Table of Contents

Preference

The Program **MolDyn** is designed to perform multiple tasks with protein structure:

1) restoration of missing coordinates of heavy atoms of side chains;
2) restoration of missing coordinates of all hydrogen atoms;
3) optimization of a protein structure via local energy optimization in an implicit/explicit water solvent;
4) optimization of a protein structure via MD simulation in water solvent;
5) optimization and folding of a protein via a user defined simulated annealing protocol coupled with force field variation.
6) optimization of a user defined flexible protein segments with user defined restraints
7) simulation of the molecular dynamical trajectory for molecular atomic coordinates and potential energy for statistical analysis.
8) exhaustive docking of flexible ligand molecule of size up to ~ 100 atoms to protein molecule.

I. Input and Compilation

1. RUN the program

```
 RUN program by the command

   ../$MDYN07HOME/mDynQ07 -i inProtcol  -c inPDB [-mdR mdRestXYZVin] [-mv moveRes]
             [-r1 inRestrainA1 ] [-r2 inRestrainA2] [-rB rigBodyFile]
             [-sa saProtocol] [-mn molName]  [-mdX mdFinalPDB] -o runOutFile [-er
errorFile]

in parenthesis [ ] are uxilarry files. The auxilary files will be used by program
if the main command file
defines respective task.

Command line DESCRIPTION:

-i inProtcol       : file MdynPar.inp  defines protocol for the mDyn particular Run

-c inPDB           : file of the initial molecular structure as molec.pdb file in
the PDB format

-mdR mdRestXYZVin  : XYZ+Velocity file to REstart MD from the last snapshot file
XYZV , see exaple t5
                     1arb.mdXYZVfin0001.pdb it is USED with $mdRestart keyword in
command file
                     inProtcol
                     NOTE!  the initial XYZ will be taken from mdRestXYZVin file !
                           the PDB file inPDB is not USED with the key -mdR

-r1 inRestrainA1   : file defines  of positional restraints for atoms of the
molecule
-r2 inRestrainA2   : file defines atom-atom distance restraints
-rB rigBodyFile    : file defines rigid body segments of the main chain of protein
-mv moveRes        : file defines List of moving Residues
-sa saProtocol     : file defines simulated annealing protocol
-mn molName        : character set defining molecula name.  molName. will be
attached to RESULT files
-o runOutFile      : run output file
-mdX mdFinalPDB    : final PDB file of the Energy/MD optimization

Current status of program run is printed on the standart output device (consol) or
can be redirected to user defined file or can be defined in the argument line:
-er errorFile      : error message file : they are dublicated in the runOutFile
#
if file name definition in the argument line is missing for a file
than the default name is used for this file

NOTE! if the command line does not include a key -X , while the command file
defines task which need data file coupled with -X keyword,  than program try to
find default (standart) name data file in the current directory.
Default names:
#
inProtcol   = ./MdynPar.inp
inPDB       = ./molec.pdb
mdRestXYZVfile = ./mdXYZVin.pdb
moveRes     = ./moveRes.inp
inRestrainA1  = ./restrAt1.inp'
inRestrainA2  = ./restrAt2.inp'
```

```
rigBodyFile = ./rigBody.inp
saProtocol  = ./SAprotocol.inp
molName     = space
runOutFile  = ./mDynSB.out
errorFile   = ./mDynSB.err
mdFinalPDB  = ./molMdFin.pdb
#
```

2. Input file and keyword description

```
inProtcol   = ./MdynPar.inp

The nain command file consist of lines with command keyword.
Keyword start with $ sign in the first position of line
One Keyword in line

#example of MdynPar.inp file and keyword description
# MdynPar.inp
$OUTfull                              ! full extended output of program run

#Initial PDB data quality
$Hread                                ! read INPUT pdb file with Hydrogens
                                      ! by default OUTshort option is ON
# DEfinition of OPtimized segments of protein:
$fullProtMD                           ! full molecule is flexible
#$MovingRes                           ! defines List of opimized segments

#FORCE FIELD MODIFICATIONS:
#
$shake=2                         ! all valence bobds are fixed by shake method

$zeroRot                    ! exclude translation and rotation of the molecule
                                        as rigid body

$hBond128 = 2.0                           ! scaling coeff for H-bonds
                                   ! default=1.0 it is standart force field

$harmAt1PosRst=0.25                       !invoke restraintsA1 type =
                             positional harmonic restraints for atom position
                                          with harmConst (kcal/A^2).
                                 program need a special file -r1 restrA1File
                                 which defines restrained segmants of protein
                                          see additional description

$distRestrA2                     !invoke restraintsA2 type atom-atom distances
                                   for user defined pairs of atoms in the file
                                -r2 restrA2File (see additional description)


$rigBody               !invoke optimization with frozen internal structure of
                     protein main chain for user defined segments of sequence
                  need file -rB rigidBodySegment (see additional description)

$compactForce = 0.5                  ! invoke additional compactization forces
                                          ! to accelerate protein folding
#
$aSoftCore = 0.5                 !invoke SOFTNES for the van der waals atom-atom
potential
                                   ! at the small (contact) atom-atom distances
                          ! Use of the softCore VDW potential helps to optimize
```

```
                  ! BAD molecular structures with many spartial atom-atom clashes
                        ! values range  0 - 1 from very Soft to standart VDW
#SOLVATION MODEL
$SolvMod = GShell
#
#
# OPIMIZATION PROTOCOL:
$engCalc                                ! do energy calculation
$engOptim                               ! do energy optimization by local Optimizer
$nOptStep=1                             !max N optim steps
#
#PROTOCOL for Molecular Dynamics:
$doMDyn                                 ! do MolDynamics
$MDSA                                   !do MolecularDynamis SimAnnealing
                                  needs SAprotocolFile -sa saProtocol File,
                                         see additional description
#
#PROTOCOL of MD equilibration:
#
$initMDTemp=50.00                       !initial Temperature to start MolDyn
$bathMDTemp=50.00           !thermostat temperature of thermostat i.e. target
temperature
$runMDnstep=2000           !number of time-steps for MD simulation
$mdTimeStep=0.002
#
$NTV=1                                 ! MD ensemble definition
#
#
# MD Trajectory writing:
$nwtra=500
$WRpdb                           ! write snarshort structures in the PDB format
                           ! default WRpdbq OPTion is ON : extended PDB format
                                        ! PDB + Qatom
#
END
#
NOTE that parameter file formatted, i.e. $ sign should be  the firs character of
the line
--------------------------------------------------------------------------------
KEYWORD LIST:
        keyw = 'OUTfull'
        keyw = 'WRpdb'
        keyw = 'Hread'
        keyw = 'fullProtMD'
        keyw = 'MovingRes'
        keyw = 'LigRes'
        keyw = 'doLigDock'
        keyw = 'MDSA'
        keyw = 'SolvMod'
        keyw = 'zeroRot'
        keyw = 'hBond128'
        keyw = 'harmAt1PosRst'
        keyw = 'distRestrA2'
        keyw = 'compactForce'
        keyw = 'shake'
        keyw = 'engCalc'
        keyw = 'engOptim'
        keyw = 'nOptStep'
        keyw = 'aSoftCore'
```

```
        keyw = 'initMDTemp'
        keyw = 'bathMDTemp'
        keyw = 'mdTimeStep'
        keyw = 'runMDnstep'
        keyw = 'doMDyn'
        keyw = 'mdRestart'
        keyw = 'NTV'
        keyw = 'nwtra'
--------------------------------------------------------------------------
KEYWORD DESCRIPTION:

#OUTPUT DETAILES:
$OUTfull                                  ! full extended output of program run
                                          ! by default OUTshort option is ON
#
# INPUT PDB FILE DETAILES:
$Hread    ! defines that all Hydrogens will be read from input molecule structure
-c inPDB   file
            otherwise the ALL HYDrogens will be restored by the program, i.e.
            all H atoms will be deleted and added according to molecular topology
for RESidues.
            Using Library in the ./dat/h_add.dat
NOTE! it is recommended start to works with a new protein without option $Hread
even if the PDB
file has all hydrogen atoms, because the hydrogen atom names for protein side
chains
have multiple definition in the PDB data base.
It is better if mDyn  program will add all hydrogens to the heavy atoms.

#DEFINITION OF OPTIMIZED RESIDUES:

$fullProtMD                              !defines FULL (i.e. ALL atoms) of the USER
molecule
                                          will be free to move in energy relaxation
or molDyn

$MovingRes              ! logical keyWord defines that only a defined set of
RESidue are free
                this keyWord is coupled with file -mv moveRes in the argument line
to start
                the program
                default name for moveRes file is ./moveRes.inp

#EXAMPLE of ./moveRes.inp
#1arb
aaaaaaIIIIiiii
#
MOVRES   1  10       !line defines first and last resudue of moving segments
integers devided by space
MOVRES  45  76
MOVRES 115 260
end                 !end or END should be last line if the file
************

#FORCE FIELD DEFINITION:

$hBond128 = 2.0                          ! scaling coeff for H-bonds

$aSoftCore = 0.5                         !invoke van der waals atom-atom potential
```

```
with modified
                                        ! SoftCore at the small (contact) atom-
atom distances
                                        ! SoftCore modification is used for
energyOPtimization
                                          and MD equilibration stages.
                                        ! Use of the softCore VDW potential helps
to optimize
                                        ! BAD structures with many starical atom-
atom clashes
                                        ! values range  0 - 1 from very Soft to
standart VDW

$harmAt1PosRst=0.25  ! digital keyWord define RESidue segments with 1 atom position
harmonic
                     restrants.
                     0.25 = harmonic restrain Constant K
                     restrEnergy = 0.5*K(r - r0)**2,
                     the reference position r0 = initialXYZinput.pdb - positions
from
                     the initial INPut PDB file which defines INItial structure
of molecule

    this keyWord is coupled with file -r1 inRestrainA1 of the argument line to
start
                the program mdyn
                default name for inRestrain file is ./restrAt1.inp

#EXAMPLE  of inRestrainA1 file:
#harmonically restrained RESidue segments
#xxxxxIIIIiiiiaaAAA
#(6x,2i4,a40)
RESTA1   1  63  PBB          ! line starts from keyWord RESTAT numbers=first/last
residue of segment
                             ! PBB (only protein backbone atoms are restrained,
i.e. side chains are free)
RESTA1  78 120  ALL          ! ALL (all atoms are restrained)
                             ! integers and words are devided by space
end
# ------------------------------------------------
$distRestrA2                 ! defines optimization/MD with atom-atom dist
RestrainA2
                             ! needs file  [-r2 inRestrainA2] in command line
-r2 inRestrainA2 : default name : restrAt2.inp
#
EXAMPLE of inRestrainA2 file:
#harmonically restrained Atom-Atom distances
#xxxxxx
#keyword atom1       atom2       distA HarmConst(kcal/mol*A^2)
RESTA2   ND2  ASN 222 : OG1 THR 219 = 7.0   1.5
RESTA2   O  GLY 170 : OG1 THR 219 = 8.0   2.5
RESTA2   OH TYR 109 : OG1 THR 111 = 7.5   3.0
END
#------------------------------------------------
$rigBody                   !defines optimizatiom/MD considering some segments of
the main chain
                           ! as a rigid body.
                           ! The List of rigid  segments of the main chain is user
defined.
```

```
                              ! Each segment will keep rigid internal structure of
the protein main chain,
                              ! has rotatational and translational degrees of
freedom.
                              ! The side chains of the rigid segments are flexible.
#Needs file rigidBody.inp
#EXAMPLE of rigidBody.inp file:
#
RIGB01  11  16       !line defines first and last resudue of moving segments
integers devided by space
RIGB02  47  59
RIGB03  77  99
end                  !end or END should be last line if the file
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$compactForce = 0.25         ! define additional compactization forces for protein
atoms
                              ! Recomended forceParameter = 0.1 - 1.0
# ------------------------------------------------

$shake=2    ! invoke shake subroutine to keep bonds fixed. =1 -bonds with Hydrogen,
=2 all bonds

------------------------------------------------------
#Defining of the SOLVation model:
there are 4 variants  of Implicit models
          1 variant of Explicit model
#:
$SolvMod = GShell            ! implicit  Gaussian Shell solvation model
$SolvMod = GShell + WBrg     ! implicit  Gaussian Shell solvation model +
WaterBridges between polar atoms
                           ! WaterBridges descride solvent mediated interactions
trough stong bound water
                           ! molecules via implicit model of water bridges

$SolvMod = GBorn             ! implicit Generalized Born model + SAS HydroPhobic
solvation
$SolvMod = GBorn + WBrg      ! implicit Generalized Born model + SAS HydroPhobic
solvation + WaterBridges

$Solv = ExWshell 4.5 [A] ! explicit water shell of 4.5 Angst around protein;
                           ! recomended thikness 3.0 - 6.0 A
-------------------------------------------------
$mdRestart    ! restart molDynamics from a snapshot [molName.]mdXYZVfin000N.pdb
                the file [molName.]mdXYZVfin000N.pdb should be copied to the file
mdyn Restart file
                mdXYZVin.pdb

$doMDyn       ! do molecular dynamics
$MDSA         ! do Molecular Dynamical Simulated Annealing
              ! coupled with file -sa SAprotocol which define protocol of the
simulated annealing

#EXAMPLE of Aprotocol.inp  file
#SA protocol
#nSAstep 2
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#      nMDstep    tempTg   SCvdW wfHb128BB wfhB128BS
SAPROT 100000      500.0     0.8    1.0     1.0        !line starts from keyword
SAPROT
```

```
SAPROT 100000      100.0     1.0    1.0     1.0
END
#
   nMDstep - number of md timeStep
   tempTg  - target temperature in K, this temperature will be reach during ntimeMX
steps
   SCvdW   - parameter 0 - 1 to define softness of the van der waals potential.
Soft potential
             modifies Potential Energy Surface and decrease  barriers of
conformational transitions
   wfHb128BB,
   wfhB128BS - (1 - 0) scaling factors for BackBone-BackBone and
               BackBone-SideChain Hydrogen Bond energy
#-------------------------------------------------
#
# OPIMIZATION PROTOCOL:
$engCalc                                ! do energy calculation
$engOptim                               ! do energy optimization by local Optimizer
$nOptStep=1                             !max N optim steps
#
#PROTOCOL for Molecular Dynamics:
$doMDyn                                 ! do MolDynamics
$MDSA                                   !do MolecularDynamis SimAnnealing
                                         needs SAprotocolFile -sa saProtocol File,

#MD EQUILIBRATION:
$initMDTemp=50.00                      !defines initial temperature to start MD
                                       ! recommended low temperature < 50K
                                       ! temperature can be steadelly increased to
the 300K and higher
                                       ! USING $MDSA option
$bathMDTemp=50.00                      ! bath temperature in the MD equilibration run
$runMDnstep=2000                       ! number of MD time steps  in the
equilibration run
$mdTimeStep=0.002                      ! value of the MD time step in ps,
                                       ! recomended 0.001 - 0.002
$NTV=1                                 ! ansemble NTV=0/1
                                       ! =1 md run with constant T
#MD TRAJECTORY WRITING
$nwtra=500                             ! structure XYZ (snapshot) will be written
                                       !as a series of molMdResXXXX.pdb files

$WRpdb                                     ! write snapshort structures in the PDB
format
                                           ! default is WRpdbq OPTion is ON :
extended PDB format
                                           ! PDB + Qatom column
#* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * *
#
-c inPDB   file  - standart pdb file

#EXAMPLE of inPDB   file:
*********************************************************************************
*****
NOTE! it is recommended to start to work with a new protein without option $Hread
even if the PDB
file has all hydrogen atoms, because the hydrogen atom names for protein side
chains
```

```
have multiple definition in the PDB data. It is better if mDyn  program will add
all hydrogens
to the heavy atoms.
********************************************************************************
********
REMARK: PDB:
ATOM      1  N   GLY A   1      11.726 -10.369  10.598
ATOM      2  H1  GLY A   1      11.921 -11.015   9.807
ATOM      3  H2  GLY A   1      12.518 -10.395  11.271
ATOM      4  H3  GLY A   1      10.852 -10.663  11.079
ATOM      5  CA  GLY A   1      11.567  -9.015  10.090
ATOM      6  HA2 GLY A   1      10.772  -8.977   9.420
ATOM      7  HA3 GLY A   1      12.439  -8.710   9.612
ATOM      8  C   GLY A   1      11.280  -8.099  11.303
ATOM      9  O   GLY A   1      11.256  -8.584  12.493
ATOM     10  N   VAL A   2      11.060  -6.876  11.020
ATOM     11  H   VAL A   2      11.066  -6.574  10.025
etc.
TER                ! CHAIN TERmination
ATOM   1302  N   GLY A  94      10.957 -15.678  12.832
ATOM   1303  H   GLY A  94      10.735 -14.663  12.877
ATOM   1303  H   GLY A  94      10.735 -14.663  12.877
ATOM   1304  CA  GLY A  94      10.193 -16.559  11.950
ATOM   1305  HA2 GLY A  94       9.428 -16.004  11.516
ATOM   1306  HA3 GLY A  94       9.784 -17.323  12.525
ATOM   1307  C   GLY A  94      11.016 -17.184  10.843
...
etc.
TER                ! CHAIN TERmination
END                ! file  END
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
#
# PDB mDyn trajectory file description:
#
        Program mDyn generate a series of snapshot files, e.g.,
1arb.molMdRes0nnn.pdb (test/t4)
the molMdResXXXX.pdb file (see example) contains all atomic coordinates and
additional information
in the REMARK: lines
####
REMARK: Md result : MdTime(ps):    2.4940
REMARK: $nstep:      1247
REMARK: $nRecPDB:       5
REMARK: RMSD(x0):     0.43   <- RMSD all atom
REMARK: badBond: n,erAv(A)  :     0  0.000    <- number and error Average for bond
length in Angstrem
REMARK: badAng : n,erAv(grd):     8   9.42    <- number and error Average for bond
angles in grad
# ENERGY TERMS for the given structure
REMARK: $ENERGY:     :Kcal
REMARK: eVbondDef:     100.89315      <-bond deformation energy
REMARK: eVangDef :     441.63705      <-angle deformation energy
REMARK: eImpDef  :      35.68147      <-Improper torsion agle [planarity] energy
REMARK: eTorsDef :     691.25769      <-torsion potentioal energy
REMARK: engVDWR1 :   -1031.16211      <- van der waals energy for cutoff R1=8 A
REMARK: ehBHxY128:    -608.70599      <- H-bondinds energy
REMARK: engCOULR1:    -816.25323      <- COULOMBIC for distances < cutoff R1
REMARK: engCOULR2:      -4.47208      <- COULOMBIC for distances Rij,  R1< rij
```

3. Ligand Docking

To run Ligand docking modules, the main command file MdynPar.inp
have to include the next keywords:

```
# keywords=value
$LigRes= 282 283          !define start/end ligandResidues

in the inPDB file
                              [(i4,1x,i4) format after= ]
                   !the residues numbers are the same as it is in the initial
                   !inPDB file [united pdb file of protein + ligand]

$doLigDock=1       !run docking for USER defined initial position of Ligand
                   ! as it is in the initial inPDB file [united pdb file of protein
+ ligand]
                   ! Docking is done via simulated annealing molDynamics
                   ! with coupled temperature and force field variation.
                   ! Ligand CMass can move in vicinity of initial
                   ! position +/- 4.0 A
                   ! Orientational global optimization are done via
                   ! simulated annealing MD with multiple start
                   ! orientations. Initial orientations are uniformly
                  ! cover all orientational phase space with distance = 90 deg

$doLigDock=2         ! run ab initio docking
                     ! This option will seach all binding sites on the
                  ! protein molecular surface including cavities and crevices.
                  ! 1) search of surface cavities, crevicies and groovs
                  ! 2) calculation and scoring of binding site candidate
    !    positions based on the number of ligand-protein atom-atom contacts.
    ! 3) ligand docking by simulated annealing molecular dynamics for best
    !    candidate binding sites.
```

#REMARKS:

1) -c inPDBfile in command line should include proteinXYZ + ligandXYZ.
it is recomended to make initial Ligand XYZ in the file inPDBfile
in a contact vicinity of Protein.

2) For a new Ligand, the **Ligand molecular topology** SHOULD BE included into the
LIBrary topology file
   bs_one_all94.dat
at the moment the topology LIB includes the next Ligands
1) benzamidine - BNA
2) biotine - BTN

Ligands of peptide nature, i.e. Ile-Val as it is in the test example, etc.
can be run with available LIBrary of molecular residue topology data.

#
RESULTs of docking:
#
1) Binding site candidates coordinates for the Ligand Center Mass
   and contact score are collected in the file:
LigBSiteOnSAS00.pdb

#
2) Final docking results are collected in series of files:

LigDockFin00n.00m.pdb,

where n-bindig site number, m=1,2,3 - three best
results of docking for different starting orientations of ligand.
File in the PDB format contans energy of interaction Ligand-Protein and
Ligand atom coordinates:

#example:
LigDockFin001.003.pdb for biotin docking on streptavidin - 1stp

```
----------------------
$ENELIG:iPos,nOrient:   1    3
eVbondDefLG:       2.88785
eVangDefLG :      18.85826
eImpDefLG  :       0.25771
eTorsDefLG :       6.50881
engVDWR1LG :     -33.97425
hBHxYeng128L     -25.57005
engCOULR1LG:     -13.67623
engCOULR2LG:      -0.06376
restr1EngLG:       0.00000
eRstHW1MLLG:       0.00000
eGeoDefLG  :      28.51263
engCOULLG  :     -13.73999
engSolvLG  :     -12.24549
engPOTENTLG:     -57.01715
$ENDLIG
REMARK: Ligand PDB:
ATOM   1745  O3  BTN A 122      14.369  -0.753  -8.542       -0.59000
ATOM   1746  C3  BTN A 122      13.171  -0.519  -8.745        0.59000
ATOM   1747  N1  BTN A 122      12.173  -0.648  -7.831       -0.53000
...
ATOM   1774  O1  BTN A 122       7.728   4.860 -13.814       -0.75000
ATOM   1775  O2  BTN A 122       8.565   3.236 -15.125       -0.75000
TER
END
```

The **best (native)** docking result file LigDockFin00n.00m.pdb
can be choosen as file with **MINIMAL value of Potential Energy** of
ligand - protein interactions: engPOTENTLG by command

```
#
grep engPOTENTLG LigDockFin* > 1stp_ePot.dat
```

1stp_ePot.dat:

```
LigDockFin000.001.pdb:engPOTENTLG:     -16.64439
LigDockFin000.002.pdb:engPOTENTLG:     -15.96837
LigDockFin000.003.pdb:engPOTENTLG:     -15.60741
LigDockFin001.001.pdb:engPOTENTLG:     -56.45260    !minimal  - nativeBindSite
LigDockFin001.002.pdb:engPOTENTLG:     -55.64628
LigDockFin001.003.pdb:engPOTENTLG:     -54.99958
LigDockFin002.001.pdb:engPOTENTLG:     -21.24794
LigDockFin002.002.pdb:engPOTENTLG:     -20.24604
LigDockFin002.003.pdb:engPOTENTLG:     -18.27375
LigDockFin003.001.pdb:engPOTENTLG:     -19.86566
LigDockFin003.002.pdb:engPOTENTLG:     -16.73701
LigDockFin003.003.pdb:engPOTENTLG:     -16.02125
#
Example of recomended main parameter file:
#
```

```
#MdynPar.inp for ligand Docking
#-----------------------------------------------
# 1stp : biotin - streptavidin complex
#234567890123456789012345678901234567890!comment
$MoveRes
$LigRes= 122  122                          !LigResN start/end [i4,1x,i4]
$doLigDock=2                               !do Lig Docking for Fixed (rigid) Protein
$hBond128=2.0                              !=scalingCoef for LibDatH128
$Hread
$SolvGS
$doMDyn
$MDSA                                      !do SimAnnealing
$engCalc
#$engOptim
$nOptStep=1                                !max N optim steps
$aSoftCore=0.20                            !softCore 0->1 hardCore
$initMDTemp=30.00
$bathMDTemp=50.0
$runMDnstep=1000
$mdTimeStep=0.002
$nwtra=1000
#END
#-----------------------------------------------------------
#
ligDock_SA_protocol.inp
# recomended Simulated annealing protocol file  for docking
# ----------------------------------------------------------
#nSAstep
4
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#234567890x12345678x123456x123456x123456
#ntimeMX    tempTg  SCvdW wfHb128BB wfhB128BS
2000        300.0     0.1   1.00    1.0
2000        600.0     0.3   1.00    1.0
2000        100.0     0.5   1.00    1.0
2000         50.0     0.8   1.00    1.0
END
#-----------------------------------------------------------
#
```

REMARKS:

1) MoveRes.inp file should include Ligand Residues
2) if $doLigDock=1 , then docking of a ligand for User defined initial ligand position
   can be done for flexible part (or ALLprotein). The moving residues list are
   defined by MoveRes.inp file. Note that the MoveRes.inp should include Lig residues and /or
   user defined protein residues.
3) if $doLigDock=2 than MoveRes.inp file should contain only LigResidues, protein is assumed to be fixed.
   Docking with flexible protein can be done as the next refinement step for rigid protein docking results.

#
RESTRICTION:

A maximum size of flexible Ligand can be docked via available method
is restricted by the size of 30-40 atoms, with topology head-tail or

tail-body-tail. For a large ligands a seach of the native docking site
or ligand binding conformation can be errornes.

#
Test examples for docking

1bty - benzamidine + trypsine complex
1dwb - benzamidine + thrombin complex
1stp - biotin + streptavidine complex
3tpi - ILE-VAL peptide + trypsinogen/BPTI complex

4. Performance

CPU time = 9-10 min/1000 MD step [athlon 1400 MHz]

for protein ~ 3000 atoms

II. Program flow and Basic algorithms of the program

1. Main program

Main Program file : MDynSBmain.f

Start from the call of the input parameters

1. call inputMDSApar

reads the main Input file
filenam = './MdynPar.inp' ! in current job_dir

the file has the fixed name and located in the current job directory
the main input file **MdynPar.inp** defines main parameters of the job (see chapter input file description)

2. call initMolecTopSeq01

reads a defined molecular PDB file, which can be defined in the **MdynPar.inp** file or has the standard name ./molec.pdb and located in the current job directory ./ ;
defines residue sequence

3. call initMolecTopSeq02
calculates 12neighbour list (covalent bonds connecting atoms) using a predefined topology
information about resdues stored in the $MDSBHOME/dat

the pair12 list array: pair12List(*) is the basic molecular topology information. Based on the pair12List(*) the all other lists are calculated, namely
Bonded triplets and quartets to form list of covalent angles, torsion angles, improper torsion angles.
The list of triplets and quartets are calculated via tree algorithm

```
      Call       vbondListPDB2(atomXYZ,
     &           natom,atomNumb,atomName,resName,chName,resNumb,
     &           nres,resNameRes,chNameRes,
     &           atomNameEx,startAtInRes,
     &           nmoveatom,moveAtomList,
```

```
     &           pair12List,startPairL12,nPairL12,np12MAX,
     &           pair13List,startPairL13,nPairL13,np13MAX,
     &           pair14List,startPairL14,nPairL14,np14MAX,
     &           bond12List,nbond12,
     &           trip123List,nTrip123,np123MAX,
     &           quar1234List,nQuar1234,np1234MAX,
     &           quarImp1234L,nImp1234,nImp1234MAX)
```

the call of the subroutine initMolecTopPDB results in the complete definition of the molecular topology
from the input molec.pdb 3D structure.

4. call initFFieldParam

Initialization of the force field parameters for the bond, angle, torsion angle, improper angle deformations,
 van der waals non bond interactions and atomic point charges for the electrostatic interactions.
 For bond, angle, torsion and improper angles a respective list of parameters are generated and stored in the arrays.

A list All force field parameters are based on the amber94 force field parameter set [Cornell et.al 1995].
 Molecular mechanical energy is based on the standard equations for the force field of second generation
 amber94 [Cornell et.al 1995].
Decoding of the atom names (residue names) to the forceField atom name is based on the look up table
ffAtomTypeFile = $MDSBHOME/dat/**atmAAmberff.dat**

5. Extraction of the data from Library file

All search of the proper names in the look up table of the MDynSB program are based on
the **hashing** of a records in the look up table, i.e. conversion of the table in numerically
sequential order. If several records of the look up table have the same hash number (degenerated case),
they are placed in a linkedLis for this hash number.

Force field parameters are taken from the file:
ffParFile = $MDSBHOME/dat/**bsparBATV.dat**
code fragment to initialize force field parameters

```
c get ff-atom code from atomNames
         call defFFatomName (ffAtomTypeFile,
     &              natom,atomNameEx,ResName,chName,
     &              ffAtomName,atomQ)
c
c define bondDef parameters for pair12List()
c
        call getBondDefPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              bond12List,nbond12,bond12ParL)
c c define valence angles def parameters
        call getVangDefPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              trip123List,nTrip123,ang123ParL)
```

```
c define Improper angle def parameters
        call getImpDefPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              quarImp1234L,nImp1234,impAng1234ParL)



c define torsion parameters
        call getTorsPar(ffParFile,
     &        natom,atomNameEx,ResName,chName,ffAtomName,
     &        quar1234List,nQuar1234,quar1234ParL,quar1234nPar)
c
c assign atomMass and vdwParameters
        call  getVDWatMass(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              nVDWtype,atomVDWtype,atomVDW12ab,atomMass)
c
c all FField Parameters are defined
```

6. call initSolvatGSmod

Defines atomic parameters of the current structure for the Gaussian Shell implicit solvation model [Lazaridis, 1999].
A parameters of the GS model are stored in the files:
solvGSPar_aa_amb.dat
solvGSPar.dat

7. call initMDStart(tempT0)

Initialize MD calculation:

Calculate the Initial nonBondPair lists
c generate three nonbonded atom pair Lists: van der Waals, Coulombic and solvation model.

```
c
        makeVdW = 1
        makeCL = 1
        makeSL = 1
c
        call initNonBondList(atomXYZ,makeVdW,makeCL,makeSL)
c
```

Calculates the forces on atoms for initial atomic coordinates
initial forces on atoms

```
c
        fcall = 0
        call initAllForce(fcall,atomXYZ,makeVdW,makeCL,makeSL,
     &              eVbondDef,vbdefForce,
     &              eVangDef,vAngdefForce,
     &              eImpDef,impDefForce,
     &              eTorsDef,torsAngForce,
     &              engVDWR1,vdwForceR1,
     &              engCOULR1,coulForceR1,
     &              engCOULR2,coulForceR2,
     &              restr1Eng,restr1AtForce,
```

```
     &              molSolEn, atomSolEn,atomSolFr)
c
```

Calculates initial atomic velocities, which are distributed according to Maxwell law

$$\text{probability}(v_i) = (\ ) \exp(-m_i v_i^2/kT)$$

```
c
        call initVelocity(temp,natom,
     &       nmoveatom,moveAtomList,atomMass,atomVel0)
c
```

8. Run MD

The subroutine mdRun perform MD run for a given number of time steps ntimeMX

```
c
        call mdRun(ntimeMX,ntime0,ntime,ntimeR1,ntimeR2,
     &             ntimeF1,ntimeF2,ntimeF3,deltat,
     &             tempTg,tauTRF,atype,optra,wtra,nwtra,cltra)
c
```

9. Simulated Annealing optimization

```
c
        call simAnnealing(nSAstep,SAProtcol)
c
```

with user defined SAProtocol(nstep,T) consisted of nSAstep.

Each step of the SA is MD run of nstep with particular temperature T.

III. Details of the atomic force calculation

All atoms of the molecular system consists of two sets of **fixed** and **moving** atoms.

The force are calculated only for the moving atom set.

1. Covalent bond deformation

For covalent bond deformation we use the GROMOS functional form

$$V^{bond}(\mathbf{r}_1,\ldots,\mathbf{r}_N) = \sum_{n=1}^{N_b} \frac{1}{4} K_{bn}[b_n^2 - b_{0n}^2]^2 = \sum_{n=1}^{N_b} V_n^{bond} \quad (1)$$

where

rij = ri - rj

bn = rij .

This functional form is equivalent to the usual harmonic function for a small deformations but a computationally is more effective.

Force on atom i due to bond **n**

$$\mathbf{f}_{in} = -\frac{\partial V_n^{bond}}{\partial b_n^2}\frac{\partial b_n^2}{\partial \mathbf{r}_i} = -K_{bn}[b_n^2 - b_{0n}^2]\mathbf{r}_{ij} \qquad (2)$$

$$\mathbf{f}_{jn} = -\mathbf{f}_{in}$$

Total bond deformation force on atom i is the sum over all bonds **n** involving the atom i.

The calculation of the force $\mathbf{f}_{in}$ is doing by

```
subroutine vbonddefenf(xyz1,xyz2,bondPar,edef,f1,f2) (see file vdefenforce.f)
```

2. Covalent angle deformation

The covalent angle deformation energy function has the form

$$V^{angle}(r_1,\ldots,r_N) = \sum_{n=1}^{N_{angle}} V_n^{angle}(\theta_n, K_{\theta_n}, \theta_{n_0}) \qquad (3)$$

$$V_n^{angle}(\theta_n, K_{\theta_n}, \theta_{n_0}) = \frac{1}{2}K_{\theta_n}[\cos\theta_n - \cos\theta_{n_0}]^2$$

This functional form is equivalent to the usual harmonic function for the angles for a small angle deformation but a computationally is more effective. The angle 2n (at the j) is between atoms i-j-k . The cosine of the angle 2n

$$\cos\theta_n = \frac{\mathbf{r}_{ij} \bullet \mathbf{r}_{kj}}{|\mathbf{r}_{ij}||\mathbf{r}_{kj}|} \qquad (4)$$

The forces on atoms i,j,k due to the deformation of the angle 2n

$$\mathbf{f}_i = -\frac{\partial V_n^{angl}}{\partial \cos\theta_n}\frac{\partial \cos\theta_n}{\partial \mathbf{r}_i}$$

$$= -K_{\theta_n}[\cos\theta_n - \cos\theta_{0n}][\frac{\mathbf{r}_{kj}}{r_{kj}} - \frac{\mathbf{r}_{ij}}{r_{ij}}\cos\theta_n]\frac{1}{r_{ij}} \qquad (5)$$

respectively force on atom k

$$\mathbf{f}_k = -\frac{\partial V_n^{angl}}{\partial \cos\theta_n}\frac{\partial \cos\theta_n}{\partial \mathbf{r}_k}$$

$$= -K_{\theta_n}[\cos\theta_n - \cos\theta_{0n}][\frac{\mathbf{r}_{ij}}{r_{ij}} - \frac{\mathbf{r}_{kj}}{r_{kj}}\cos\theta_n]\frac{1}{r_{kj}} \qquad (6)$$

force on atom j is given from the conservation of the total force acting on three atoms

$$\mathbf{f}_j = -\mathbf{f}_i - \mathbf{f}_k \qquad (7)$$

The covalent angle deformation energy and force are calculated in subroutine

```
subroutine vangldefenf(xyz1,xyz2,xyz3,angPar,
     &                         edef,f1,f2,f3)
```

(see file vdefenforce.f)

3. Torsion angle energy and force

The total torsion energy is a sum over a set of torsion angles for the four atoms i-j-k-l with a rotation around bond j-k ,

$$V^{tors}(\mathbf{r}_1,\ldots,\mathbf{r}_N) = \sum_{n=1}^{N_t} V_n^{tors}(\varphi_n; torsPar)$$

$$V_n^{tors}(\varphi_n; torPar) = \sum_{\alpha=1}^{n_\alpha} K_{n\alpha}[1 + \delta_\alpha \cos(m_\alpha \varphi_n)] \qquad (8)$$

where torsion energy for bond j-k can have several torsion barriers with different multiplicity.

Torsion angle N is defined as

$$\phi = sign(-\mathbf{r}_{jk} \cdot (\mathbf{r}_{ij} \times \mathbf{r}_{kl})) \cdot \arccos(\frac{\mathbf{r}_{im} \cdot \mathbf{r}_{ln}}{r_{im} r_{ln}}) \qquad (9)$$

$$\cos\phi = \frac{\mathbf{r}_{im} \cdot \mathbf{r}_{ln}}{r_{im} r_{ln}}$$

where

$$\mathbf{r}_{im} = \mathbf{r}_{ij} - \frac{(\mathbf{r}_{ij} \bullet \mathbf{r}_{kj})}{r_{kj}^{2}} \mathbf{r}_{kj} \qquad (10)$$

$$\mathbf{r}_{ln} = -\mathbf{r}_{kl} + \frac{(\mathbf{r}_{kl} \bullet \mathbf{r}_{kj})}{r_{kj}^{2}} \mathbf{r}_{kj} \qquad (11)$$

The forces on atoms i,j,k,l due to the single term of eq.(8b) are

$$\mathbf{f}_i = -\frac{\partial V_{n\alpha}^{tors}}{\partial \mathbf{r}_i} = -\frac{\partial V_{n\alpha}^{tors}}{\partial \cos(m_\alpha \varphi_n)} \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} \frac{\partial \cos(\varphi_n)}{\partial \mathbf{r}_i}$$
$$= -K_{n\alpha}\delta_\alpha \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} [\frac{\mathbf{r}_{ln}}{r_{ln}} - \frac{\mathbf{r}_{im}}{r_{im}} \cos\varphi_n] \frac{1}{r_{im}} \quad (12)$$

$$\mathbf{f}_l = -\frac{\partial V_{n\alpha}^{tors}}{\partial \mathbf{r}_l} = -\frac{\partial V_{n\alpha}^{tors}}{\partial \cos(m_\alpha \varphi_n)} \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} \frac{\partial \cos(\varphi_n)}{\partial \mathbf{r}_l}$$
$$= -K_{n\alpha}\delta_\alpha \frac{\partial \cos(m_\alpha \varphi_n)}{\partial \cos(\varphi_n)} [\frac{\mathbf{r}_{im}}{r_{im}} - \frac{\mathbf{r}_{ln}}{r_{ln}} \cos\varphi_n] \frac{1}{r_{ln}} \quad (13)$$

$$\mathbf{f}_j = [\frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{kj}}{r_{kj}^2} - 1]\mathbf{f}_i - \frac{\mathbf{r}_{kl} \cdot \mathbf{r}_{kj}}{r_{kj}^2} \mathbf{f}_l \quad (14)$$

and finally

$$\mathbf{f}_k = -(\mathbf{f}_i + \mathbf{f}_j + \mathbf{f}_l) \quad (15)$$

The torsion energy and force are calculated via

```
      subroutine torsanglenf(xyz1,xyz2,xyz3,xyz4,nTorsH,
     &                       torsPar,eTors,f1,f2,f3,f4)
```

```
c torsPar(4*nTorsH) = {pass,Vt/2/pass,cos(delta),nFi },…
c  eTors = sum{ Ki*[1+cos(delti)cos(i*Ftors)] }; i=1,..,nTorsH
c
```

Torsion parameters are taken from the LibData = bsparBATV.dat

The extraction of the torsion parameters from LibData = bsparBATV.dat for all quartets is done by

```
      subroutine getTorsPar(ffParFile,
     &              natom,atomNameEx,ResName,chName,ffAtomName,
     &              quar1234L,nQuar1234,quar1234Par,quar1234nPar)
c
c InPut:
c       ffParFile - ffParameters file
c       natom,atomNameEx,ResName,chName : PDB info
c       ffAtomName(ia) - FFatomName to search table
c       the quar1234L(i),i=1,..,nQuar1234 : the QuartetList
c RESULT: quar1234Par(16*nQuar1234) - torsionFF parameters for list
```

```
c          of  quartets
c          pass,Vt/2,delta,nFi - (printed) for each torsHarmonics,
c          pass,Vt/2/pass,cos(delta),nFi - finally in array
c         4- torsionHarmanics is possible.
c         quar1234nPar(iQuart) - number of torsHarmonics for the torsAngl
c
```

4. Improper Torsion Angle (out of plane) deformation

The improper torsion angle deformation keeps the four atoms 1-2-3-4 (i-j-k-l) in specified geometry. The first atom in the improper quartet is a planar or (tetrahedral) atom. For example atoms Ci-CAi-N(i+1)-Oi are kept planar. The out of plane potential

$$V^{imp}(\mathbf{r}_{1,\ldots,}\mathbf{r}_n) = \sum_{n=1}^{N_{imp}} V_n^{imp}(\xi_n;\xi_0, K_{\xi 0})$$
$$V_n^{imp}(\xi_n;\xi_0, K_{\xi 0}) = \frac{1}{2} K_{\xi 0}(\xi_n - \xi_0)^2 \qquad (16)$$

CA-N-C-CB are kept in the tetrahedral configuration (L-amino acid) or CA-C-N-CB (D-amino acid) if CA in the united atom (CH) presentation.

The out of plane angle is defined for j-i-k four atoms with i is the planar (tetrahedral)

L

angle between to planes (i-j-k) and (j-k-l) with rotation angle around j-k, other words the

torsion angle in the sequence i-j-k-l

$$\xi_n = sign(\mathbf{r}_{ij} \cdot \mathbf{r}_{nk}) \arccos\left(\frac{\mathbf{r}_{mj} \cdot \mathbf{r}_{nk}}{r_{mj} r_{nk}}\right) \qquad (17)$$

where

$$\mathbf{r}_{mj} = \mathbf{r}_{ij} \times \mathbf{r}_{kj} \qquad (18)$$
$$\mathbf{r}_{nk} = \mathbf{r}_{kj} \times \mathbf{r}_{kl} \qquad (19)$$

The forces on atoms i,j,kl due to a single term Vn

$$\mathbf{f}_i = -\frac{\partial V_n^{imp}}{\partial \xi_n}\frac{\partial \xi_n}{\partial \mathbf{r}_i} = -K_{\xi n}[\xi_n - \xi_0]\frac{r_{kj}}{r_{mj}^2}\mathbf{r}_{mj} \tag{20}$$

$$\mathbf{f}_l = -\frac{\partial V_n^{imp}}{\partial \xi_n}\frac{\partial \xi_n}{\partial \mathbf{r}_l} = K_{\xi n}[\xi_n - \xi_0]\frac{r_{kj}}{r_{nk}^2}\mathbf{r}_{nk} \tag{21}$$

$$\mathbf{f}_j = -\frac{\partial V_n^{imp}}{\partial \xi_n}\frac{\partial \xi_n}{\partial \mathbf{r}_j} = [\frac{\mathbf{r}_{ij}\cdot\mathbf{r}_{kj}}{r_{kj}^2} - 1]\mathbf{f}_i - \frac{\mathbf{r}_{kl}\cdot\mathbf{r}_{kj}}{r_{kj}^2}\mathbf{f}_l \tag{22}$$

finally from the third Newton law

$$\mathbf{f}_k = -(\mathbf{f}_i + \mathbf{f}_j + \mathbf{f}_l) \tag{23}$$

The improper energy and forces for a given improper quartet of atoms are calculated by the subroutine

```
c improper torsion energy force
c
        subroutine imprtorsanglenf(xyz1,xyz2,xyz3,xyz4,impPar,
     &                           eImpt,f1,f2,f3,f4)

c
c ImptPar(2) = K1, ksi0
```

5. Covalent back-bond deformation calculation

All valence back-bond deformation are calculated in the file initAllForce.f

```
subroutine initAllForce(fcall,atomXYZ,
     &              makeVdWs,makeCLs,makeSLs,
     &              eVbondDef,vbdefForce,
     &              eVangDef,vAngdefForce,
```

```
     &              eImpDef,impDefForce,
     &              eTorsDef,torsAngForce,
     &              engVDWR1,vdwForceR1,
     &              engCOULR1,coulForceR1,
     &              engCOULR2,coulForceR2,
     &              restr1Eng,restr1AtForce,
     &              molSolEn, atomSolEn, atomSolFr)
c
        include 'xyzPDBsize.h'
        include 'xyzPDBinfo.h'
        include 'pair1234array.h'
        include 'nbondPairVCS.h'
        include 'vdw12Par.h'
        include 'restrainInfo.h'
        include 'loopInfo.h'
        include 'movingAtom.h'
        include 'solvGSarray.h'
        include 'optionPar.h'
c
 . . . . . .

c
c all GeoDef forces are calculated at each step

        call allAtVBondEForce(atomXYZ,
     &           natom,bond12List,nbond12,bond12ParL,
     &           eVbondDef,vbdefForce )
c
c
        call allAtVangEForce(atomXYZ,
     &           natom,trip123List,nTrip123,ang123ParL,
     &           eVangDef,vAngdefForce )
c
c
        call allAtImpTEForce(atomXYZ,
     &           natom,quarImp1234L,nImp1234,impAng1234ParL,
     &           eImpDef,impDefForce )
c
c torsionEnForces
c
        call allAtTorsEForce(atomXYZ,
     &           natom,quar1234List,nQuar1234,
     &           quar1234ParL,quar1234nPar,
     &           eTorsDef,torsAngForce )
c

 . . . . . .
. . . . . .
```

The deformation forces are calculated at each time step in the MD run.

6. Non bonded pair list calculation

The non bonded pair interactions are calculated for the pair list. Pair list for the central atom i is a sequence of atom numbers for atom within the radius R from the central atom. Three separate pair lists are calculated. The Van der Waals pair list(i) includes atom j if

$r_{ij} \le R1+)R$ (24)

where)R is the buffer size. The buffer size defines the rate of pair list updating frequency

$N_{UPDATE} =)R/[)tVmax]$ (25)

where Vmax is the maximal velocity of an atoms and)t is the time step. The optimal (over CPU time) value of the buffer size can be found. A default value is)R=1 Å.

The pair list calculated with via the lattice algorithm:

1. a) the atomic coordinates $\mathbf{r}_1,\ldots,\mathbf{r}_N$ are projected on the cubic lattice, the integer coordinates of the atoms $\mathbf{h}_1,\ldots,\mathbf{h}_N$ are obtained. The lattice size is quite small ~ 2 A, to include just one atom.
2.

The linked list and all pairList (nnbPairLV, nnbPairLC, nnbPairLS) are calculated in the subroutine

```
c
        subroutine nonbondListVCS(rcutV,rcutC,rcutS,atomXYZ,atomQ,
     &           rbuffV,rbuffC,rbuffS,
     &           makeVdW,makeCL,makeS,
     &           natom,atomNumb,atomName,resName,chName,resNumb,
     &           nres,resNameRes,chNameRes,
     &           atomNameEx,startAtInRes,
     &           nmoveatom,moveAtomList,moveFlag,
     &           pair12List,startPairL12,nPairL12,
     &           pair13List,startPairL13,nPairL13,
     &           pair14List,startPairL14,nPairL14,
     &           nbpairListV,startnbPairLV,nnbPairLV,nnbpLVMAX,
     &           nbpairListC,startnbPairLC,nnbPairLC,nnbpLCMAX,
     &           nbpairListS,startnbPairLS,nnbPairLS,nnbpLSMAX)
c
```

fragment of code for the linked list calculation:

```
c distribute atoms over cells
c make linked list of atoms in cells
c headat(n) - head(incellN)
c linkList(ia) - linkedList
        ixm=1
        iym=1
        izm=1
        do ia = 1,natom
c calculate cell numb
        i3=3*ia-3
        xyzi(1)=atomXYZ(i3+1)-xMIN(1)
        xyzi(2)=atomXYZ(i3+2)-xMIN(2)
        xyzi(3)=atomXYZ(i3+3)-xMIN(3)
        ix = xyzi(1)/cellh+1
        iy = xyzi(2)/cellh+1
        iz = xyzi(3)/cellh+1
        if(ixm .lt. ix)ixm = ix
        if(iym .lt. iy)iym = iy
        if(izm .lt. iz)izm = iz
c cell number
```

```
        ncell = ix + (iy-1)*nsiz(1) + (iz-1)*nsiz(1)*nsiz(2)
        if(ncell .gt. ncell3MAX)then
        write(kanalp,*)'ERROR!:nonbondList: ncell3MAX is low !!'
        stop
        end if!

c make linked list
        linkList(ia) =  headat(ncell)
        headat(ncell) = ia
        end do !ia
c end of linked list calculation

The pair lists VDW and COULOMbic energy exclude 12, 13, 14 covalent bonded pairs.
The Solvent model pairList
include all 12,13, 14 pairs.
The pair list are calculated for the range respectively:
c
        rcutV2 = (rcutV + rbuffV)**2     ! range for List1 -
                                                    VDWaals -  nbPairListV
        rcutV2m = (rcutV - rbuffC)**2    ! range for List2 - Coulombic twin
                                                       range - nbPairListC

        rcutC2p = (rcutC + rbuffC)**2    ! range for List2
        rcutS2 = (rcutS + rbuffS)**2     ! range for SolvationGSList -
                                                            nbPairListS
c

see file nonbobdListVCS.f
```

7. Non bonded force calculation

Van der waals forces are calculated for the non-bonded pair list nbpairListV()for atoms j within rij < RCUTV the cutoff radius for van der waals interactions. The modified potential 6-12 are used

$$U_{vdw} = \sum_{j=1}^{Nj} V_{6-12}^{s}(r_{ij}) \tag{26}$$

where the modified potential is a smoothed 6-12 for a small distances r

$$V_{6-12}^{s}(r) = \frac{Al2}{r^{12}} - \frac{B6}{r^{6}} \quad \text{if } r_{ij} > r_s$$

$$= \frac{\partial V_{6\text{-}12}(r_s)}{\partial r}[r_{ij} - r_s] + V_{6\text{-}12}(r_s) \text{ if } r_{ij} < r_s \tag{27}$$

the pair list for atom i includes atoms j > i, to count each pair interaction once. The force $\mathbf{F}^{vdwi}$ on atom i due to interaction with atoms in the pair list

$$\mathbf{F}_i^{vdw} = \sum_{j=1}^{Nj} \mathbf{f}_{ij} = \sum_{j=1}^{Nj} \frac{\partial V_{6-12}^{s}(r_{ij})}{\partial r_{ij}} \qquad (28)$$

The modified (smoothed) 6-12 potential prevents over-flow when atoms are too close and generates smooth driving forces to resolve clash problems between atoms in molecular dynamics simulations, see

```
c
        subroutine vdwenforceij(dij2,dij1,rij,A12,B12,evdw,fi)
c
```

The coulombic energy and forces for atom i are calculated for all pairs within the radius RCUTC.

The coulombic energy/forces for a central atom i are calculated for the classical coulombic law or as a coulombic interaction between two charges on the compensating background charge uniformly distributed within the sphere of radius RCUTC

$$v_{cl}(r_{ij}) = \frac{q_i q_j}{r_{ij}} \qquad (29)$$

The modified electrostatic potential on the compensating background charge

$$v_{ucl}(r_{ij}) = \frac{q_i q_j}{r_{ij}}(1 + \frac{r_{ij}^3}{2R_c^3} - \frac{3r_{ij}}{2R_c})\Theta(R_c - r_{ij}) \qquad (30)$$

has zero interaction energy and forces for the rij > RCUTC. This form of electrostatic interactions is better suitable to prevent energy conservation in the molecular dynamic calculation, see

```
c
        subroutine coulenforceij(var,rcutC,dij2,dij1,rij,qi,qj,ecoul,fi)
c
```

The nonbonded energy and force within short range RCUTV=R1 are calculated in the subroutine

```
c allAtNonBondEForce : VDW and COULOMBIC
c
        subroutine allAtVDWEForceR1(atomXYZ,atomQ,
     &           natom,nmoveatom,moveAtomList,
     &           nbpairListV,startnbPairLV,nnbPairLV,
     &           pair14List,startPairL14,nPairL14,
     &           nVDWtype,atomVDWtype,atomVDW12ab,
     &           rcutV,rcutC,engVDW,vdwForce,engCOULR1,coulForceR1)
c
```

for the pair list nbpairListV() and pair14List(). The last one includes all 1-4 neihgbours for which the **amber** force field uses the scaling factors for van der waals and coulombic interactions.

To increase performance of the van der waals energy/force calculations the table of coefficient A12, B12 for all atom types are precalculated and then right values A12/B12 for a given atom types in the pair ij are extracted from the vdw AB-parameter table

```
c get pointer to the AB table
        call vdw12TablePos(nVDWtype,t1,t2,t12)
        p4 = 4*t12
        A12 = atomVDW12ab(p4-3)
        B12 = atomVDW12ab(p4-2)
c
```

The long-range electrostatic forces within RCUTV < rij < RCUTC are calculated via the subroutine

```
c
        subroutine allAtVDWEForceR2(atomXYZ,atomQ,
     &           natom,nmoveatom,moveAtomList,
     &           nbpairListC,startnbPairLC,nnbPairLC,
     &           rcutR1,rcutR2,engCOULR2,coulForceR2)
c
c LongRamge -   RCUT1 < rij < RCUT2
```

The program keep separately the short-range and the long-range electrostatic energy and force.

8. Solvation energy/force calculation

The implicit solvation model - the Gaussian Shell model of Lazaridis & Karplus is used to calculate the solvation energy [POTEINS 35: 133-152, 1999]. The solvation free energy of the atom i

$$\Delta G_i^{sl} = \Delta G_i^{ref} - \sum_{j \neq i} g_i(r_{ij}) V_j \qquad (31)$$

where sum is going over all neighbors of atom i which exclude volume Vj from the solvation volume around of the atom i. The function gi(r) describe the solvation energy density in the volume around the atom i and is approximated by the Gaussian function

$$g_i(r) = \frac{\Delta G_i^{free}}{2\pi r^2 \sqrt{\pi}\, \lambda_i} \exp(-[\frac{r - R_i}{\lambda_i}]^2) \qquad (32)$$

where the solvation model parameters)Gref i ,)Gfree i , Vi , 8i , Ri are defined empirically and stored in /data/ directory file **solvGSpar.dat**.

The solvation force on atom i

$$\mathbf{f}_i = -\frac{\partial G^{sl}}{\partial \mathbf{r}_i} = -\sum_{j \neq i} g_i(r_{ij})[\frac{r_{ij} - R_i}{\lambda_i^2} + \frac{1}{r_{ij}}]\frac{V_j}{r_{ij}}(\mathbf{r}_i - \mathbf{r}_j) - \sum_{j \neq i} g_j(r_{ij})[\frac{r_{ij} - R_j}{\lambda_j^2} + \frac{1}{r_{ij}}]\frac{V_i}{r_{ij}}(\mathbf{r}_i - \mathbf{r}_j) \qquad (33)$$

The sum over all solvation forces **fi** is zero.

The solvation forces are calculated by subroutine

```
c
                 call SolventEnForces(natom, atomXYZ,
     &          atomName,startPairL12,nPairL12,pair12List,
     &          nbpairListS,startnbPairLS,nnbPairLS,
     &          atomSolPar, molSolEn, atomSolEn, atomSolFr)
c
```

IV. Details of MD run

An MD run is performed by subroutine

```
c
        subroutine mdRun(ntimeMX,ntime0,ntime,ntimeR1,ntimeR2,
     &                ntimeF1,ntimeF2,ntimeF3,deltat,
     &                tempTg,tauTRF,atype,optra,wtra,nwtra,cltra)
c
c MD RUN propagates MDtraj from files in mdAtomXYZvel.h
c                                     [ atomXYZ0(*),atomVel0(*) ]
c    call initMDStart(T) inits the MD start
c                          from the INput atomXYZ(*)-->atom0XYZ(*)
c
c ntimeMX max number of time steps
c ntime0 - executed number of timesteps in the previous call
c ntime   executed number of timesteps in this call
c ntimeR1, ntimeR2 - update frequency  for R1, R2 pairLists
c ntimeF1,ntimeF2 - update freq for R1=(vdw+coulR1), R2-coulR2 en/forces
c ntimeF3 -  SOLVation forces
c GeoEn/force ntimeFg=1 - standart
c deltat- timestep, temp - initial(temp) of MD run
c tempTg - target T for NTV ansemble[K]
c tauTRF - tau Relaxation Factor [ps]
c atype - ansamble type = 0/1 - NEV, NTV
```

The MD algorithm consist of a long loop over the time steps.

For each time step MD trajectory is propagated for the)t = 1-2 femto sec, as defined by user.

1. Pair lists

The pair lists are updated for each n-th timestep equal to ntimeR1, ntimeR2 for the short-range and for the twin-range long-range electrostatic energy calculations.

```
c
        call initNonBondList(atomXYZ0,makeVdW,makeCL,makeSL)
c
```

2. The atomic forces

The atomic forces due to deformation of covalent structure and short-range non-bonded calculation are updated for the each ntimeF1-th time step, the long-range electrostatic are updated for the each ntimeF2-th step and solvation forces are updated for each ntimeF3-th time step.

{Note! In the current version the multiple time step for pair list update and md equation integration are equal. The general case is not tested !}

```
c update forces/energy
        call initAllForce(fcall,atomXYZ0,doVdWef,doCLef,doSLef,
     &              eVbondDef,vbdefForce,
     &              eVangDef,vAngdefForce,
     &              eImpDef,impDefForce,
     &              eTorsDef,torsAngForce,
     &              engVDWR1,vdwForceR1,
     &              engCOULR1,coulForceR2,
     &              engCOULR2,coulForceR2,
     &              restr1Eng,restr1AtForce,
     &              molSolEn, atomSolEn, atomSolFr)
```

MD simulation can be done with a specified set of forces. The set of forces can be specified by the array fEngWF(*)

```
c
          eGeoDef   = fEngWF(1)*eVbondDef + fEngWF(2)*eVangDef
     &                + fEngWF(3)*eImpDef + fEngWF(4)*eTorsDef
     &                + fEngWF(8)* restr1Eng
          engCOUL   = fEngWF(6)*engCOULR1 + fEngWF(7)*engCOULR2
          engPOTENT = eGeoDef + fEngWF(5)*engVDWR1 + engCOUL +
     &                molSolEn*fEngWF(9)
c
```

3. Propogation of the trajectory

For one time step propagation of the MD trajectory is done by the subroutine

```
c make mdStep
        call mdTimeStepProp(nmoveatom,moveAtomList,deltat)
c
```

which uses multi step leap-frog algorithm to calculate velocities and positions at time (t+deltat).

$$\mathbf{v}_i(t_n + \Delta t/2) = \mathbf{v}_i(t_n - \Delta t/2) + m_i^{-1}\mathbf{f}_i\ (t_n)$$
$$\mathbf{r}_i(t_n + \Delta t) = \mathbf{r}_i(t_n) + \mathbf{v}_i(t_n + \Delta t/2)\Delta t \qquad (34)$$

with different time steps for updating the short range (Δt) , long range ($2\Delta t$) and solvation forces ($4\Delta t$).

4. Temperature control - Berendsen thermostat method

At each time step the temperature control routine performs calculation of the total kinetic energy of the moving atoms. The relaxation the average temperature of the atomic system to the specified value are give via the *weak-coupling method* or Berendsen method, which scale the velocity by the factor lambTR(t)

$$V'_i(t) = V_i(t) * \texttt{lambTR}(t) \quad (35)$$

the velocity scaling describes energy exchange with bath thermostat with temperature relaxation time τ_T. The respective scaling factor is equal

$$\texttt{lambTR(t) = sqrt(1 + (tempTg-tempT0(t))/}\tau_T\texttt{)*(tempTg/tempT0 -1.0))} \quad (36)$$

where tempT0 is the effective temperature at the time=t, and tempTg is the target temperature to relax. The effective temperature tempT0(t) is defined by the all atomic velocities

$$T0(t) = \frac{1}{k_B N_{deg\,Freed}} \sum_{i=1}^{Nat} m_i V_i^2(t) \quad (37)$$

where $N_{degFreed}$ is the number degrees of freedom, k_B is the Boltzman constant. For proteins in water solvent a reasonable value of the temperature relaxation time τ_T is equal to 0.4-0.5 ps. The value of τ_T should be sufficiently small to achieve required temperature, but sufficiently large to avoid disturbance of the properties of protein by strong coupling to the temperature bath.

5. Trajectory writing

Trajectory is written for each nwtra time steps. The trajectory can be written for atomic positions (and for atomic velocietis) in the user specified file.

6. Docking Methods

Docking method is performed by subroutine runLigDock02 in the mdyn07 program procedure **runLigDock02** perform ab initio docking of molecular ligand of size up to ~100 atoms.

The algorithm flow can be described as

1) Calculation of the accessible surface of the protein. Calculation of a surface grid for probe sphere of radius ~ average atomic radius, and contact positions [**bindSiteAt01(*)**]with protein atoms. Calculation are done by subroutine **surf_SAS04**.

2) Calculation of a surface grid points for a probe ligand of radius of typical aromatic ring [benzene] **gridsizeSAS** ~ 3.0 A. The surface grid are calculated by clustering of surface contact positions **bindSiteAt01(*)** and the surface grid **bindGridXYZSAS01(*)** is generated. The contact score [**nsasGridPoint(*)**] equal to the number of contact atomic positions included in to the surface grid point **bindGridXYZSAS01(*)** is calculated.
The **bindGridXYZSAS01(*)** are sorted by descent of the contact score value **nsasGridPoint(*)** and presents an initial trial positions for refined docking of ligand.

3) Refined docking is performed via subroutine **runLigDock01**(ig,bindGridXYZSAS01loc). For each

initial positions **bindGridXYZSAS01(*)** for ligand center.
Procedure **runLigDock01** perform global optimization of ligand orientation and position in a restrained region of 3D-space. Spatial restraints are a sphere of radius equal to **gridsizeSAS**. Orientational optimization based on exhaustive search via optimization from different initial orientations uniformly covering all orientational space. The orientational optimization can be done in two mode. Coarse grain mode consist of 24 orientations with 90deg between two neighbor orientations, fine mode consist of 144 orientations with 45deg angle between two neighbor orientations. For each initial ligand orientation the molecular dynamic simulated annealing coupled with van der waals potential scaling is performed for flexible ligand and fixed protein atoms. A variant of deformable potential energy surface global optimization method is used. Three best final position/orientations of ligand are collected for each initial positions **bindGridXYZSAS01(*) in the files** LigDockFinMMM.nnn.pdb - where MMM - grid position number, nnn - 001,002,003 - orientations
The best docking variant for the ligand can be chosen as a file LigDockFinMMM.nnn.pdb with minimal potential energy engPOTENTLG.

Examples

1bty : benzamidine-trypsine complex

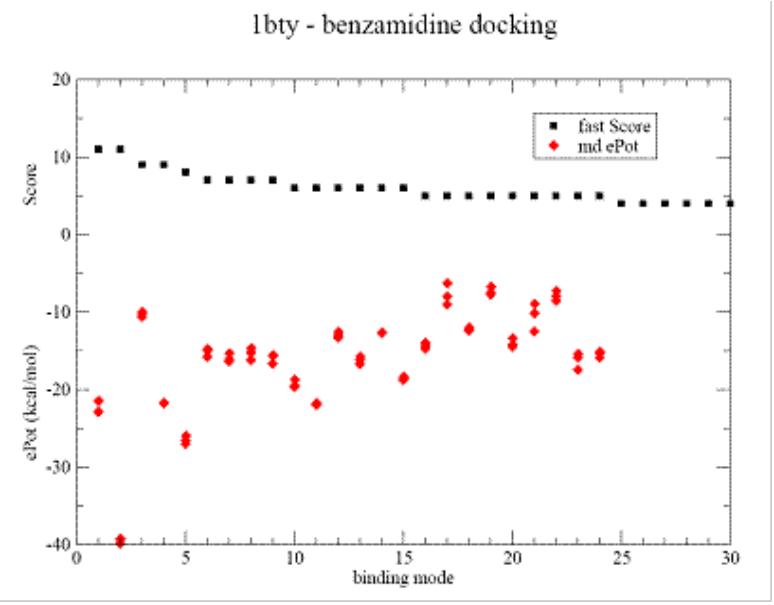
File	#LigBindGridOnSAS:		X	Y	Z	contactScore
ATOM	1	LBSt 1	16.536	26.130	8.764	11
ATOM	2	LBSt 2	29.319	14.972	16.378	11
ATOM	3	LBSt 3	6.595	15.454	32.366	9
ATOM	4	LBSt 4	28.049	26.396	3.572	9
ATOM	5	LBSt 5	37.370	14.662	29.278	8
ATOM	6	LBSt 6	9.605	28.662	39.481	7
ATOM	7	LBSt 7	18.280	35.574	15.402	7
ATOM	8	LBSt 8	30.648	34.679	44.060	7
ATOM	9	LBSt 9	34.040	33.767	21.484	7
ATOM	10	LBSt 10	5.056	19.922	18.987	6
ATOM	11	LBSt 11	25.308	5.865	13.437	6
ATOM	12	LBSt 12	13.241	31.812	30.019	6
ATOM	13	LBSt 13	6.174	15.317	15.623	6
ATOM	14	LBSt 14	15.230	11.995	39.322	6
ATOM	15	LBSt 15	42.858	27.966	33.933	6
ATOM	16	LBSt 16	39.046	14.805	5.421	5
ATOM	17	LBSt 17	24.676	37.002	14.221	5
ATOM	18	LBSt 18	39.100	25.116	6.122	5
ATOM	19	LBSt 19	25.156	6.498	5.813	5
ATOM	20	LBSt 20	14.736	13.757	2.279	5
ATOM	21	LBSt 21	35.933	31.703	11.547	5
ATOM	22	LBSt 22	45.035	21.844	22.099	5
ATOM	23	LBSt 23	12.210	8.874	28.161	5
ATOM	24	LBSt 24	11.197	11.080	32.573	5
ATOM	25	LBSt 25	25.549	16.554	-0.897	4
ATOM	26	LBSt 26	34.793	8.348	15.236	4
ATOM	27	LBSt 27	26.857	9.202	21.336	4
ATOM	28	LBSt 28	34.072	12.246	27.335	4

…..

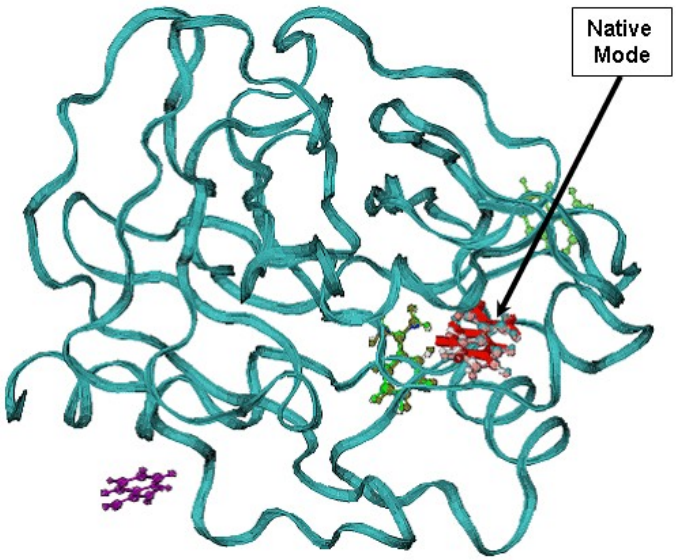
1) 1bty complex benzamidine on trypsine

Fig.1. Docking results for benzamidine on trypsine - 1bty complex.
A - contact Score (black square) for binding grid points vs refined potential energy of ligand

binding (red diamonds).



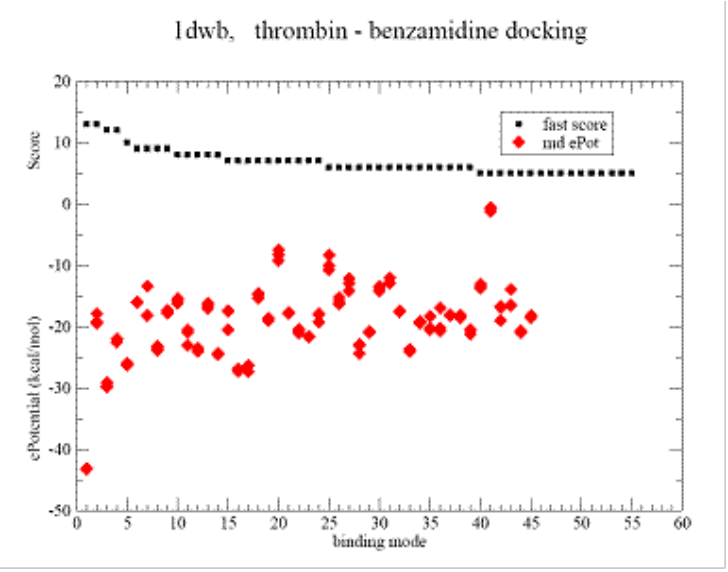
B - minimum energy docking mode (red bonds), RMSD = 0.54 A for all non Hydrogen atoms ligand of the native binding mode. CPK- green and violet are less favorable binding modes with low binding energy are shown in (**A**). CPK (pink) - native binding mode of benzamidine in 1bty.



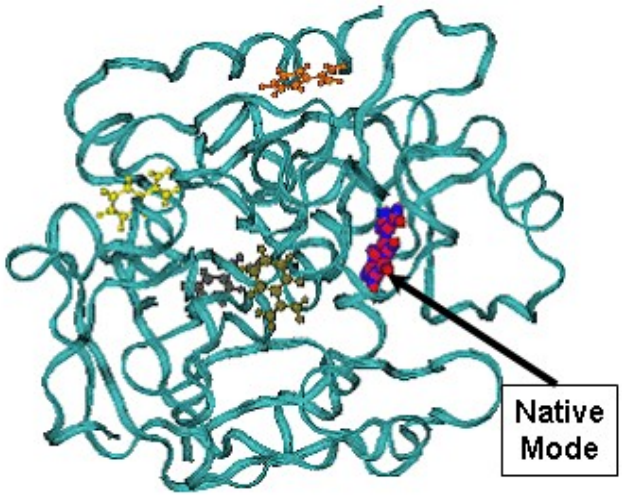
2) 1dwb : thrombin + benzamidine complex

Fig.2 Docking results for benzamidine on thrombin.

A - Contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B(CPK blue) - minimum energy docking mode. Less favorable binding modes are shown - yellow, brown, green. CPK- (red) native benzamidine binding mode in 1dwb complex,

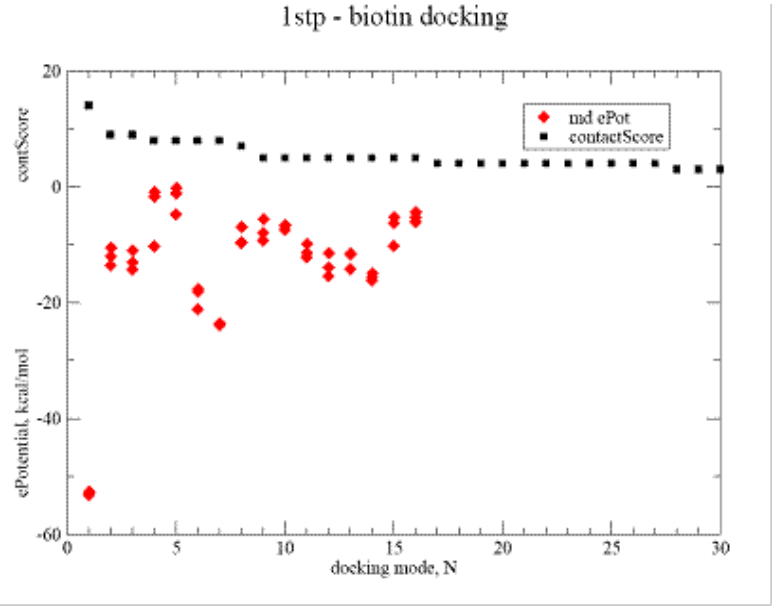


Minimum energy mode has RMSD = 0.27 A from the native binding mode of benzamidine.

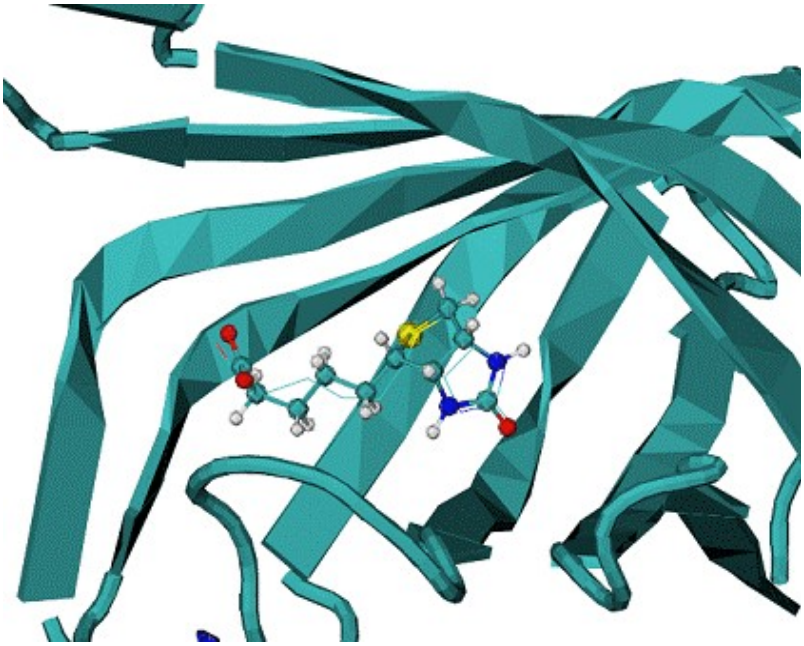
3) Biotine - streptavidine complex - 1stp

Fig.3. Docking result for biotine on streptavidine , 1stp complex.

A - contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B - minimum energy docking mode structure of biotine - CPK, lines - native biotine in the 1stp complex.

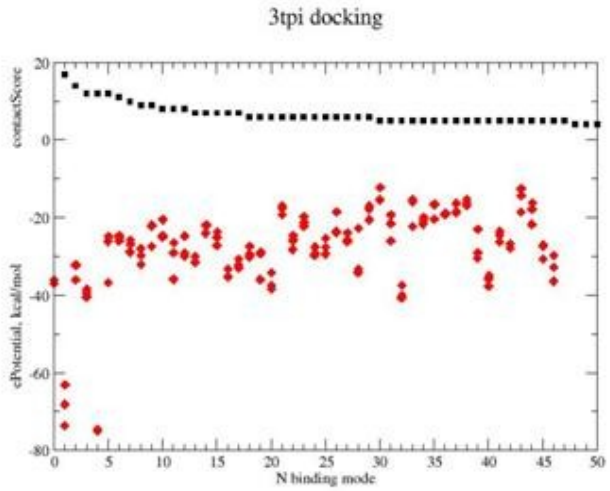


Minimum energy mode has RMSD = 0.96 A from the native binding mode of biotine.

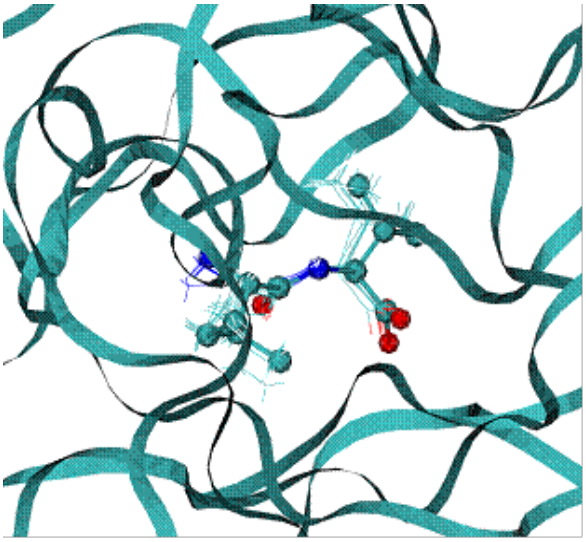
4) Trypsinogen/pancreatic trypsin inhibitor + Ile-Val peptide complex : 3tpi

Fig. 4. Docking result for ILE-VAL dipeptide on Trypsinogen/pancreatic trypsin inhibitor.

A - contact Score (black square) for binding grid points vs refined potential energy of ligand binding (red diamonds).



B - Lines are minimum energy docking modes of rank 1- 4 structures of ILE-VAL peptide - lines, CPK - native binding mode of biotine in the 1stp complex.



The best binding energy mode has RMSD = 0.46 A from the native binding mode of dipeptide ILE-VAL

Table 1. Energies of top ranked binding modes, and RMSD from the native binding mode.

Binding mode	**ePL, kcal/mole**	**RMSD, A**
Rank 1 - LigDockFin001.001.pdb	-76.07	0.46
Rank2 - LigDockFin001.002.pdb	-75.6	0.58
Rank3 - LigDockFin001.002.pdb	75.5	0.78
Rank4 - LigDockFin004.001.pdb	-74.8	0.88

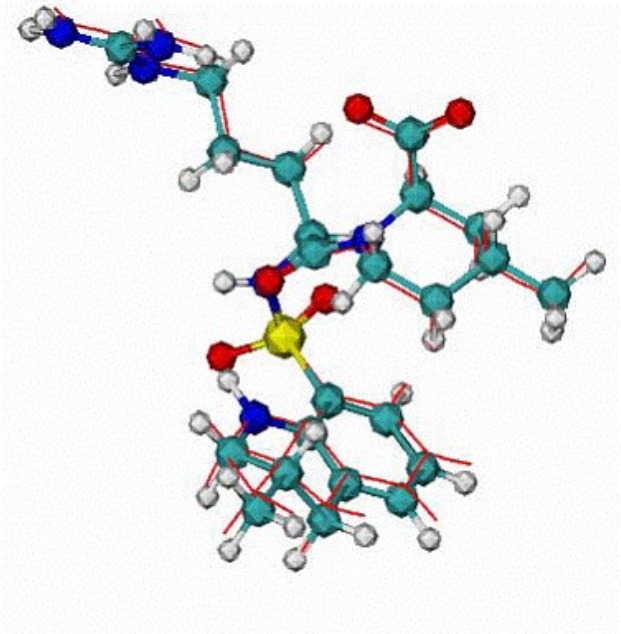
5) 1dwc complex of Human thrombin with thrombin-inhibitor MIT

Fig. 5. 1dwc complex of Human thrombin with thrombin-inhibitor MIT .

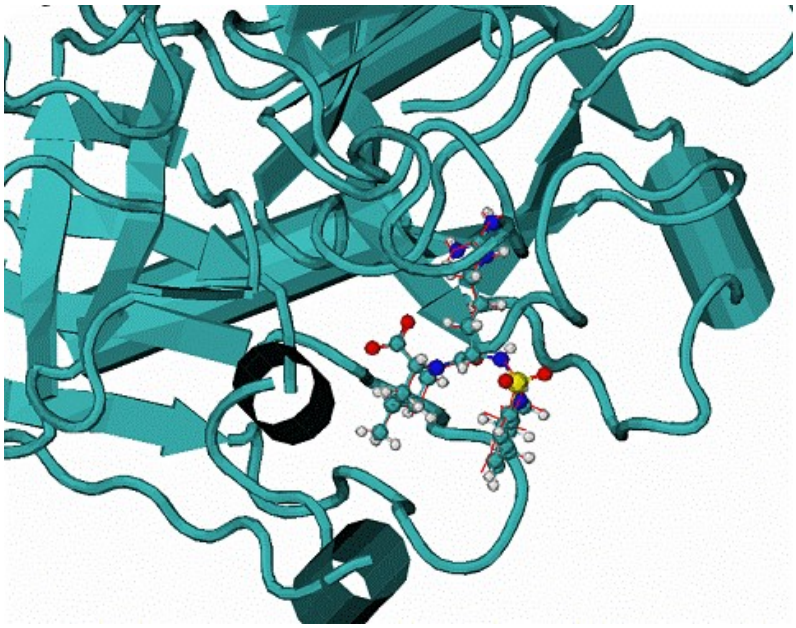
Human thrombin - 296 residues;

MIT - molecule includes 80 atoms

A - Top Ranked calculated docking mode - red lines, CPK - native MIT in the native binding mode, RMSD = 0.2 A for calculated docking mode from the native.



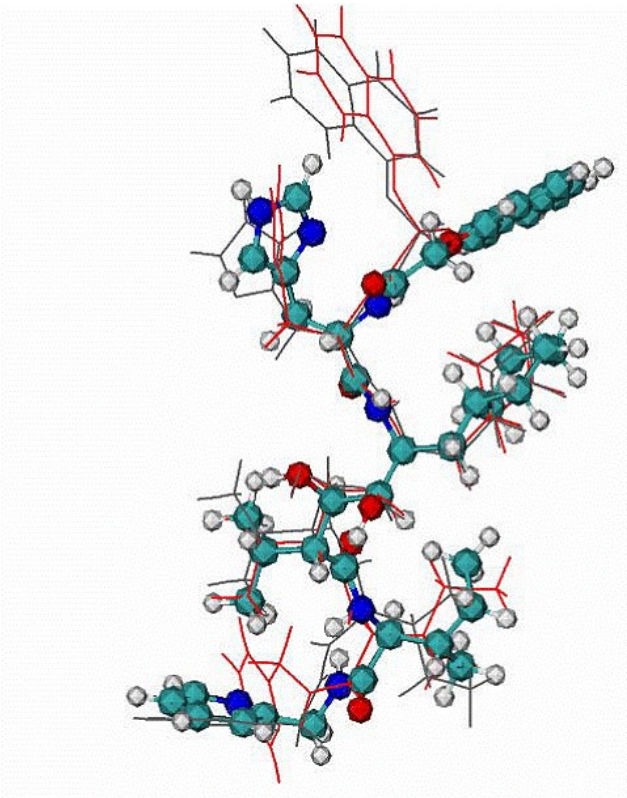
B - 1dwc complex. Red lines is docked MIT ligand, CPK is the native mode..



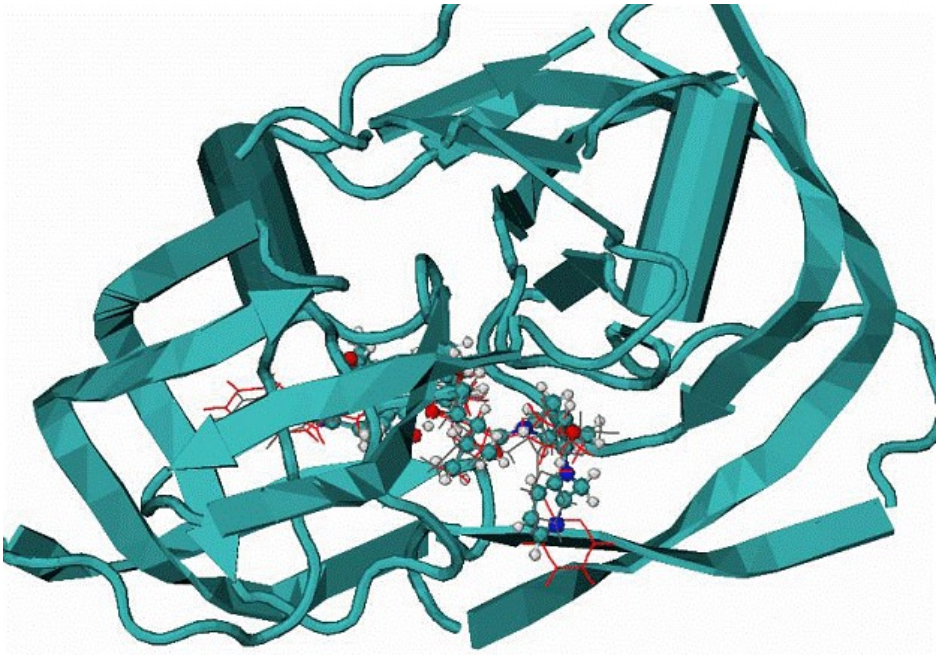
6) 1hiv complex of HIV1 protease with inhibitor NOA

Fig. 6. 1hiv complex of HIV1 protease with inhibitor NOA

A - Two top ranked calculated binding modes of NOA in comparison with the NOA ligand in the native binding mode of 1hiv complex. CPK - native binding mode, lines (red and grey) the top ranked mode by energy of binding. The RMSD from the native are ~3.1A for all atoms. The major difference between native and calculated modes are the orientation of one aromatic double-ring at the top of molecule NOA, the RMSD = 1.1. A over all atoms except the later aromatic system.



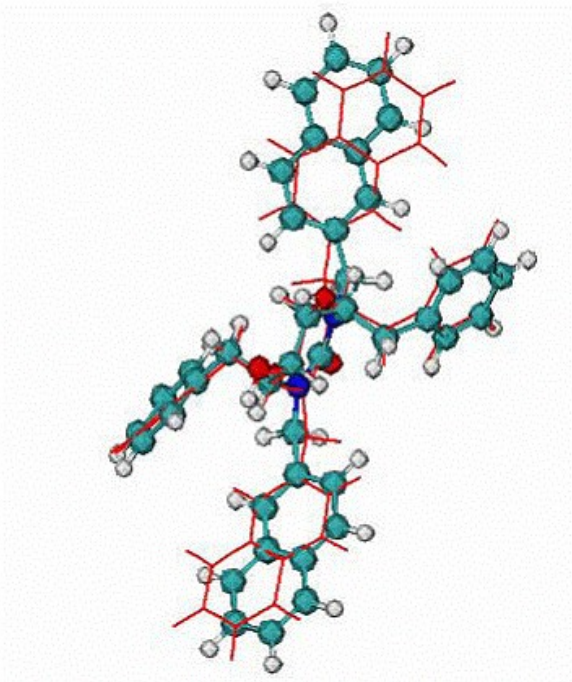
B - 1hiv complex of HIV1 protease with inhibitor NOA. CPK - native mode, red and grey lines - are calculated modes.



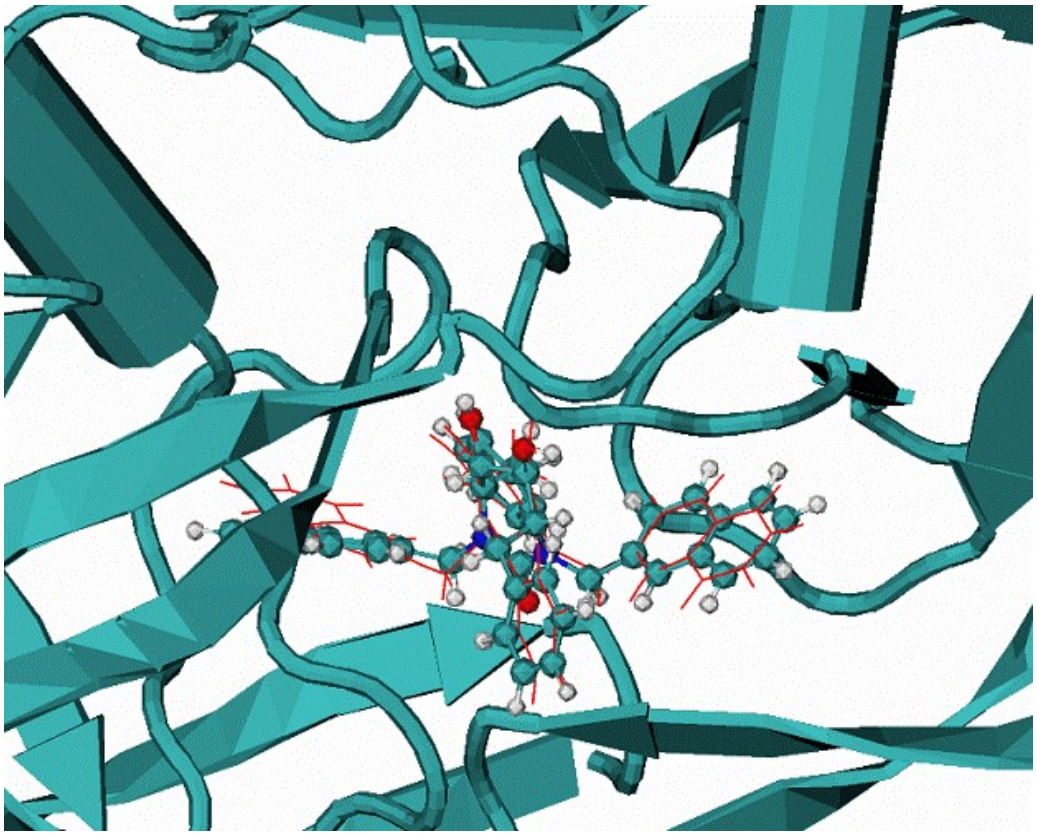
7) 1hvr complex of HIV1 protease with inhibitor XK2

Fig. 7. 1hvr complex of HIV1 protease with inhibitor XK2

A - Calculated binding mode of XK2, red lines, CPK - native binding mode of XK2 ligand. RMSD = 0.95 A for all atom.

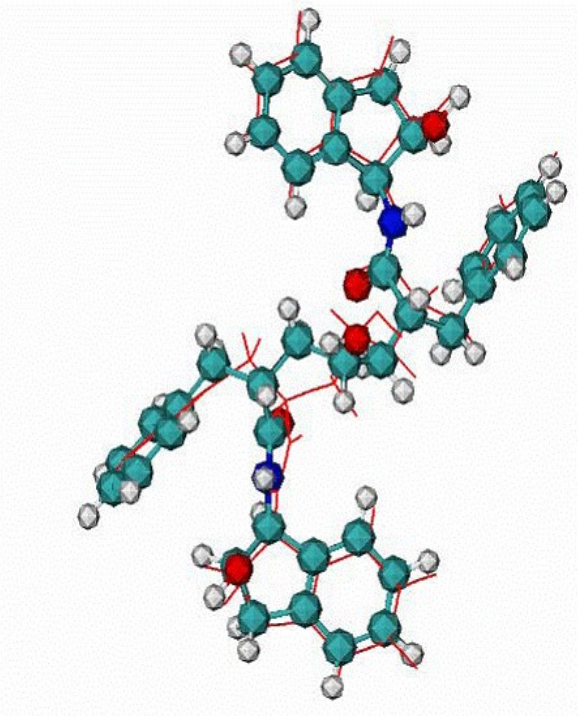


B - Calculated docking mode for the ligand XK2 in complex with HIV1 protease, CPK - the native binding mode of the XK2 ligand.



8) 1hvp complex of 1HIV protease with VAC molecule inhibitor

Fig. 8. 1hvp complex of 1HIV protease with VAC molecule inhibitor
A - Calculated best binding mode of VAC is in red lines, CPK - native VAC inhibitor in the 1hvp complex; the RMSD = 0.99 A.



B - 4hvp complex, red lines is the calculated mode, CPK - the native binding mode of VAC inhibitor.

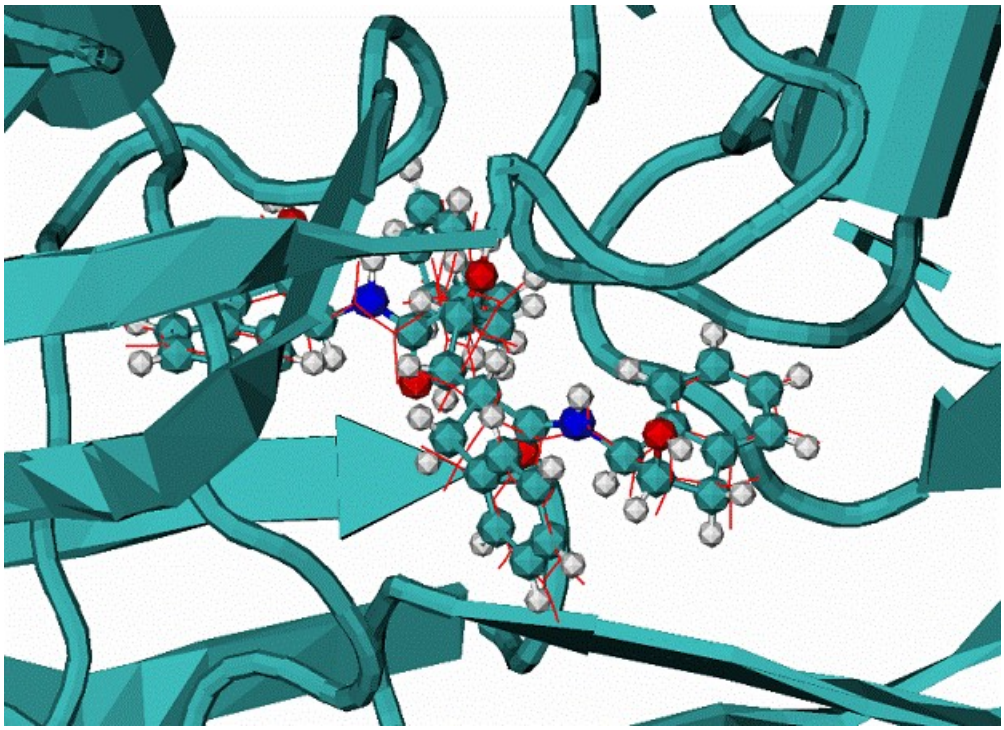


Table 1. Results of MdDock method for a set of complexes			
complex	**Ntors**	**RMSD, A**	**ΔEgap**
1) 1bty trypsin/benz	0	0.5	9.7

2) 1dwb α-thrombin/benz	0	0.5	13.3
3) 1stp streptavidine/biotin	5	0.96	29.5
4) 3tpi trypsinogen/Ile-Vla	6	0.42	10.6
5) 1dwc α-thrombin/MIT	8	0.2	10.8
6) 1hiv HIV1 protease/NOA	16	1.1/3.1	2.6
7) 1hvr HIV1 protease/XK263	8	0.95	39.1
8) 4phv HIV1 protease/VAC	15	0.9	3.4

Ntors - number of flexible torsion angles.

ΔEgap - energy gap between lowest energy binding mod and the next energy mode.

Conclusion:
The developed method of blind docking has show a good accuracy in prediction of the native bindig modes of flexible ligands. At the test set of 8 ligands the method shows 100% accuracy, i.e. the native binding mode are found as the mode with highest binding affinity.

References

Tamar Schlick. Molecular Modeling and simulation. Springer-Verlag, New York, 2000.
Cornell W.D., Cieplak P., Bayly C.I., Gould I.R., Mertz K.M., Ferguson D., Spellmeyer D.C., Fox T., Caldwell J.W., Kollam P.A. A second generation force field for the simulation of proteins, nucleic acids and organic molecules. *J.Am.Chem.Soc.* 1995: **117**, p.5179-5197
Lazaridis T., Karplus M. *Proteins: Structu, Funct., and Gen.* 1999: **35**, p.133-152

Parameters
Molecule name
Input file
PDB file
Info file
Detail log file
moveRes file
Restrain file
saProtocol file

Molecule name - molecule name myMolec. The name will be added to the left of all files generated by the program, I.e. sequence of molecular dynamics trajectory snapshot files myMolec_mdResXXXX.pdb, molecular dynamic trajectory energy file myMolec_engMd.tra, the final result of mdynSB rum file myMolec_mdXYZVfin.pdb

Input file

```
Input file. The  inProtocol file defines protocol of mdyn  calculations.
Default file name ./MdynPar.inp .
inProtocol file consist of sequense of lines. Line starts from keyWord [and its
value].
Example of inProtocol file:
#MdynPar.inp for HomologyModel refinement
```

```
#234567890123456789012345678901234567890!comment
$fullProtMD                                                    !
#$MovingRes
$harmAt1PosRst=0.25                                            !harmConst (kcal/A^2)
$Hread
$shake=2                                                       !0/1/2
$zeroRot
#$SolvateExWat=4.5                                            !ExplicitWaterShell 4.5A
#$SolvGS
$SolvWbrg
$SolvGBorn                                                     !SolvGBorn
#$mdRestart
$doMDyn
$MDSA                                                          !do SimAnnealing
$engCalc
$engOptim
$nOptStep=1                                                    !max N optim steps
$aSoftCore=1.0                                                 ! 1.0= standart VDW, < 1.0
-0.0-softCore
$initMDTemp=10.00                                              ! initial temperature in K
$bathMDTemp=50.00                                              ! bath termostat
temperature
$runMDnstep=2000                                               ! number mdyn time step to
run
$mdTimeStep=0.002                                              ! md time step
$NTV=1                                                         ! statistical ensemble
type NTV/NEV = 1/0
$nwtra=500                                                     ! write on HD protein
structur in pdb format for each nwtra mdstep
#
END
#
NOTE that parameter file formatted, i.e. $ sign should be in the firs position of
the line  No SPACE to assign value after keyword.
#
Description:
parameter file consists of lines starting from the $ simbol and keyWord
keyWord can be two types: logical  and digital
$MovingRes                 ! logical  required special file to define moving
RESidues list
$harmAt1PosRst=0.25        ! digital  NO SPACE to assign value for keyword

keyWord  switch on a respective modul of program,
some keyWord switch on moduls which in turn needs some special User defined
file to work properly.

KEYWORD DESCRIPTION
#234567890123456789012345678901234567890!comment

$fullProtMD                               !defines FULL (i.e. ALL atoms) of the USER
molecule
                                           will be free to move in energy relaxation
or molDyn
$MovingRes                                ! logical keyWord defines that ONLY  a
defined set of RESidue are free to move
                                            this keyWord is coupled with file -mv
moveRes in the argument line of
                                            the program mdynSB0
                                          default name for moveRes file is
```

```
./moveRes.inp
#example of ./moveRes.inp
#1arb
#aaaaaaIIIIiiii
#
MOVRES   1  10       !line defines first and last resudues of moving segment
MOVRES  45  76
MOVRES 115 260
end
************
$harmAt1PosRst=0.25  ! digital keyWord define RESidue segments with 1 atom position
harmonic restrants.
                                    0.25 = harmonic restrain Constant K
                                    restrEnergy = 0.5*K(r - r0)**2,
                                    the reference position r0 =
initialXYZinput.pdb - positions from
                                    the initial INPut PDB file which defines
INItial structure of molecule

                                    this keyWord is coupled with file -r
inRestrain of the argument line of
                                    the program mdynSB05
                                    default name for inRestrain file is
./restrAt1.inp

EXample of inRestrain file:
#harmonically restrained RESidue segments
#xxxxxIIIIiiiiaaAAAA
#(6x,2i4,a40)
RESTAT   1   63  PBB          ! line starts from keyWord RESTAT numbers=first/last
residue of segment
                               ! PBB (only protein backbone atoms are restrained,
i.e. side chains are free)
RESTAT  78 120  ALL            ! ALL (all atoms are restrained)
end
#
 ---------------------------------------------------

$Hread    ! defines that all Hydrogens will be read from input molecule structure
-c inPDB   file
          otherwise the ALL HYDrogens will be restored by the program
mdynSB05
          RECOMENDED: at the first run of a protein with unknown (or
partially known) Hydrogen atom.
                              start the mdynSB with off $Hread option, i.e.
                              #$Hread
-----------------------------------------------------
$shake=2    ! invoke shake subroutine to keep bonds fixed. shake=1  X--Hydr bonds,
(shake=2 all bonds) are fixed

-------------------------------------------------------------------------------
---------------------
$zeroRot   ! invoke procedure to stop overal rotation and translation of molecule

-----------------------------------------------------
$SolvateExWat=4.5   ! build explicit water solvation shell of 4.5 A around protein
molecule

-----------------------------------------------------
```

```
$SolvGS         ! invoke implicit  Gaussian Shell solvation model
$SolvWbrg       ! implicit  WaterBridges between polar atoms
$SolvGBorn      ! implicit Generalized Born model + SAS HydroPhobic solvation

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

$mdRestart    ! restart molDynamics from the last snapshot  mdXYZVfin.pdb
                the file mdXYZVfin.pdb should be copied to the file mdyn inRestart
file
                mdXYZVin.pdb

$doMDyn       ! do molecular dynamics
$MDSA         ! do Molecular Dynamical Simulated Annealing
              ! coupled with file -sa SAprotocol which define protocol of the
simulated annealing

Example of SAprotocol.inp  file
#SA protocol
#nSAstep
2
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#234567890x12345678x123456x123456x123456
#ntimeMX    tempTg  SCvdW wfHb128BB wfhB128BS
100000      500.0     0.8    1.0     1.0
100000      100.0     1.0    1.0     1.0
END
#
   ntimeMX - number of md timeStep
   tempTg  - target temperature in K, this temperature will be reach during ntimeMX
steps
   SCvdW   - parameter 0 - 1 to defile softness of the van der waals potential.
Soft potential
             modifies Potential Energy Surface decrease a barriers of
conformational transitions
   wfHb128BB, wfhB128BS - scaling factors for BackBone-BackBone and BackBone-
SideChain Hydrogen Bond energy
#-----------------------------------------------------

* * * * * * * * * * * * * * * * * * * * * * * *
-c inPDB   file  - standart pdb file
REMARK: PDB:
ATOM      1  N    GLY A   1      11.726 -10.369  10.598
ATOM      2  H1   GLY A   1      11.921 -11.015   9.807
ATOM      3  H2   GLY A   1      12.518 -10.395  11.271
ATOM      4  H3   GLY A   1      10.852 -10.663  11.079
ATOM      5  CA   GLY A   1      11.567  -9.015  10.090
ATOM      6  HA2  GLY A   1      10.772  -8.977   9.420
ATOM      7  HA3  GLY A   1      12.439  -8.710   9.612
ATOM      8  C    GLY A   1      11.280  -8.099  11.303
ATOM      9  O    GLY A   1      11.256  -8.584  12.493
ATOM     10  N    VAL A   2      11.060  -6.876  11.020
ATOM     11  H    VAL A   2      11.066  -6.574  10.025
etc.
TER                 ! CHAIN TERmination
ATOM   1302  N   GLY A  94      10.957 -15.678  12.832
ATOM   1303  H   GLY A  94      10.735 -14.663  12.877
ATOM   1304  CA  GLY A  94      10.193 -16.559  11.950
ATOM   1305  HA2 GLY A  94       9.428 -16.004  11.516
```

```
ATOM   1306  HA3 GLY A  94       9.784 -17.323  12.525
ATOM   1307  C   GLY A  94      11.016 -17.184  10.843
...
etc.
TER                ! CHAIN TERmination
END                ! file  END
* * * * * * * * * * * * * * * * * * * * * * * * *
```

PDB file - inPDB file Default name ./molec.pdb

Info file - OutPut file

Detail log file - OutPut file

moveRes file - moveRes file. User defined moving residue segments Default name ./moveRes.inp.

Restrain file

```
inRestrain file. Default name ./restrAt1.inp
# * * * * * * * * * * * * * * * * * * * * * * * *
# EXAMPLE
-r inRestrain ( ./restrAt1.inp )
#
User defined harmonically restrained RESidue segments. Atom positions are
harmonically restrained around initial positions (coordinates) with harmonic
constant defined in the ./MdydPar.inp  file
(6x,2i4,a40)
xxxxxxIIIIiiiiAAAAAAAAAA
RESTAT   1 119  PBB CA                : ProtBackBone CA restrained
RESTAT 131 175  ALL                   : ALL atoms restrained
RESTAT 191 216  ALL                   : ALL atoms
END
#
* * * * * * * * * * * * * * * * * * * * * * * * *
```

saProtocol file

```
saProtocol file . User defined protocol for simulated annealing molecular dynamics.
Default file name ./Saprotocol.inp
Example of SAprotocol.inp  file
#SA protocol
#nSAstep
2
#(f10.1,1x,f8.1,1x,3(f6.1,1x)
#234567890x12345678x123456x123456x123456
#ntimeMX    tempTg  SCvdW wfHb128BB wfhB128BS
100000      500.0     0.8    1.0     1.0
100000      100.0     1.0    1.0     1.0
END
#
   ntimeMX - number of md timeStep
   tempTg  - target temperature in K, this temperature will be reach during ntimeMX
steps
   SCvdW   - parameter 0 - 1 to defile softness of the van der waals potential.
Soft potential
             modifies Potential Energy Surface decrease a barriers of
conformational transitions
   wfHb128BB, wfhB128BS - scaling factors for BackBone-BackBone and BackBone-
SideChain Hydrogen Bond energy
```