.

- CONFIG

  USAGE: config 0 5 or config 1 0 or config 1 5 or
  The first number tells if we want (1) or not (0) a calculation for the original structure. In the case we want it, only the H of this structure will be relaxed. The second number specifies the number of MD-generated configurations we want to include in the PDLD averaging.

- 7).eq.'MD_PARM

  USAGE: md_parm or md_parm_r or md_parm_w or md_parm_p or
  changes the MD parameters for the run. It is a general level, used by practically all the program. The _r _w and _p suffixes are used when different settings are requested by relaxation, water, and protein runs in PDLD calculation.

- TOP_IN

  USAGE: top_in test.top
  reads a topology file (e.g. test.top). The structure from this top file will overwrite the structure from the pdb file.

- ALLRES

  USAGE: allres
  lists the names and numbers for all the protein residues

- RESTYPE

  USAGE: restype ASP
  lists the residue numbers for all the residues of the specified type

- RESATOM

  USAGE: resatom 13
  lists all the atoms in the specified residue

- RESANG

  USAGE: resang 13
  lists all the angles in the specified residue

- RESTOR

  USAGE: restor 13
  lists all the torsions in the specified residue

- RESITOR

  USAGE: resitor 13
  lists all the improper torsions in the specified residue

- DISTATOM

  USAGE: distatom 234 456
  calculates the distance between the specified 2 atoms

- CHKBOND

  USAGE: chkbond 50.0
  finds out all bad bonds with bond energy > user-specified energy (e.g. > 50.0 kcal/mol)

- CENTER_S

  USAGE: center_s
  calculates the coordinates of the center of the system

- CENTER_R

  USAGE: center_r 5 12
  calculates the coordinates of the center of the given residues (e.g. residues 5 and 12)

- SPHEREION

  USAGE: sphereres 12.50 3.64 -6.28 10.63
  finds all the ionizable residues within a radius (e.g. 12.50A) from a center (e.g. 3.64 -6.28 10.63). Also writes a file ionres.dat in the output directory containing these data.

- SPHERERES

  USAGE: sphereres 12.50 3.64 -6.28 10.63
  finds all the residues within a radius (e.g. 12.50A) from a center (e.g. 3.64 -6.28 10.63).

- ADDBOND

  USAGE: addbond 2 5 9 10 18
  add new bonds between the specified atoms (e.g. add 2 new bonds between atoms 5 and 9, 10 and 18)

- VIEWMOVIE

  USAGE: viewmovie
  makes coordinate files for viewing structures along an MD run. Requires previous generation of a movie.dat file. See the md_parm level in POLARIS and ENZYMX.

- VIEWPOT

  USAGE: viewpot
  makes files for viewing the electrostatic potential surface around region I atoms. See the set_opt level in POLARIS/PRE_POL

- MAKEPDB

  USAGE: makepdb
  makes a new pdb file for atoms with updated coordinates.

- MAKELIB1

  USAGE: makelib1
  makes a new entry in the amino library for a new molecule

- DOCK

  USAGE: dock
  docks or superimposes one structure to another structure

- VDWSURF

  USAGE: vdwsurf
  makes a file with the vdw surface of the protein. Requires the user to have MSMS installed in the system.

- RESBOND

  USAGE: resbond 13
  lists all the bonds in the specified residue (e.g in residue 13)

- ANALYZE

  USAGE: analyze
  analyzing structures of macromolecules

- POLARIS

  USAGE: polaris
  using the PDLD approach for studies of molecules (pKa, REDOX, binding, solvation energies...

- ENZYMX

  USAGE: enzymix
  simulating properties of proteins using the surface constrained all-atom protein/solvent model.

- PRE_ENZ'.or.string.eq.'3

  USAGE: pre_enz
  preparing for the enzymix run (setting new charges for atoms, ionizing residues, reading restart file, ... etc.)

- EVB'.or.string.eq.'3

  USAGE: evb
  simulating enzymatic reaction by the EVB method

- EVB2'.or.string.eq.'3

  USAGE: evb2
  simulating enzymatic reaction by the EVB method with changes of the EVB definition during the MD run, having a set of subruns for these different EVB definitions.

- AC'.or.string.eq.'3

  USAGE: ac
  evaluating solvation free energies by the adiabatic charging FEP approach

- RELAX'.or.string.eq.'3

  USAGE: relax
  MD simulations of the whole protein system relaxation

- EVB_AB'.or.string.eq.'3

  USAGE: evb_ab
  adding the protein/solute interaction (solvation energies) to ab initio potential surface, creating trajectory files for ab initio programs.

- ADIAB_POT'.or.string.eq.'3

  USAGE: adiab_pot
  simulating actual EVB adiabatic potential surface

- ADIAB_TEM'.or.string.eq.'3

  USAGE: adiab_tem
  simulating actual EVB adiabatic potential surface with MD running at constant temperature

- PRE_POL

  USAGE: pre_pol
  preparing polaris run

- SOLV_PDLD

  USAGE: solv_pdld
  calculating solvation energies in water and protein using the PDLD approach

- SOLV_FEP

  USAGE: solv_fep
  calculating solvation energies in water and protein using the FEP approach

- AI_PDLD

  USAGE: ai_pdld
  adding the protein/solute interaction energies to the ab initio quantum mechanical region in QM/PDLD calculation

- BIND_PDLD

  USAGE: bind_pdld
  calculating binding free energies using the PDLD/S, LRA and LIE approaches

- BIND_FEP

  USAGE: bind_fep
  calculating binding energies using the FEP approach

- PKA_PDLD

  USAGE: pka_pdld
  calculating pKa of ionizable groups in protein using the PDLD and related approaches

- PKA_FEP

  USAGE: pka_fep
  calculating pKa of ionizable groups in protein using the FEP approach

- REDOX_PDLD

  USAGE: redox_pdld
  calculating REDOX potential in protein using the PDLD and related approaches

- REDOX_FEP

  USAGE: redox_fep
  calculating REDOX potential in protein using the FEP approach

- LOGP

  USAGE: logp
  calculating log P for molecules in solutions

- TITRA_PH_0

  USAGE: titra_ph_0
  calculating pH titration curves with pre-assigned intrinsic pKa

- TITRA_PH

  USAGE: titra_ph
  calculating pH titration curves with multi_pKa results

- PKA_MULTI

  USAGE: pka_multi
  a fast estimate of the intrinsic pKa of all ionizable residues

- EVB_PDLD

  USAGE: evb_pdld
  calculating activation energies of enzymatic reactions using structures generated by the EVB and the PDLD/LRA approaches

- IONRES

  USAGE: ionres 3
  ionizes the given residue (e.g. residue 3)

- UNIONRES

  USAGE: unionres 3
  unionizes the given residue (e.g. residue 3)

- IONTYP

  USAGE: iontyp ASP
  ionizes all the residue of the specified type (e.g. type ASP)

- SETCRG

  USAGE: setcrg 1 0.50
  sets a new charge for the specified atom (e.g. charge 0.5 for atom 1)

- SETCRG0

  USAGE: setcrg0 2 OR

  USAGE: setcrg0 1 to 10
  sets charge 0.0 for every atom in the specified residue(s) (e.g. 0.0 for every atom in residue 2 in the 1st example 0.0 for every atom in residue 1 through 10 in the exmaple)

- RG

  USAGE: rg 18.5
  sets the radius of the langevin grid (e.g. 18.5A)

- OUTER

  USAGE: outer 3.0
  the spacing of the outer langevin grid (e.g. 3.0A)

- INNER

  USAGE: inner 1.0
  the spacing of the inner langevin grid (e.g. 1.0A)

- RDCUTL

  USAGE: rdcutl 6.0
  from 6.0 up, the cutoff radius for the interactions between the Langevin dipoles (the rest of the system is treated by the LRF approach)

- RDCUTP

  USAGE: rdcutp 10.0
  from 10.0 up, the cutoff radius for the interactions between the protein dipoles in the PDLD SCI calculations

- MICRO

  USAGE: micro 0
  0: semimicroscopic PDLD/S 1: microscopic PDLD/M

- LOG

  USAGE: log 0
  0: gives a long output file 1: gives a short output file

- LGVN

  USAGE: lgvn 0
  0: uses linear polarization function 1: uses langevin function

- NDXP

  USAGE: ndxp 30
  from 1 up, # of sets of different langevin grids generated by randomly moving around the centers of the grids and calculating the non-iterative langevin energies

- NCENTER

  USAGE: ncenter 3
  1-5, # of sets of different langevin grids with best centers (lowest langevin energies calcualetd non-iteratively) to be used for iterative langevin energy calculations

- ITL

  USAGE: itl 30
  from 10 up, # of langevin SCI iterations for each new grid

- ITP

  USAGE: itp 3
  1-5, # of protein SCI iterations

- FIELD

  USAGE: field 0
  1: creates an output file: field.dat, which contains the electric field data at the of all the protein atoms and langevin dipoles 0: no field created

- POTENT

  USAGE: potent 0
  0: no electrostatic potential created 1: creates an output file: map_pot.dat, which contains the electric potential for a grid around the sites of all the region I atoms 2: creates output files: epot_a.out and epot_b.out, which are in pdb format and have average electrostatic potential data at each region I atom site 3: do 1+2

- IREF

  USAGE: iref 1
  1: performs reference run of region I in water 0: no reference run of region I in water

- ICUT

  USAGE: icut 1
  1: uses the cutoff rdcutl and LRF 0: no cutoffs for the interaction between the dipoles

- RP
  USAGE: rp yes rp 40 2.30
  the closest distances between a grid dipole and the indicated protein atom. In the above example: sets 2.30A as the rp value for the new atom whose atom type is 40

atom type	rp	atom type in library
1	1.50	hydrogen: H1, H2
2	2.00	oxygen: O1, O2
3	2.20	nitrogen: N2, N3, N4
4	2.40	sp3 carbon: C4
5	2.30	sp2 carbon: C3
6	3.20	phosphorous: P4
7	3.10	sulfur: S2
8	2.12	sodium: NA
9	2.23	calsium: CA
10	1.88	magnesium: MG
11	1.81	aluminium: AL
12	1.58	arsenic: AS
13	1.35	boron: B2
14	1.23	boron: B3
15	2.60	bromine: BR
16	2.00	fluorine: F1
17	1.10	gallium: GA
18	1.60	germanium: GE
19	2.90	iodine: I1
20	1.81	indium: IN
21	1.67	ruthenium: RU
22	1.76	antimony: SB
23	1.60	selenium: SE
24	2.80	silicon: SI
25	1.80	tin: SN
26	2.00	technetium: TC
27	3.10	tellurium: TE
28	2.00	titanium: TI
29	1.42	zinc: ZN
30	2.52	chlorine: CL
31	1.10	iron: FE
32	1.50	hydrogen in water: WH
33	2.50	nitrgen ion: N+

Adding a new rp: Do not change the above rp values, instead to add rp for a new type of atom, use keywords: pdld_atom and rp

- PDLD_ATOM

  USAGE: pdld_atom 20
  finds out the atom PDLD type for the specified atom in the system (e.g. atom 20)

- SET_OPT

  USAGE: set_opt
  changes default parameter values for the PDLD calculations

- RESIDUE

  USAGE: residue 2 or residue 2 4 or residue all or residue all+w or residue 2 to 5
  residues for which the new PDB file will be created in the makepdb sublevel

- FILE_NM

  USAGE: file_nm file.extension
  specifies the name of a file to be created in the given level or from which to read data. For example, in a DOCK calculation, it is the output PDB file In a VIEWMOVIE run, it is used to specify the name of the movie.dat file, the binary created in a previous MD run containing the information of the trajectory.

- AB_VDW

  USAGE: ab_vdw 20 H1 DH
  assigns vdw type of H1 for the state A and DH for the state B of the 20th atom.

- READ_AC

  USAGE: read_ac file.ac
  reads the charges and VDW types from the specified file (e.g. file.ac) for ac atoms, instead of using the above keywords

- MAP_LAMBDA

  USAGE: map_lambda 1.0
  AC mapping parameter from state A to B (e.g. at 100

- MAP_PF

  USAGE: map_pf 1 or map_pf 11 1 2
  specifies the # of the steps for MD simulations to map free energy profile (eg, just 1 file or 11 files from state 1 to 2).

- RESTART1

  USAGE: restart1 rest1.in
  specifies the restart file name created in the previous EVB run in an EVB_PDLD calculation

- RESTART2

  USAGE: restart2 rest2.in
  specifies the restart file name created in the previous EVB run

- REDOX_W

  USAGE: redox_w 200
  specifies the redox potential E(O/R) for regin I in water (e.g. E(O/R) = 200 mv for region I from oxidzed state to reduced state.

- PRE_AB

  USAGE: pre_ab
  preparation of the input file for the ab initio program

- RESPKA

  USAGE: respka 10 3.0
  assigns for the specified residue the pKa in the protein, which overwrites the default value (the value in water).

- TYPPKA

  USAGE: typpka SER 14.5
  assigns a intrinsic pKa value to all the residues of a given type (e.g. pintrinsic Ka=14.5 for residue type SER)

- CALPKA

  USAGE: calpka
  calculates intrinsic pKa values for residues to be used in titration

- TITRA_R

  USAGE: titra_r 4 20.0
  specifies the center residue and radius from it. Titration curves are calculated for all the ionizable residues inside the sphere with that radius. (e.g. 20.0A from residue 4)

- MONTE_R

  USAGE: monte_r 10.0
  specifies the radius for the Monte Carlo evaluation. The first 10 ionizable residues within mont_r will be used by Monte Carlo approach, default monte_r=20.0A

- LIST_INT

  USAGE: list_int
  lists the intrinsic pKa to be used in the calculation

- DIELECT

  USAGE: dielect 40.0
  assigns the effective dielectric for charge-charge interactions

- PH_FOCUS

  USAGE: ph_focus 1 4 0.2
  specifies the pH range and the pH spacing for the titration (e.g. pH is from 1.0 to 4.0, and the pH spacing = 0.20, the default range is 0 to 20, pH spacing = 0.5)

- TEMP'.or.string.eq.'TEMPERATURE

  USAGE: temperature 300.0
  temperature in K

- NSTEPS

  USAGE: nsteps 2000
  number of steps in the MD, when used in md_parm number of steps in the titration, when used in titra_pH (in this case, the default value=30, maxmum=50)

- PDB2

  USAGE: pdb2 mole2.pdb
  specifies pdb file path and name (e.g. mole.pdb) for the structure to be docked or superimposed to the original structure

- RES_IN_1

  USAGE: res_in_1 3
  specifies the residue # in the original structure to be superimposed to by the same residue in the 2nd structure

- RES_IN_2

  USAGE: res_in_2 3
  specifies the residue # in structure 2 (e.g. residue 3) for superimposing it to the same residue in the original structures. Note: res_in_1 and res_in_2 could have different values, because of the sequence # of the residues, but they have to refer to the same residues in the 2 structures of the same molecule

- VIEW_FQ

  USAGE: view_fq 20
  viewing the structures at every 20 MD steps

- VWALL

  USAGE: vwall
  viewing all the atoms in the movie.dat file

- VWRES

  USAGE: vwres 2 or vwres 2 5 or vwres 2 to 20 or vwres rg1 or vwres wat
  creates a movie for the specified residue(s) This can include region I atoms (rg1) or even the water molecules if they were saved (wat)

- FORM

  USAGE: form pdb or form AMBER or form CHARMM or form XMOL
  makes a series of files in pdb, or makes trajectory files in AMBER, CHARMM, or XMOL formats

- FRAME

  USAGE: frame 40
  analyzing the structure of the specified frame (e.g. the structure at the 40th MD step)

- TOLERANCE_TEMP

  USAGE: tolerance_temp 3000.
  the high temparature (K) allowed for the MD to fluctuate up to and write out file: restart.crash. then runs automatic smaller stepsize relaxation

- STEPSIZE

  USAGE: stepsize 0.001
  the time (in ps) interval for the MD run

- NBUPDATE

  USAGE: nbupdate 30
  the number of the steps for updating the nonbond interaction pairlists

- GAS_PHASE

  USAGE: gas_phase 1
  1: MD run in gas phase, program automatically sets water_r and langevin_r to 0.0

- REGION2A_R

  USAGE: region2a_r 18.0
  16.0 - 24.0, the radius for region II

- WATER_R

  USAGE: water_r 18.0
  0 or =>region2a_r, the radius of water grid

- LANGEVIN_R

  USAGE: langevin_r 20.0
  the radius of langevin grid

- EX_W_CENTER

  USAGE: ex_w_center 3.0 4.5 2.34
  the exclusion and water grid center coordinates (the default center is the center of region I, which is automatically calculated. This keyword is for experienced users)

- INDUCE

  USAGE: induce 0
  0: no induce energy 1: induce for protein 2: induce for protein+water

- INDFORCE

  USAGE: indforce 0
  0: no induce force 1: induce force is used if induce is not 0 10: no induce force. Only update induce energies every 10 steps

- CONSTRAINT_1

  USAGE: constraint_1 0.03
  constraint for region I atoms

- CONSTRAINT_2

  USAGE: constraint_2 0.03
  constraint for region II atoms

- CONSTRAINT_PAIR

  USAGE: constraint_pair 10 34 10.0 1.3
  constrain a distance for a pair of atoms, the numbers are: atom1 atom2 force distance

- CONSTRAINT_ANG

  USAGE: constraint_ang 10 34 35 10.0 120.
  constrain an angle for 3 atoms, the numbers are: atom1 atom2 atom3 force angle

- CONSTRAINT_POST

  USAGE: constraint_post 10 10. 10. 10. 3.4 -4.6 4.7
  constrain one atom at a fixed position, the numbers are atom force_X force_Y force_Z X Y Z

- H_CONSTRAIN

  USAGE: h_constraint 0
  0: do not constraint H atoms in region I 1: do constraint H atoms in region I

- CUTPP

  USAGE: cutpp 10
  the cutoff radius for protein-protein interactions The default value is optimum, do not change

- CUTPW

  USAGE: cutpw 10
  the cutoff radius for protein-water interactions The default value is optimum, do not change

- CUTWW

  USAGE: cutww 10
  the cutoff radius for water-water interactions The default value is optimum, do not change

- MOVIE_CO

  USAGE: movie_co 0 or movie_co 20 or movie_co 1 to 20 or movie_co rg1 or movie_co all or movie_co wat or
  writes out the coordinates for the specified residues in a file called movie.dat. This binary file can be later processed using the viewmovie keyword in ANALYZE. In this context, ALL means all the residues (not wat).

- MOVIE_FQ

  USAGE: movie_fq 10
  from 10 up, out the coordinates to the movie file: movie.dat eg, 20: writes out the coordinates every 20 steps

- ACF_WIDTH

  USAGE: acf_width 5.
  width, in ps, of the autocorrelation plot: from C(0) to C(autocorr_wide)

- HRGAUSS

  USAGE: Hrgauss 1 2 6 4 9 -120 0.6
  the same than Hr, but with a Gaussian expression. specifies 3-atom off-diagonal elements (e.g. i=1, j=2, n1=6, n2=4,n3=9, A=-120, mu=0.6 for the formula: Hij=A*exp(-mu*(r_n1,n2-r_n2,n3)**2)

- HDIRECT

  USAGE: Hdirect 1 2 30.
  Value of A for the off-diagonal term of the secular matrix. the Hij read from the GAP file is multiplied by a constant (30.)

- HGAP

  USAGE: Hgap 1 2 30. 2.5
  Value of A for the off-diagonal term of the secular matrix. GAP dependent off-diagonal term.

- NUM_BIN

  USAGE: num_bin 20
  Number of bins in the umbrella sampling

- POINTS_THROW

  USAGE: points_throw 10
  Number of points to trow off before consider them valids for the umbrella sampling compilation.

- BEADS

  USAGE: beads 20
  in a type QCP calculation, the number of quasiparticles used for the centroid approach mapping

- REORG_ENERGY

  USAGE: reorg_energy 240.
  in a type AUTOCORR calculations, the reorganization energy is needed for normalizing the power spectra obtained We suggest to run first a regular EVB mapping and from it obtain the value of reorg_ener.

- REORG_ENERGY_IN

  USAGE: reorg_energy_in 150.
  in a type AUTOCORR calculations, the reorganization energy is needed for normalizing the power spectra obtained This value is used for the internal spectra.

- REORG_ENERGY_EL

  USAGE: reorg_energy_el 90.
  in a type AUTOCORR calculations, the reorganization energy is needed for normalizing the power spectra obtained This value is used for the solvent spectra.

- TOLERANCE

  USAGE: tolerance 3
  When doing the umbrella sampling, minimum number of points in a bin for this to be considered statistically signifficant

- FILEROOT

  USAGE: fileroot /tmp/map_evb.gap 12
  root of the filename of the gap files. The number indicates how many gap files we are taking into account.

- FILEGROOT

  USAGE: filegroot graph_out
  root of the filename of the graph output file. In this case it would be graph_out.graph the default is to add .graph to fileroot

- SINGLEGAP

  USAGE: singlegap /tmp/map_gap.out
  calculation is performed on a single gap file with the name indicated by the singlegap value. This option is useful for mapping_type AUTOCORR when just a single gap file is being processed. Also useful for regular mapping calcs where the numbering of the files is not so automatic as to use the fileroot keyword

- MAPPING_TYPE

  USAGE: mapping_type AUTOCORR
  calculates autocorrelation functions for the given files / USAGE: mapping_type AC
  mapping for the adiabatic charging process. / USAGE: mapping_type EVB
  mapping for getting the free energy profile on the ground state potential for a reaction studied by the EVB method / USAGE: mapping_type ENTROPY

mapping from an entropy calculation / USAGE: mapping_type PMF
mapping for a PMF calculation (under development) / USAGE: mapping_type QCP
mapping for a quantum classical path (centroid) calculation for the inclusion of quantum effects into an EVB calculation. MAPPING_TYPE=EVB is implicitly included in this case

- ELECTROSTATIC_ONLY

  USAGE: electrostatic_only
  The mapping will be done only for electrostatic contributions

- INCLUDE_INTRA

  USAGE: include_intra
  Flag for use also the intramolecular energy when performing an AC calculation

- FL_BAR_PARM

  USAGE: fl_bar_parm 30. 1.0 1.2
  parameters for the universal EVB The first number is the K_lambda. The next values are r_1^0 and r_2^0, respectively. This calculation allows the plot of a fluctuating Eg barrier. (under development)

- FILEQROOT

  USAGE: fileqroot /tmp/map_qgap.out 12
  the root part of the filename of the quantum gap files in a type QCP run. The number indicates how many gap files we are taking into account.

- ADD_ANG

  USAGE: add_ang 2 5 8 200 120 2
  adds a new evb angle for evb atoms with specified force constant and equilibrium angle for some specified evb states (e.g. evb angle formed by evb atoms 2, 5 and 8 with force constant = 200 and equilibrium angle = 120 degree for evb state 2)

- EVB_ENTROPY

  USAGE: evb_entropy 1
  specifies the part of the system under evb/entropy constraint (e.g. for region I)

- SCRIPT

  USAGE: script qmmm.spt
  specifies the script file name (e.g. qmmm.spt) for running a stand-along quantum program. MOLARIS program will write out coordinates for region I, protein and water atoms to a a file. At this point MOLARIS program pauses, while quantum program starts to run and reads in data from the file. When quantum program finishes running, the forces for region I atoms are written to a file, and MOLARIS resumes running and reads in the force from the file for region I atoms

- START_STEP

  USAGE: start_step 100
  specifies the starting MD step (e.g. 100th step, the default is at the 10th step), at which the quantum program starts the first run

- Q_STETPSIZE

  USAGE: q_stepsize 20
  specifies the step interval (e.g. every 20 steps, the default is 10). After the first run, the quantum progran will run once every this interval

- QMMM_IN

  USAGE: qmmm_in ATOM.IN
  specifies the file name (e.g. ATOM.IN, which is the default) for MOLARIS to write out coordinates for region I, protein and water atoms and for quantum program to read in. The file format is as the following: structure ID number (free formatted) # of region I atoms (free formatted) atom_symbol, x,y,z,charge (a2,4f15.9) # of region I‘ frozen atoms, # of groups (free formatted) group #, # of frozen atoms, # of link atoms in group 1 (free) atom_symbol, x,y,z,charge in frozen group 1 (a2,4f15.9) ..... <for other frozen groups> # of region I‘ frozen water atoms (free formatted) atom_symbol, x,y,z,charge (a2,4f15.9) # of region II atoms (free formatted) atom_symbol, x,y,z,charge (a2,4f15.9) # of water atoms (free formatted) atom_symbol, x,y,z,charge (a2,4f15.9)

- QMMM_OUT

  USAGE: qmmm_out FORCE.OUT
  specifies the file name (e.g. FORCE.OUT, which is the default) for quantum program to write out the total quantum energy, forces and charges of region I atoms for MOLARIS to read in. The file format should be: total_energy force_x, force_y, force_z (for region I atom 1) force_x, force_y, force_z (for region I atom 2) ... force_x, force_y, force_z (for region I‘ frozen atom 1) force_x, force_y, force_z (for region I‘ frozen atom 2) ... charge (for region I atom 1) charge (for region I atom 2) ... (all records are free formatted)

- QMMM

  USAGE: qmmm
  simulating properties of proteins using the surface constrained all-atom protein/solvent model, with the forces for region I atoms at each specified MD step calculated from a quantum program

- PMF

  USAGE: pmf
  evaluating the potential of mean force (pmf) of the system by gradually moving region I atoms from initial positions to final positions. This is done by performing 11 steps with the mapping parameter theta(m)_pmf step

- INIT_FN

  USAGE: init_fn init.dat
  coordinates file name (e.g. init.dat) for region I atoms to provide the initial positions. The format is as the following: 2 0.2092 -3.0658 -3.0454 7 -0.4605 -3.2021 -2.3022 ... (where in each line, the first number is the atom #, followed by coordinates x, y, z. All are free formatted)

- FINAL_FN

  USAGE: final_fn final.dat
  coordinates file name (e.g. final.dat) for region I atoms to provide the finale positions. The format is as the following: 2 -5.8032 3.0998 3.0513 7 -5.9208 2.9438 2.0609 ... (where in each line, the first number is the atom #, followed by coordinates x, y, z. All are free formatted)

- ELECTRO

  USAGE: electro 1 18.0 4
  specifies the number of residues, radius and residue numbers (e.g. one residue, radius=18A and that residue is the residue 4) for calculating the electrostatic interaction of these residues with all the other residues inside the radius

- MAP_PMF

  USAGE: map_pmf 11
  specifies the # of the steps for PMF simulations to map free energy profile (e.g. 11 steps with stepsize of 0.1). 11 and 21 are good choices

- REGION1_RES

  USAGE: region1_res 2 3
  specifies the residues in fdft region 1 (e.g. residues 2 and 3)

- REGION1_ATM

  USAGE: region1_atm 10 to 20
  specifies the atom in fdft region 1 (e.g. atoms from 10 through 20)

- FROZEN_GRP_ATM

  USAGE: frozen_grp_atm 1 25 to 40
  specifies the atoms in fdft frozen region and group (e.g. atoms from 25 through 40 are in group 1 of fdft frozen region)

- LINK_ATM

  USAGE: link_atm 4 3
  specifies the link atom in fdft region 1 (e.g. protein atom 4, which is connected to atom 3, will will be changed to link atom H with new right coordinates)

- FDFT_IN

  USAGE: fdft_in ATOM.IN
  specifies the file name (e.g. ATOM.IN, which is the default) for MOLARIS to write out coordinates for full fdft region, frozen fdft region, region II and water atoms, for fdft program to read in. The file format is as the following: energy_shift configuration# classical_energy # of_fdft_atoms total_charge spin_multiplicity atom_symbol, x,y,z,charge (a2,4f15.9) # of groups in fdft frozen region number_of_atoms # of atoms in group 1 total_charge 1 atom_symbol, x,y,z,charge (a2,4f15.9) <for group 2> . . . # of water atoms within fdft radius atom_symbol, x,y,z,charge (a2,4f15.9) # of protein atoms and water atoms within region II atom_symbol, x,y,z,charge (a2,4f15.9)

- FDFT_OUT

  USAGE: fdft_out FORCE.OUT
  specifies the file name (e.g. FORCE.OUT, which is the default) for fdft program to write out the total quantum energy, and charges of region 1 atoms for MOLARIS to read in. The file format should be: total_energy charge (for region I atom 1) charge (for region I atom 2) ... (all records are free formatted)

- FDFT_WAT

  USAGE: fdft_wat 12.5 3.64 -6.28 10.63
  specifies the radius and the center for water molecules, all the water molecules within that sphere will be a part of the fdft system. (e.g. radius = 12.5A, center coordinate = 3.64 -6.28 10.63 the default radius and center are the same as that specified with the last keyword sphereres_, respectively)

- FD_STETPSIZE

  USAGE: fd_stepsize 20
  specifies the step interval (e.g. every 20 steps, the default is 10). After the first run, the quantum progran will run once every this interval

- SPHERERES_R

  USAGE: sphereres_r 12.50 6
  finds all the residues within a radius (e.g. 12.50A) from the center of a residue (e.g. residue 6).

- SPHERERES_A

  USAGE: sphereres_a 12.50 210
  finds all the residues within a radius (e.g. 12.50A) from an atom (e.g. atom 210).

- FIX_REGION

  USAGE: fix_region 1
  fixes the positions of atoms in the specified region (e.g. all atoms in region I) by using the original coordinates fix_region 1 for fixing region I, fix_region 2 for fixing region II fix_region 3 for fixing waters fix_region 4 for fixing region II + waters

- FIX_ATOM

  USAGE: fix_atom 8
  fixes the position of the specified atom (e.g. atom 8) by using the original coordinates

- STEEP_MINI

  USAGE: steep_mini 1
  specifying the option for minimizing the structure of the system using the approach of the steepest descent: 1: using steepest descent to minimize the system 0: without minimizing by the steepest descent (default)

- SP_MULTIPLICITY

  USAGE: sp_multiplicity 1
  specifying electron spin multiplicity for the system in fdft calculation

- ELEVEL_SHIFT

  USAGE: elevel_shift 5.0
  specifying quantum energy level shift for fdft calculation (e.g. 5.0 kcal/mol)

- DF_MINI

  USAGE: df_mini 1 0.0001
  specifying the option for minimizing the structure of the system using the approach of davidon-fletcher and the tolerance for converge (e.g. tolerance=0.0001) 1: using davidon-fletcher approach to minimize the system 0: without minimizing by DF approach (default)

- TITRA_N

  USAGE: titra_n 10 13 17
  specifies the residues for which perform the titration (further versions will improve this selection, including a given type of residue, for example)

- EVB_P_CON

  USAGE: evb_p_con 2 10.0 10.0 10.0 6.363 2.010 -0.367 1 specifies the position constraints for evb atoms (e.g. for atoms 2, force_x=10.0 force_y=10.0 force_z=10.0 x=6.363 y=2.010 z=-0.367 for evb state 1). The constraint energy is added to the energy of evb state 1 and energy output file

- OPT_HIS

  USAGE: opt_his 1 optimizes the histidine form (hie or his) depending on which form has the lowest electrostatic energy

- IONIC

  USAGE: ionic 1 0.5
  specifies that the ionic strength effect is calculated and the value for ionic concentration (i.g. 1: activated, default is 0: not used. And ion concentration 0.5M, which is the default)

- MUTATE_RES

  USAGE: mutate_res 2 SER
  mutates one residue to another type residue and put it at the end of the topology file. The unmutated and mutated are both with the same sequential connections. To do the MD calculation for mutation, in ac.dat file the mutated one should be all dummy (all charges 0 and vdw 0) for state A, and the unmutated should be all dummy for state B.
  note: after mutate_res, if you make a pdb file and read it in with molaris again, the sequential connections this time would not be right for mutation. The new pdb file can only be used to check the structure for mutation with some graphic program

- LOG_DETAIL

  USAGE: log_detail 1
  specifies that the log file has detailed output (for gas phase and long md run, the default is short output file)

- EVB_SIMP

  USAGE: evb_simp 1 18.0 6
  specifies that the md is for simple evb, the radius and the # of langevin grid sets (e.g. 1: run simple evb, default is 0, no simple evb; 18A for langevin grid sphere; langevin solvation energy averaged over 6 different grid sets)

- R_FROZEN_W

  USAGE: r_frozen_w 6
  specifies a frozen water atoms layer outside qmmm atoms, these water molecule are within the specified radius (e.g. radius of 6A from the center of region I)

- ROTATE_H

  USAGE: rotate_h 3 36
  specifies the residue # and heavy atom # (e.g. residue 3 and atom 36) for rotating the H atoms bonded to that heavy atom to get a configuration with lowest energy

- PDLD_CENTER

  USAGE: pdld_center 12.5 2.3 8,9
  specifies the pdld grid center for pdld calculations. The default is the center of the pdld region I atoms, which is calculated automatically by MOLARIS

- FROZEN_GP

  USAGE: frozen_gp 3 7 9 10 11
  specifies frozen protein atoms in a frozen group (e.g. atoms 3 7 9 10 11 are in the same group

- REST_FQ

USAGE: rest_fq 100
specifies step interval for writing a restart file (e.g. every 100 MD steps to write a restart file)

- PMF_WRITE

  USAGE: pmf_write 10
  specifies step interval for writing the pmf energies to gap file (e.g. every 10 MD steps to write to gap file)

- CONSTRAINT_W

  USAGE: constraint_w 30.0
  constraint for water solvent atoms

- DIST_ATOMS

  USAGE: dist_atoms 2 5
  calculates the distance between atom pairs (e.g. atoms 2 and 5) writes out the distance in the file dist.dat, the data in it are: step, atom1, atom2, distance, electrostatic energy

- DIST_WRITE_FQ

  USAGE: dist_write_fq 10
  writes out the distance between user-specified atoms in the file dist.dat with the specified freqency (e.g. every 10 steps)

- CONSTRAINT_R

  USAGE: constraint_r 5 10.0 50.0 2.0 4.6 7.3
  specifies the distance constraint for an atom from a point (e.g. for atom 5, with constraint force of 50.0 and distance of 10.0A from a point (2.0 4.6 7.3))

- REST_TO_PDB

  USAGE: rest_to_pdb rest.pdb
  specifies the pdb file name, which will be created using the coordinates from a rest file for all the atoms in original pdb file and molaris-generated water atoms (e.g. the new pdb file name is "rest.pdb")

- AVE_ELEC

  USAGE: ave_elec 500
  calculating the average electrostatic energy contribution of each residue to each avb resenant state over every 10 MD steps and write out the averages at the specified interval (e.g. write it out every 500 steps. The default is 0, no calculation and write-out of this average)

- END

  USAGE: end
  exit the current level and starts the calculation

- EXIT

  USAGE: exit
  exit the current level without doing anything (rejecting changes)

- HELP

  USAGE: help OR

  USAGE: help <keyword1> <keyword2>...
  shows the table of contents of the current level or gives help on a particular keyword or series of keywords

Chapter 3

Apendices

3.1 The coordinates file formats

MOLARIS accepts entries in the PDB format and in MOL2 format, or files where a combination of the two formats is present.

When using the makelib1 keyword in the analyze task, another free format is also accepted. The **makelib1** command reads all necessary information about the new molecule from a free formatted x,y,z coordinate file, adds a new entry to an existing library file and creates a PDB file for the new molecule. This PDB file can be read using the **readbrk** command. The file format is explained using CH4 as an example

```
1 CH CH4 .000 .000 .000 -0.160 4 2 3 4 5
2 H1 CH4 1.043 .000 .000 0.040 1 1
3 H2 CH4 -.369 1.040 .000 0.040 1 1
4 H3 CH4 -.367 -.521 -.901 0.040 1 1
5 H4 CH4 -.367 -.521 .901 0.040 1 1
```

There is one line per atom and the data is separated by spaces. Each line contains the atom number, atom name, molecule name, the x,y,z-coordinates of the atom, the atomic charge, the number of atoms connected to this atom and the atom numbers of these bonded atoms. The atom name can be any combination of two to four upper case letters and/or numbers. The first symbol, however, has to be a letter and the names for all atoms must be different. The molecule name is a three letter code (upper case letters and/or numbers), which will also be the name of the new library entry.

3.2 MOLARIS libraries

3.2.1 The Amino Acid Library File Format (amino00.lib)

MOLARIS requires that the user provides a library entry for all residues. The library amino98.lib that is provided with the program has the templates for all the usual amino acids along with a few substrates and small molecules commonly encountered in biological systems (*e.g.* H2O, Na+, Ca++, etc.). If the user has a substrate that is not represented in the library (this is a common event) a new entry must be created. Currently MOLARIS

does this automatically once the PDB file is read, but in this section we describe the format of the amino acid library, which the user is encouraged to check for errors once the library has been updated with our new residues. Because of the sometimes difficult task of the automatic generation of entries in the library this has been proved to be a source of errors in previous runs with the program. These errors are easily avoidable by a careful checking of the library. Each residue in the library has an entry similar to the following:

```
-------------------------------------
 1GLY
   7                    Number of atoms
   1 N      N3  -0.400 Atom number, name, type, charge
   2 HN     H2   0.400
   3 CA     C4  -0.194
   4 HA1    H1   0.097
   5 HA2    H1   0.097
   6 C      C3   0.550
   7 O      O1  -0.550
   6                    Number of bonds
   1    2
   1    3
   3    4
   3    5
   3    6
   6    7
   1    6
   3                    Number of electroneutral groups (EN)
   2    1   3.000
   1    2
   3    3   3.000
   3    4    5
   2    6   3.000
   6    7
   0
-------------------------------------
```

The file is read in a formatted manner so the positions of the numbers are important. The sections of the file are as follows:

- **Library entry number and name** - an I3 format number (which should be one more than the number of the previous library entry) and a A3 format name for the residue. This will be the same as the name found in the PDB (Brookhaven) file.
- **Number of atoms** - an I5 format number specifying the number of atoms in the residue.
- **Atom names and ID's** - For each atom in the residue there should be a line with the following information:
  - An I5 format number that is the atomŠs number in the residue
  - One space
  - An A4 format name that is the atom name in the PDB (Brookhaven) file. For hydrogens the name may not be in the PDB (Brookhaven) file. ENZYMIX adds the missing hydrogens according to the amino acid library entries.
  - Three spaces
  - An A2 format letter code that is the atom type. Atom type codes are:

H1	Hydrogen bound to carbon
H2	Hydrogen bound to oxygen or nitrogen
H3	Hydrogen bound to nitrogen in DNA
C3	Carbon bound to three atoms (sp2 Carbon)
C4	Carbon bound to four atoms (sp3 Carbon)
O1	Oxygen bound to one atom (carbonyl oxygen)
O2	Oxygen bound to two atoms (alcoholic oxygen)
N2	Nitrogen bound to two atoms
N3	Nitrogen bound to three atoms
N4	Nitrogen bound to four atoms (eg. -NH4+)
S2	Sulfur bound to two atoms
P4	phosphorous bound to four atoms
CA	Calcium
MG	Magnesium

Note: If you need other special atom types, for example for K+, and Zn++ you have to make sure that the parameters for these types exist in the parameter library parm.lib. See section **??**.

  - An F8.3 number that is the atomic charge on each atom.

- **Number of bonds** - An I5 format number that gives the number of bonds in the residue.
- **Bond descriptions** - For each bond in the residue there is a line with the following information:
  - An I5 number that is the atom number of the first atom in the bond
  - An I5 number that is the atom number of the second atom in the bond
- **Begin and end atoms** - this is one line that has an I5 format number giving the begin atom number and an I5 number giving the end atom number. The begin atom is the atom that gets attached to the previous residue and the end atom is the atom that gets attached to the following residue. If the residue is isolated both numbers have to be 0.
- **Number of Electroneutral groups** - The residue is divided up into a number of smaller groups (with 1-15 atoms) for various purposes. The electro-neutral groups should be small groups of atoms that contain a few methyl groups, carbonyl groups, N-H groups etc. The electro neutral groups should NOT split up such things as CH3, CH2, CH, OH, NH2, NH, CO groups.
- **Electroneutral group specification** - For each electroneutral group there are two lines specifying the atoms in the group. The first line consists of:
  - An I5 number that is the number of atoms in the group
  - An I5 number that is the (approximate) central atom of the group
  - An F8.3 number that is the radius of the group (should be 3-6Å)
  - The second line consists of 1-15 I5 numbers that are the atoms in the group.

When manually inserting a new library entry into the amino acid library, keep the following points in mind:

- Between each library entry there must be a line of -'s no less than 20 characters long.
- The last four lines of the file must be: (1) a line of -'s, (2) the number 0 somewhere in the first 5 columns,(3) a blank line, and (4) another line of \'s.
- The library entries must conform to the (quite rigorous) format outlined above.

3.2.2 The Force Field Parameter File (parm.lib)

The library file parm.lib controls the molecular dynamics force field that is used to obtain different configurations. If the user introduces a new atom type in amino98.lib, then the corresponding force field parameters have to be added in parm.lib.

Below is an example of a parm.lib file.

```
1 Number of bond types [ 1]
H1 C4 331.000 1.000 Atom codes,force,bond length\{2A5,2F8.3\} [ 2]
1 Number of angle types [ 3]
H1 C3 H1 60.0 120.0 Atom codes,force,angle\{3A5,F8.3,F8.1\} [ 4]
1 Number of torsion types [ 5]
C4 C4 1.500 3.000 120.0 Atom codes, .... \{2A5,2F8.3,F8.1\} [ 6]
1 Number of improper torsion [ 7]
N3 15.000 atom code, force constant\{A5,F8.3\} [ 8]
2 Number of Van der Waals types [ 9]
H1 0900.00 25.000 12.000 Atom type, atom code,\{3x,A2,1X,3F9.3\} [10]
O- 1000.00 25.000 16.000 [11]
'end' End of file indicator [12]
```

The meaning of each line is as follows:

- **Line 1** - This is the number of bond types with the parameters given below. This number has to match the number of lines of bond parameters below. This line is written in [I5] format.
- **Line 2** - The first two columns correspond to the atom types of the bonded atoms. The last two numbers are the force constant and the bond length. This line is written in [2(3X,A2),2F8.3] format.
- **Line 3** - This is the number of angle types with the parameters given below. This number has to match with the number of lines of angle parameters below. This line is written in [I5] format.
- **Line 4** - The first three columns correspond to the atoms types forming the angle. The last two numbers are the force constant and the angle in degree. This line is written in [3(3X,A2),2F8.3] format.
- **Line 5** - This is the number of torsion types with the parameters given below. This number has to match the number of lines of torsion parameters below. This line is written in [I5] format.
- **Line 6** - The first two columns correspond to the atom types of the two atoms, around which the torsion is defined. The last three numbers are the force constant, the periodicity of rotational function and the torsion angle in degree. This line is written in [2(3X,A2),3F8.3] format.

- **Line 7** - This is the number of improper torsion angle types. This number has to match with the number of lines of improper torsion parameters below. This line is written in (I5) format.
- **Line 8** - The first column corresponds to the atom type of the center atom, around which the improper torsion is defined. The second number is the force constant. This line is written in [3X,A2,1X,F8.3] format.
- **Line 9** - This is the number of Van der Waals types. This number should match with the number of lines of parameters below. This line is written in [I5] format.
- **Line 10** - The first column corresponds to the atom type, the second number to the Van der Waals parameter A, the third one to the Van der Waals parameter B and the last one to the mass of the atom. This line is written in [3X,A2,1X,3F8.3] format.
- **Line 11** - This is the number of hydrogen bond vdW parameters. This number has to match with the number of lines of parameters below. This line is written in (I5) format.
- **Line 12** - The first two columns corresponds to the atom types forming the hydrogen bond, the third number to the van der Waals parameter A, the fourth to the van der Waals parameter B. This line is written in [2{3X,A2),1X,2(F9.2,1X)] format.

3.2.3 The EVB library file

This library contains the force field for the EVB atoms. The library is atomic based and adding a new EVB atom means to modify the file in several places. The data in this file is free formatted, user may change or add data in any line, but should not remove any line containing the words 'morse', 'angle', 'torsion' ... etc. For comment line put a '#' at the beginning of the line. The file is read sequentially, so the user can overwrite previous parameters easily without deleting them.

Bonds and angle parameters

For an evb bond A-B: bond energy $E_{\text{bond}} = D\{1 - \exp[-2(r - r_0)]\}^2$, where $D = \sqrt{morsed_A \times morsed_B}$ and bond length $r_0 = radius_A + radius_B$:

```
morse_type      morse_d     radius
'H0'            105.0       0.4
'H+'            105.0       0.4
'H-'            110.0       0.4
'C0'            96.0        0.7
'C+'            96.0        0.7
'C-'            96.0        0.7
'N0'            92.0        0.7
'N+'            92.0        0.7
'N-'            92.0        0.7
'O0'            90.0        0.7
'O+'            90.0        0.7
(...)
```

Similarly, for the bending we have $E_{\text{angle}} = force_A \times (angle - angle0)^2$

```
angle_type     force      angle
'H0'            0.00        0.0
'H+'            0.00        0.0
'H-'            0.00        0.0
'C0'           50.00      109.5
'C+'           50.00      120.0
'C-'           50.00      120.0
'N0'           50.00      109.5
'N+'           50.00      120.0
'N-'           50.00      120.0
'O0'           50.00      109.5
'O+'           50.00      120.0
'O-'           50.00      120.0
'D0'            0.00        0.0
'OD'           50.00      120.0
'Ca'            0.00        0.0
(...)
```

For torsions: $E_{\text{torsion}} = force \times (1 + cos(periodicity * angle - phaseangle))$

```
torsion_type   force      periodicity  phase_angle
'C0'  'C0'     10.0       3.0          0.00
(...)
```

and, finally, for improper torsions "centered" in a given atom, $E_{\text{itorsion}} = force \times (1 + cos(periodicity * angle - phaseangle))$

```
itorsion_type  force      periodicity  phase_angle
'N-'           10.0       2.0          180.0
'C+'           20.0       2.0          180.0
'Oh'            0.0       1.0            0.0
'Ow'            0.0       1.0            0.0
(...)
```

Non-bonded parameters

EVB nonbond interaction for evb atoms A...B, molaris uses exponential and 6-12 vdw formula for nonbond pair atoms pair, which are bonded or angled to each other in one of the resonance states:

$$E_{\text{nb}} = \sqrt{exrepl_A * exrepl_B} * exp - \sqrt{\beta_A * \beta_B} * r + \frac{vdwa_A * vdwa_B}{r^{12}} - \frac{vdwb_A * vdwb_B}{r^6} \tag{3.1}$$

```
exnonbond_type  ex_repl   beta      vdw_a      vdw_b
'H0'            5.00      2.50       7.00       0.00
'H+'           50.00      2.50     100.00       0.00
'H-'            5.00      2.50       7.00       0.00
'C0'           91.00      2.50     632.00      24.00
'C+'           91.00      2.50     632.00      24.00
'C-'           93.00      2.50     632.00      24.00
'N0'           43.00      2.50     774.00      24.00
(...)
```

Nonbond parameters for evb-evb pairs, which are bonded or angled in one of the evb

state. The parameters will override the above ones for the specified pairs:

$$E_{\rm nb} = exrepl * exp{-}\beta * r + \frac{vdwa}{r^{12}} - \frac{vdwb}{r^6} \tag{3.2}$$

Notice the defined parameters are combined ones for the pair types, rather than the above ones based on individual type

```
x_pair          ex_repl    beta      vdw_a     vdw_b
#'C0' 'C+'      88.00     2.50     400000.0     600.0
'L0' 'C0'        00.0     0.00       50000.        0.
'L-' 'C0'        00.0     0.00       50000.        0.
'O-' 'C0'        00.0     0.00        1000.        0.
'O0' 'C0'        00.0     0.00        1000.        0.
'H0' 'C+'       100.0     2.50           0.        0.
'H0' 'O-'       100.0     2.50        3000.        0.
'H+' 'O-'       100.0     2.50           0.        0.
(...)
```

EVB VDW nonbond interaction for evb atoms A...B, which are never bonded or angled to each other in any resonance states. The keyword 'vdw_pair' inside evb level can override the parameters here.

$$E_{\rm nb} = \frac{vdwa_A * vdwa_B}{r^{12}} - \frac{vdwb_A * vdwb_B}{r^6} \tag{3.3}$$

```
vdwevb    vdw_a       vdw_b
'H0'       7.00        0.00
'H+'     100.00        0.00
'H-'       7.00        0.00
'C0'     632.00       24.00
'C+'     632.00       24.00
'C-'     632.00       24.00
'N0'     774.00       24.00
'N+'     774.00       24.00
'N-'     774.00       24.00
'O0'     774.00       24.00
(...)
```

Nonbond parameters for evb-evb pairs, which are never bonded or angled in any of the evb state. The parameters will override the above ones for the specified pairs:

$$E_{\rm nb} = \frac{vdwa}{r^{12}} - \frac{vdwb}{r^6} \tag{3.4}$$

Notice the defined parameters are combined ones for the pair types, rather than the above ones based on individual type

```
v_pair              vdw_a       vdw_b
#'C0' 'C+'         400000.     600.
'L-' 'H0'           50000.       0.
'L0' 'H0'           50000.       0.
'L0' 'L-'          100000.       0.
'L-' 'O-'          100000.       0.
'O-' 'H0'            1000.       0.
(...)
```

VDW nonbond interaction for evb with protein or evb with water,

$$E_{\rm nb} = \frac{vdwa_A * vdwa_B}{r^{12}} - \frac{vdwb_A * vdwb_B}{r^6} \tag{3.5}$$

```
solvdw   vdw_a      vdw_b
'H0'      7.00      0.00
'H+'    100.00      0.00
'H-'      7.00      0.00
'C0'    632.00     24.00
'C+'    632.00     24.00
'C-'    632.00     24.00
'N0'    774.00     24.00
'N+'    774.00     24.00
'N-'    774.00     24.00
'O0'    774.00     24.00
(...)
```

The inductive interaction between evb atoms, by the formula:

$$E_{\rm ind} = -166.0 \frac{\sqrt{alfh(i) \times alfh(j)}}{R_{i,j} * *4} \tag{3.6}$$

is used by default, The keyword 'R4' inside evb level will override the parameters here. If the keyword 'induce_evb 1' inside evb level is used, then the above and the next inductive formulas, and the keyword 'R4' are overridden by the exact evb induce formula:

$$E_{\rm ind} = -166.0 alfh(i) \sum \frac{charge(j) r(i,j)}{R_{i,j} * *3 screen(i,j)} \tag{3.7}$$

where alph(i) is the polarizability for evb atom i and charge(j) is the evb charge for evb atom j in a given evb state, and

$$screen(i,j) = 1.0 - \exp \frac{-R(i,j) * *4}{(screen(i) * screen(j)) * *2} \tag{3.8}$$

```
induct  alph  screen
'H0'    0.0     1.0
'H+'    0.0     1.0
'H-'    0.0     1.0
'C0'    0.0     1.0
'C+'    0.0     1.0
'C-'    0.0     1.0
'N0'    0.0     1.0
'N+'    0.0     1.0
(...)
```

The same as the above for inductive interaction but alfa is based on evb atom pair, by the formula:

$$E_{\rm ind} = -166.0 \frac{alfh(i)}{R_{i,j} * *4} \tag{3.9}$$

which will override the above inductive interaction formula.

```
a_induct   alph
#'C0' 'C+'  4.07
'C0' 'L0'   0.00
'C0' 'O0'   0.00
'H0' 'L0'   0.0
'H0' 'L-'   0.0
(...)
```

The electrostatic (screening) correction term for evb-evb atom nonbond interaction by the formula:

$$E_{\text{scr}} = -332.0Q(i)Q(j)\frac{\exp -[mu_s(i) + mu_s(j)] * R_ij * *2}{R_ij} \tag{3.10}$$

where Q's are charges and R_ij is the distance between atom i and j. The keyword 'el_correc' inside evb level can override the parameters here.

```
elect   mu_s
'H0'    2.00
'H+'    2.00
'H-'    2.00
'C0'    2.00
'C+'    2.00
'C-'    2.00
'N0'    2.00
(...)
```

The same as the above for screening correction term but mu_s is based on evb atom pair, by the formula:

$$E_{\text{scr}} = -332.0Q(i)Q(j)\frac{\exp -[mu_s(j)] * R_ij * *2}{R_ij} \tag{3.11}$$

which will override the above ones

```
a_elect    mu_s
'L-' 'C0'   4.0
'L-' 'H0'   4.0
# h3o h2o
'Oh' 'Ow'   2.5
'Oh' 'Hw'   2.2
'Ow' 'Hh'   3.5
'Hh' 'Hw'   2.5
# oh h2o
'Oo' 'Hw'   1.7
(...)
```

parameters for evbsimp solvation

```
simple_evb rsolv
'H0'       2.20
'H+'       2.20
'H-'       2.20
'C0'       2.65
'C+'       3.00
'C-'       3.25
'N0'       2.65
'N+'       2.65
(...)
```

3.3 LD Calculations Using *ab initio* Charges (the ChemSol Program)

The program ChemSol is designed for the calculations of solvation free energies by using a Langevin Dipoles (LD) model of the solvent. The implementation, parameterization for aqueous solution, and some applications of the model are described in detail in refs. Florián, 1997 and Florián, 1999. To obtain a copy of these manuscripts send an e-mail to Jan Florián (florian@rcf.usc.edu). Also, you can go to our web page `http://laetro.usc.edu` for contact information and for running ChemSol on-line free of charge.

Copies of the program ChemSol can be downloaded free of charge from anonymous ftp at: ftp.usc.edu, directory pub/warshel/cs.

3.4 Demos included in the package

In this section we describe the demos included in this release of the program. The demos are located in the directory $MOLARIS_PATH/demo. You can run them altogether using the script shrunall located in the same directory or you can run a particular demo by typing:

```
$ shrun demoX
```

This short script will change to the corresponding subdirectory and run `MOLARIS` with the adequate input file.

3.4.1 ez_relax: Relaxation of a protein

```
#
#   MOLARIS DEMO 1
#
# authors: CVS and JV
# date: April 2000
# description:
# In this demo we perform a ridiculously short relaxation (this calculation
# is just for teaching purposes, of course) of a protein using
# a simple MD procedure changing some of the default parameters.
# At the end a new PDB file with the relaxed structure is generated
#

../pdb/bpti.pdb keeph
analyze
   allres
   resatom 33
   resatom 57
end
enzymix
   relax
      md_parm        #an example of how to change the default parameters
         nsteps 100
         temperature 200.0
         stepsize 0.0002
         constraint_1 30.
         constraint_2 60.
         langevin_r 16.
         water_r  16.
```

```
         region2a_r 16.
         movie_co 1 to 58
         movie_co wat
         movie_fq 10
      end
   end
end
analyze
   makepdb     #creating a .pdb file with relaxed structure
      residue all
      file_nm $OUT_DIR/bpti_relax.pdb
   end
   viewmovie
      file_nm $OUT_DIR/movie.dat
      view_fq 10
      vwall
      form xmol
   end

end
end
```

3.4.2 pl_pka_pdld: pKa calculation

```
#
#   MOLARIS DEMO 2
#
# authors: CVS and JV
# date: April 2000
# description:
# In this demo we perform a calculation of the pKa of residue 3
# in the given PDB file.  The way the intrinsic pKa is assigned is
# illustrated.
# note the three sections on the MD calculation. The first correspond
# to the initial relaxation, and the other two correspond to water and
# protein runs respectively (for each configuration specified by
# config the twoxtwo -for A and B- MD runs are performed
# for getting new structures)
# Of course, this is for demo purposes only

../pdb/bpti.pdb

#gendebug
analyze                        # get some infor from the PDB file
   allres
   resatom               1
   sphereion 20.0  23.721 24.373 -6.467
   distatom 78 868
   distatom 480 802
end
polaris                        # enter the POLARIS module
   pre_pol
      set_opt                  # set some parameters for PDLD
         rg 18.0
         ndxp 3
         itl 10
      end
   end
   pKa_pdld                    # enter the pKa calculation
      reg1_res 1               # identify region I. Do pKa of residue 1
      config 1 2               # configurations to be done
      md_parm_r                # relaxation dynamics
         nsteps 1000
      end
      md_parm_w                # dynamics in water
         nsteps 500
```

```
      end
      md_parm_p               # dynamics in protein
         nsteps 1000
      end
   end
end

# we are done!
end
```

3.4.3 ez_EVB: EVB calculation

This demo consists of two different runs, one in water and one in protein. The corresponding input files are:

1. in water:

```
# description:
# This is an EVB calculation for the subtilisine reaction mechanism
# in water. The coordinates of the file subwat.pdb are taken from the
# protein PDB file and the atoms are numbered accordingly
#

wat-ser.pdb keephis
# first we relax the system
enzymix
   evb
      evb_state 3 1.0 0.0 0.0 1
        evb_atm    8    0.035   C+    0.164   C+    0.164   C+
       evb_atm    9    0.015   N+   -0.161   N+   -0.161   N+
       evb_atm   10    0.187   H0    0.187   H0    0.187   H0
       evb_atm   11    0.110   C+    0.545   C+    0.545   C+
       evb_atm   12    0.075   H0    0.075   H0    0.075   H0
       evb_atm   13   -0.520   N+   -0.161   N+   -0.161   N+
       evb_atm   14    0.028   C+    0.096   C+    0.096   C+
       evb_atm   15    0.070   H0    0.068   H0    0.068   H0
       evb_atm   25   -0.427   O0   -1.000   O-   -0.200   O0
       evb_atm   26    0.427   H0    0.187   H0    0.187   H0
       evb_atm   48    0.392   C+    0.392   C+    0.200   C0
       evb_atm   49   -0.392   O0   -0.392   O0   -1.000   O-
# add new evb atom
       evb_atm   31    -0.097   C0   -0.097   C0   -0.097   C0
       evb_atm   32     0.097   H0    0.097   H0    0.097   H0
       evb_atm   50    -0.400   N+   -0.400   N+   -0.400   N+
       evb_atm   51     0.400   H0    0.400   H0    0.400   H0
      evb_bnd 0 8 9
      evb_bnd 0 8 14
      evb_bnd 0 9 10
      evb_bnd 0 9 11
      evb_bnd 0 11 12
      evb_bnd 0 11 13
      evb_bnd 0 13 14
      evb_bnd 0 14 15
      evb_bnd 1 25 26
      evb_bnd 2 13 26
      evb_bnd 3 13 26
      evb_bnd 3 25 48
      evb_bnd 0 48 49
#add new bond
      evb_bnd 0 31 32
      evb_bnd 0 31 48
      evb_bnd 0 48 50
      evb_bnd 0 50 51
      gas_dg 1   0.0
      gas_dg 2 115.0
      gas_dg 3  50.0
```

```
      evb_parm
         iflag_r4 0
      end
      rest_out evb12_subwat-rx.res
      md_parm
         temperature 300.0
         nsteps 4000
         ss 0.001
         region2a_r 16
         water_r 16
         langevin_r 18
        constraint_1 0.3
        constraint_post 25 3.0 3.0 3.0 21.4 27.0 20.6
        constraint_pair 13 25 3.0 3.0
        constraint_pair 25 48 3.0 3.0
        log_write_fq 200
      end
   end
end
 # and finally we run the actual EVB calculation
enzymix
   evb
      evb_state 3 1.00 0.00 0.0 1
      map_pf 11 1 2
       evb_atm    8    0.035   C+    0.164   C+    0.164   C+
       evb_atm    9    0.015   N+   -0.161   N+   -0.161   N+
       evb_atm   10    0.187   H0    0.187   H0    0.187   H0
       evb_atm   11    0.110   C+    0.545   C+    0.545   C+
       evb_atm   12    0.075   H0    0.075   H0    0.075   H0
       evb_atm   13   -0.520   N+   -0.161   N+   -0.161   N+
       evb_atm   14    0.028   C+    0.096   C+    0.096   C+
       evb_atm   15    0.070   H0    0.068   H0    0.068   H0
       evb_atm   25   -0.427   O0   -1.000   O-   -0.200   O0
       evb_atm   26    0.427   H0    0.187   H0    0.187   H0
       evb_atm   48    0.392   C+    0.392   C+    0.200   C0
       evb_atm   49   -0.392   O0   -0.392   O0   -1.000   O-
# add new evb atom
       evb_atm   31    -0.097   C0   -0.097   C0   -0.097   C0
       evb_atm   32     0.097   H0    0.097   H0    0.097   H0
       evb_atm   50    -0.400   N+   -0.400   N+   -0.400   N+
       evb_atm   51     0.400   H0    0.400   H0    0.400   H0
      evb_bnd 0 8 9
      evb_bnd 0 8 14
      evb_bnd 0 9 10
      evb_bnd 0 9 11
      evb_bnd 0 11 12
      evb_bnd 0 11 13
      evb_bnd 0 13 14
      evb_bnd 0 14 15
      evb_bnd 1 25 26
      evb_bnd 2 13 26
      evb_bnd 3 13 26
      evb_bnd 3 25 48
      evb_bnd 0 48 49
#add new bond
      evb_bnd 0 31 32
      evb_bnd 0 31 48
      evb_bnd 0 48 50
      evb_bnd 0 50 51
      gas_dg 1   0.0
      gas_dg 2 115.0
      gas_dg 3  50.0
      evb_parm
         iflag_r4 0
      end
      rest_in $OUT_DIR/evb12_subwat-rx.res
      rest_out subwat_12.res
        md_parm
```

```
         temperature 300.0
         nsteps 4000
         ss 0.001
         region2a_r 16
         water_r 16
         langevin_r 18
         constraint_1 0.3
          constraint_post 25 3.0 3.0 3.0 21.4 27.0 20.6
         constraint_pair 13 25 3.0 3.0
         constraint_pair 25 48 3.0 3.0
          log_write_fq 200
       end
    end
 end
 # we are done!!
 end
```

2. in protein:

   The protein calculation uses PDB code 1SBT, which has a quite low resolution. Thus, previous to the protein we minimize the system by 500 steps of steepest descent using the minim_sub.inp file:

```
#
#   MOLARIS DEMO 3
#
# description:
# Minimization with the steepest descent option of the initial protein
# to eliminate errors in the original PDB
#

../pdb/sub.pdb
enzymix
   relax
      md_parm
         steep_mini 1
      end
   end
end
analyze
   makepdb
      file_nm ../pdb/subopt.pdb
      residue all
   end
end

# we are done!!
end
```

   and then we run the regular run in a similar way as we did with the water calculation, only changing the numbering of the atoms (see file evb_sub_12.inp).

The calculation is run for water and protein, both for the proton transfer step (files *12*) and the nucleophilic attack (files *23*). The FEP/US runs are done by running mapping on the provided mapping inputs:

1. water

```
mapping_type evb
fileroot demo_output/evb_subwat_12/map_evb.gap 11
temperature 300.
```

```
tolerance 3
points_throw 10
hij 1 2 45.5  2.5
gas_dg 2 130.
map_pf 11 1 2
end
```

2. protein

```
mapping_type evb
fileroot demo_output/evb_sub_12/map_evb.gap 11
temperature 300.
tolerance 3
points_throw 10
hij 1 2 45.5 2.5
gas_dg 2 130.
map_pf 11 1 2
end
```

3.5 Bug reports

Program MOLARIS is by far not bug free and bug reports are very welcome. Please, report any bug to us, including if possible all the files (input, output and libraries) used in the failed calculation in order to try to solve it. Contact us using the addresses in http://laetro.usc.edu

Chapter 4

Indexes

4.1 Alphabetical index