VM2
Version 2.8.2

Quick Start: Keyword Reference

VeraChem LLC

VeraChem has been issued a patent **(USPTO Patent No. 8,140,268)** for the VM2 method.

Contact:

For information regarding VM2 software package licensing contact VeraChem LLC at sales@verachem.com

For technical support contact VeraChem LLC at support@verachem.com

For general enquiries contact VeraChem LLC at info@verachem.com

VM2 Package Keyword Listing

==========

1. Choice of System Type and Calculation Type and Other Top Level Control.

molSystemType

Choose the type of molecular system. There is no default; this option must be given. See below for additional input required dependent on this choice.

'protein'	Protein receptor calculation (could include explicit water, ions, etc.). Part of the system must be fixed in space (see Section 2).
'host'	Host molecule calculation. These should be 'small' receptor systems of a few hundred atoms or less e.g. cyclodextrins.
'ligand'	Ligand calculation; for example, a 'drug like' small molecule.
'protein+ligand'	Protein-ligand complex.
'host+ligand'	Host-guest complex.

calcnType

Choose type of calculation to be carried out. There is no default; this option must be given. All calculation types can be initiated with one or multiple input conformers.

'vm2'	VeraChem Second-generation mining minima (VM2) free energy calculation.
'feprocess'	Free energy processing of one or multiple conformers supplied by the user.
'confsearch'	Conformational search (potential energy only).
'rmsd'	Structural comparison of read-in conformers.
'filter'	Filter out repeats contained in read-in conformers.
'geomopt'	Geometry optimization.
'geomoptHatoms'	Optimize positions of just hydrogen atoms. Only allowed for molSystemType 'protein' and 'protein+ligand'.
'energy+grad'	Single-point energy and gradient.

‘energy’ Single-point energy.

timeLimit

Time limit for calculations given in wall clock hours. Currently only relevant for calcnType ‘vm2’. The program terminates cleanly and outputs all data files when the limit is projected to be reached in the next phase of a calculation. The default is 96.0 hours.

readInConfs

Optionally read in molecular conformations (one or more) from a text file or multiple text files to initiate a calculation. The text file formats may be **.xyz**, **.sdf**, Macromodel **.dat**, or **.crd**. This option may be used, for example, to read in a previously generated ensemble of ligand conformations to generate initial protein-ligand conformations, or simply to read in previously generated ensemble of protein-ligand conformations. If this option is not used a single starting conformation is taken from the input **.crd** coordinates – see Sections 2-4.

The readInConfs option may be given up to a maximum of **three** times, providing multiple types of conformer ensembles. For each instance of readInConfs multiple conformer source files may be read in. The program automatically makes appropriate combinations of conformer types read-in. For example, if molSystemType is ‘protein+ligand’ and if ‘complex’, ‘protein’, and ‘ligand’ conformer ensembles are read-in, the ‘complex’ conformers are taken as is and all unique combinations of the ‘protein’ and ‘ligand’ ensembles make additional ‘protein+ligand’ start conformers. The maximum number of start conformations is 1000. The program makes sensible truncations if the conformer files provided result in more.

‘complex’ Formatted file(s) containing protein-ligand or host-guest conformers.

‘protein’ Formatted file(s) containing only protein conformers.

‘host’ Formatted file(s) containing only host molecule conformers.

‘ligand’ Formatted file(s) containing only ligand conformers.

ligandConfsToCrd

Only relevant when using the readInConfs option to read in ‘ligand’ conformers. Controls how, if at all, read-in ligand conformers are superimposed on the ligand input .**crd** coordinates. (Note that the input **.crd** coordinates themselves can be moved *prior* to this by superimposition on template coordinates – see Section 4.)

‘no’ Use the coordinates of the ligand conformers as read-in. This is the **default**.

‘byConf1COG’ Translate the center of geometry (COG) of the first ligand conformer read-in to the COG of the ligand **.crd**. Apply the same translation to all subsequent ligand conformers read-in.

‘byConfsCOG’ Translate the COG of each ligand conformer read-in to the COG of the ligand **.crd**.

‘byConf1All’ Carry out a rotation/translation superposition of all heavy atoms (non hydrogens) of the first ligand conformer read-in on the corresponding ligand **.crd** atom positions. Apply the same rotation/translation to all subsequent ligand conformers read-in.

‘byConfsAll’ Carry out a rotation/translation superposition of all heavy atoms (non hydrogens) of the each ligand conformer read-in on the corresponding ligand **.crd** atom positions.

‘byConf1PairsMap’ Carry out a rotation/translation superposition of the first ligand conformer read-in with the ligand **.crd** coordinates using the atom indexes provided on the very next line. Apply the same rotation/translation to all subsequent ligand conformers read-in e.g.

byConf1PairsMap
3 5 18 21 22 23

‘byConfPairsMap’ Carry out a rotation/translation superposition of each ligand conformer read-in with the ligand **.crd** coordinates using the atom indexes provided on the very next line e.g.

byConfsPairsMap
3 5 18 21 22 23

useCrdAsTemplate

Only relevant when using the readInConfs option to read in ‘complex’ conformers plus another type of conformer (e.g. ‘protein’, ‘host’, or ‘ligand’) and molSystemType is protein+ligand or host+ligand (i.e. a complex). Controls whether to use the **.crd** input coordinates (see Sections 2-4) as a template for generation of complex conformers (‘yes’) or whether to use the coordinates of the first ‘complex’ conformer read-in as a template (‘no’).

‘yes’

‘no’ This is the **default**.

useCrdAsConf

Only relevant when using the readInConfs option. Controls whether to use the **.crd** input coordinates (see Sections 2-4) as a starting conformation in addition to the ones generated through readInConfs. Note that if readInConfs option is not used the **.crd** coordinates are *always* used to define a single starting conformation.

'yes'	This is the **default**.
'no'	

outputFormats

Choose any number of the following file formats. Currently **.xyz** and **.pdb** formats are always output in addition to those chosen. Place one per line directly following the keyword with no blank lines.

'sdf'	A structure-data file (SDfile) with standard V2000 or V3000 molfile formatting.
'mol2'	Tripos mol2 file.
'dat'	Macromodel data file.
'csv'	Comma-separated-values file containing energy data.
'gms'	Basic template input files for the GAMESS electronic structure software package.
'g09'	Basic template input files for the Gaussian09 software package.

fullEnergyBreakdown

Requests that for output of **.sdf** and **.csv** files a full breakdown of the energy into constituent terms is written out. If 'no' is selected a limited number of constituent energy terms are output.

'yes'	This is the **default**.
'no'	

splitOutputFormats

Mostly relevant for molSystemType ‘protein+ligand’ and ‘host+ligand’. The same as outputFormats above, but a separate formatted file is output for each of the molecules comprising the complex. Currently **.crd** files are always output in addition to those chosen, even for non-complexes. The base-name for the split output files is taken from the input **.crd** file names; a descriptor is added based on the calculation type e.g. xxxxx.vm2.sdf, xxxxx.vm2_rank1.crd. Place one output format type per line directly following the keyword with no blank lines.

‘sdf’	A structure-data file (SDfile) with standard V2000 or V3000 molfile formatting.
‘xyz’	Standard xyz file format.

limitConfsToOutput

The way that the number of conformers written to the formatted output files is limited can be chosen using this keyword.

‘byCount’	The user sets the maximum number of conformers to be output. Follow this line directly with an integer. This is the **default** with a maximum number of conformers set as **1000**.
‘byPopulation’	The user sets the maximum cumulative conformer population that limits the number of conformers output. Follow this line with a percentage value e.g. 99.9. Note that this option only makes sense for calcnType’s ‘vm2’ and ‘feprocess’.

atomsToOutput

This is relevant for systems that include proteins as not all the atoms are required to be present in calculations, and not all atoms present are mobile.

‘all’	All atoms are included in the formatted output. This is the **default**.
‘real’	Only ‘real’ atoms are included in the formatted output. (Real atoms are those atoms that are included in the energy calculation; however, they are not necessarily free to move.)
‘live’	Only live (flexible) atoms are included in the formatted output.

binaryFileRestart

Restart a calculation from a VeraChem binary data file. The binary file has the suffix **.vcbin**. The program expects the base name of the binary restart file to have the same base name of the **.inp** file.

'crashed'	Use when calculation quits unexpectedly. This option is currently only available for calcnType 'vm2'.
'extendRun'	Use for carrying out additional iterations of a calculation that finished, but, for example, did not converge. This option is currently only available for calcnType 'vm2'.
'reprocess'	Uses the conformations produced from a prior run as a starting point, but reprocesses them for energies, carrying out a geometry optimizations as necessary, and proceeds with the requested calculation. The user can change the energy potential (e.g. different solvation model) from the original run if desired. This option is currently only available for calcnType 'vm2'.
'textOutput'	Read a VeraChem binary data file and output the data as formatted text files (see outputFormats above.) This option is currently only available for calcnType 'vm2'.

Example usage 1

```
----------------------------------------------------------------
#
molSystemType
protein+ligand
#
calcnType
vm2
#
timeLimit
48.0
#
readInConfs
ligand
ligand_confs.xyz
#
outputFormats
sdf
csv
#
limitConfsToOutput
byPopulation
99.9
```

#

==========

2. Molecular System Definition Options for Protein Macromolecules

Relevant for molsystemType ‘protein’ and ‘protein+ligand’.

inputProtein		Names of input files containing protein system data and real/live set definition related data. They are mandatory and must be given in order with no blank lines.
	1	Signifies protein molecule one. A single protein molecule is the current limit.
	~/path/protein_name.crd	Starting coordinates, atom names, residue names etc. Files must conform to standard **.crd** format (regular or extended).
	~/path/protein_name.top	Topology and molecular mechanics parameters. See Section XII for format specification.
	~/path/protein_name.mol	Provides protein molecule bond orders and stereocenter information. File must be standard V2000 or V3000 mol format.
setChainIds		If present controls relabeling of protein chain and residue Ids given in the .crd file. Requires that the very next line contain an integer, or integers, corresponding to the count(s) of the last residue of each newly defined chain. Optionally the next line can provide the new chain Ids. If this second line is not present the defaults are A, B, C, … and so on. E.g. setChainIds 99 198 199 A B C
constructLiveReal		Controls how the protein real/live set is defined i.e. the protein atoms that are included in the energy calculation (real), and which atoms are also allowed to move in the calculation (live). The live set is a subset of the real set. This keyword is **mandatory**.

‘readIn’ Read in a formatted text file that defines the protein real/live set. See Section XII for format specification. The name of the file must be provided on the very next line e.g.

readIn
~/path/protein_real_live.txt

‘byTemplateCOGs’ Read in a template molecule’s atomic coordinates, from a **.crd**, **.xyz**, **.sdf**, **.mol**, **.pdb**, or Macromodel **.dat** formatted file, distances to this molecules center of geometry (COG) will define the protein real/live set. For example, use co-crystalized ligand coordinates. The name of the file must be provided on the very next line e.g.

byTemplateCOG
~/path/template_real_live.crd

‘byTemplateAtoms’ Read in a template molecule’s atomic coordinates, from a **.crd**, **.xyz**, **.sdf**, **.mol**, **.pdb**, or Macromodel **.dat** formatted file, distances to which will define the protein real/live set. For example, use co-crystalized ligand coordinates. The name of the file must be provided on the very next line e.g.

byTemplateAtoms
~/path/template_real_live.crd

‘byXYZ’ Cartesian coordinates to be used as a reference point to define the protein real/live set. The coordinates must be provided on the very next line e.g.

byXYZ
3.2345 5.7941 9.7745

The following are relevant for the constructLiveReal choices ‘byTemplateCOG’, byTemplateAtoms’, and ‘byXYZ’

realCutoffDist The **default** is 9.0 Angstroms. This cutoff is residue based. The distance is from any protein atom to any template molecule atom for option ‘byTemplate’ or to a single user defined point for option ‘byXYZ’. Any residue with an atom within this distance is ‘real’ i.e. its atoms are included in the energy calculation, but are not necessarily mobile.

liveCutoffDist The **default** is 7.0 Angstoms. This cutoff is atom based. The distance is from any protein atom to any template molecule atom for option ‘byTemplate’ or to a single user defined for option ‘byXYZ’. Any atoms within this

distance are ‘live’ i.e. mobile. They are subset of the ‘real’ set.

symmetrizeRealSet

If ‘yes’ multiple chains are present and are symmetric, based on exact matching of residue and atom names between chains, residues will be added to real set as necessary to make it symmetric.

‘yes’

‘no’ This is the **default**.

symmetrizeLiveSet

If multiple chains are present and are symmetric, based on exact matching of residue and atom names between chains, atoms will be added to live set as necessary to make it symmetric.

‘yes’

‘no’ This is the **default**.

Example usage 2

```
----------------------------------------------------------
#
inputProtein
1
~/path/protein_name.crd
~/path/protein_name.top
~/path/protein_name.mol
#
constructLiveReal
readIn
~/path/protein_real_live.txt
#
----------------------------------------------------------
```

Example usage 3

```
----------------------------------------------------------
#
inputProtein
1
~/path/protein_name.crd
~/path/protein_name.top
~/path/protein_name.mol
```

```
#
constructLiveReal
byTemplateAtoms
~/path/template_real_live.crd
#
realCutoffDist
8.0
#
liveCutoffDist
6.0
#
-------------------------------------------------------------
```

==========

3. Molecular System Definition Options for Host Molecules

Relevant or molsystemType 'host' and 'host+ligand'.

inputHost		Names of input files containing host molecule data. They are mandatory and must be given in order with no blank lines. The program checks they are present by examination of their suffixes.
	1	Signifies that names of formatted data files for host molecule 1 will follow. Currently, one 'molecule' is the limit; however, a system comprising two hosts could still be run by including the data for both host molecules in each file.
	~/path/host_name.crd	Starting coordinates, atom names, etc. Files must conform to standard .crd format (regular or extended).
	~/path/host_name.top	Topology and molecular mechanics parameters. See Section XII for format specification.
	~/path/host_name.mol	Provides host molecule bond orders and stereocenter information. File must be standard V2000 or V3000 mol format.

Example usage 4

```
-------------------------------------------------------------
#
inputHost
1
~/path/host_name.crd
~/path/host_name.top
~/path/host_name.mol
```

#

==========

4. Molecular System Definition Options for Ligand Molecules

Relevant or molsystemType 'protein+ligand' and 'host+ligand' and 'ligand'.

inputLigand		Names of input files containing host molecule data. They are mandatory and must be given in order with no blank lines.
	1	Signifies that names of formatted data files for ligand molecule 1 will follow. Currently, one ligand molecule is the limit.
	~/path/ligand_name.crd	Starting coordinates, atom names, etc. Files must conform to standard **.crd** format (regular or extended).
	~/path/ligand_name.top	Topology and molecular mechanics parameters. See Section XII for format specification.
	~/path/ligand_name.mol	Provides ligand molecule bond orders and stereocenter information. File must be standard V2000 or V3000 **.mol** format.
placeLigandMethod		Controls how, if at all, the ligand will be moved from the **.crd** starting coordinates given above before the start of a calculation by placement relative to a user supplied position in space or template set of coordinates. (Note: Calculation of center of geometry (COG) excludes hydrogen atoms, as does the least squares fit for superpositions.) The moved ligand coordinates then redefine what the 'input' **.crd** coordinates are.
	'none'	The ligand is not moved from the starting coordinates defined in **.crd** above. This is the **default**.
	'byReceptorCOG'	Only relevant for molSystemType's 'protein+ligand' and 'host+ligand'. The receptor's (protein or host) center of geometry (COG) is used as a reference point that the ligand COG is translated to.

‘byXYZ’	Cartesian coordinates to be used as a reference point that the ligand center of geometry (COG) is translated to, and the *very* next line after that must contain the Cartesian coordinates, e.g.
.	byXYZ 3.2745 5.7654 9.7653
‘byTemplateCOG’	Read in a template molecule, **.crd**, **.xyz**, **.sdf**, **.mol**, **.pdb**, or Macromodel **.dat** format, and use its center of geometry (COG) as a reference point that the ligand COG is translated to. For this option the *very* next line must contain the name of a formatted file containing the template e.g.
	byTemplateCOG ~/path/template_molecule.xyz
‘byTemplateAll’	Read in a template molecule, **.crd**, **.xyz**, **.sdf**, **.mol**, **.pdb**, or Macromodel **.dat** format, and superimpose all heavy atoms of the template onto the ligand atoms. The template should be a conformer of the same ligand defined by the starting coordinate **.crd** file above, with atoms in the same order. For this option the *very* next line must contain the name of a formatted file containing the template e.g.
	byTemplateAll ~/path/template_conformer.sdf
‘byTemplatePairsMap’	Read in a template molecule, **.crd**, **.xyz**, **.sdf**, **.mol**, **.pdb**, or Macromodel **.dat** format, and superimpose the ligand by chosen pairs of atoms to map onto each other. For this option the *very* next line must contain the name of a formatted file containing the template, the following line must contain the template atom indexes for use in superposition, and the subsequent line must contain the corresponding ligand atom indexes e.g.
	byTemplatePairsMap ~/path/template_molecule.crd 7 8 9 10 11 12 13 3 5 11 15 19 20 21
doSnapTemplatePairs	If ‘yes’ a harmonic potential (see below) is applied to the ligand atoms defined by the ‘byTemplatePairsMap’ setting above, but at the position of the template atoms. This

guides/snaps the chosen ligand atoms to the template positions during conformational searches/geometry optimizations. Only relvent when placeLigandMethod option 'byTemplatePairsMap' is used.

'yes'

'no' This is the **default**.

snapTemplatePairsFC Relevant when doSnapTemplatePairs is 'yes'. Sets the harmonic potential force constant. The default value is 2.0 Kcal/mol/Angs.

Example usage 5

```
----------------------------------------------------------
#
inputLigand
1
~/path/ligand_name.crd
~/path/ligand_name.top
~/path/ligand_name.mol
#
placeLigandMethod
byTemplateCOG
~/path/template_molecule.xyz
#
----------------------------------------------------------
```

==========

5. Math Related Options.

randomSeedsMethod

Choose method to generate seeds for the KISS random number generator. Random number generation is required for various stochastic algorithms in the VeraChem computational chemistry package.

'byWallClock' Uses wall clock timing data combined with process ID data to automatically generate a different set of seeds every run. Note that for parallel runs a different seeds are produced for each process, but only the master process's set is written to output files. This is the **default**.

'byUser' The seeds are supplied by the user (see below). This option must be used if deterministic parallel processor runs are required.

setRandomSeeds

For ‘byUser’ option above include this keyword and supply four integers in the following four lines.

Example usage 6

```
----------------------------------------------------------
#
randomSeedsMethod
byUser
#
setRandomSeeds
9759
9850
7072
203
#
----------------------------------------------------------
```

==========

6. VeraChem Mining Minima VM2 Calculation Options.

Relevant for calcnType ‘vm2’.

convTolVm2

Specifies the free energy difference between VM2 iterations that signifies convergence. At least 3 iterations must have been carried out and the free energy must have gone down compared to the last 2 iterations. The **default** is 0.01 Kcal/mol.

maxVm2Iters

Specifies the maximum number of VM2 iterations to be carried out before quitting whether converged or not. The **default** is 60.

Example usage 7

```
----------------------------------------------------------
#
convTolVm2
0.001
#
```

maxVm2Iters
30
#

==========

7. General Conformational Search Control Options.

Relevant for calcnType ‘vm2’ and ‘confsearch’.

The VeraChem conformational search capability comprises various vibrational mode-distort-minimize types as well as rigid body translation-rotation distort-minimize algorithms. The ‘canned’ search styles use various combinations of these algorithms suitable for specific chemical system-based search demands. For fine control of these algorithms a ‘custom’ search may be requested (see Section 9).

Iteration and convergence control: only relevant for calcnType option ‘confsearch’.

convTolConfsearch

Specifies the potential energy difference between confsearch iterations that signifies convergence. At least 3 iterations must have been carried out and the potential energy must have gone down compared to the last 2 iterations. The **default** is 0.01 Kcal/mol.

maxConfsearchIters

Specifies the maximum number of confsearch iterations to be carried out before quitting whether converged or not. The **default** is 60.

Search methods control: relevant for calcnType options ‘vm2’ and ‘confsearch’.

confSearchStyle

Specifies the style of conformational search to be carried out. **Note:** See Section 9 for default ligand box constraint settings associated with confSearchStyle settings.

‘standard’	Requests the standard single-mode based sampling of conformational space. The quickest ‘canned’ search style, but will not consistently find the lowest energy conformers of a system, so use with caution.
‘enhanced’	Requests an enhanced sampling of conformational space. In addition to the single-mode based sampling, search drivers built from random combinations of pairs of single modes

	are used. Usually appropriate when the approximate pose/position of the ligand is known – for example by superposition on a ligand with the same scaffold that was co-crystallized with the receptor. This is the **default**.
'rigorous'	Requests a rigorous sampling of conformational space. Useful when the active/binding site is known, but the receptor and/or ligand itself may be quite flexible with large R groups etc. As well as single-mode and random-pair-modes searches, it includes searches using focused drivers where fewer torsions are included in each driver, but distortions tend to be more pronounced.
'vrigorous'	Requests a very rigorous sampling of conformational space. Useful when the active/binding site is known, but nothing is known about the pose and position of the ligand in the active/binding site. Large translations and rotations are included in the search as well as mode distortions.
'confgen1'	This setting is designed solely to generate a diverse set of conformations for starting points in other calculations. It carries out only one vm2/confsearch iteration and uses stricter than default filtering and expanded energy cutoff to achieve diversity of structures as opposed to energy convergence.
'confgen2'	Relevant for molSystemType 'ligand' only. The same process as 'congen1' above, but in addition the resulting conformers are rotated about their 3 principal axes 180 degrees. The 4-fold expanded set of conformers then have some orientational as well conformational diversity.
'confgen3'	Placeholder – ongoing implementation.
'confgen4'	Relevant for molSystemType 'ligand' only. The same process as 'congen1' above, but in addition a maximum of 20 of the resulting conformers are randomly rotated about their 3 principal axes between 0 and 360 degrees to generate 1000 final conformations. This provides large orientational diversity. For use when no information on the ligand pose is known.
'custom'	All search methods and parameters can be finely controlled according to the user's choice. Combinations of the many available conformational search options can be employed. Recommended for expert users who want detailed control of the search procedures. See custom search control parameters in Section 8 below.

confGenLengthSort

Only relevant for molSystemType ‘ligand’ calculations with confSearchStyle ‘confgen1’, ‘confgen2’, and ‘confgen3’. If ‘yes’ ligand conformers are sorted according to their length (longest first) before any rotomers are generated and conformers output.

‘yes’ This is the **default**.

‘no’

maxSearches

The maximum number of searches for each mode-distort-minimize search type strung together to form the search style. The **default** is 400. This may be automatically adjusted downwards for small systems. It may also be automatically adjusted for MPI parallel runs for load balancing.

modeRotnMax

The maximum rotation angle for a mode distortion.
The **default** is 180.0 (degrees).

switchToRandomRotnMax

The *‘vm2’* or *‘confsearch’* iteration at which the maximum rotation angle for mode distortions is randomly chosen from the range modeRotnMax/2 to modeRotnMax. The **default** is 7.

numRlsearch

The number of random ligand fixed-body translation-rotation searches to be carried out. Only relevant when a ‘vrigorous’ search style is requested or when a random ligand rotation/translation search is requested through the custom search option. The **default** is 24.

ligandTranMax

The maximum ligand fixed-body translation distortion length.
The **default** is 2.0 (Angstroms).

ligandRotnMax

The maximum angle for ligand fixed-body rotation distortions.
The **default** is 180.0 (degrees).

excludeBackBone

Only relevant for systemType ‘protein’ and ‘protein+ligand’. If ‘yes’ the protein

backbone atoms are excluded from drivers for conformational searches; if ‘no’ the protein backbone atoms are included in mode-distort conformational searching. Note that regardless, live (mobile) backbone atoms are always included in geometry optimizations after mode distortions.

‘yes’ This is the **default**.

‘no’

excludeSideChains

Only relevant for systemType ‘protein’ and ‘protein+ligand’. If ‘yes’ the protein sidechain atoms are excluded from drivers for conformational searches; if ‘no’ the protein sidechain atoms are included in mode-distort conformational searching. Note that regardless, live (mobile) sidechain atoms are always included in geometry optimizations after mode distortions.

‘yes’

‘no’ This is the **default**.

excludedAtomsFile

Optionally specify a text file that provides a list of atoms to be excluded from drivers for conformational searches. See Section XII for format.

~/path/file_name_excluded_atoms.txt

forceConstCutoff

Mode drivers with force constants larger than this cutoff are excluded from the mode search. The **default** is 5000.0.

deltaLevel1Cutoff

Relevant when there is a level 2 correction to the level 1 energy e.g. single –point energy with PBSA solvation model at geometry determined with GB solvation model. For level 1 energy differences between the lowest energy conformer and the conformer just found that are greater than this cutoff, the level 2 energy correction is skipped and the current conformer discarded. The **default** is 20.0 Kcal/mol.

nonBlockingUpdate

This keyword is only relevant for MPI multi-processor runs. If ‘yes’, non-blocking sends and receives are used to communicate low energy structures between MPI processes every *‘vm2’* or *‘confsearch’* iteration; if ‘no’, blocking collective operations are used, which can result in large latencies.

‘yes’ This is the **default** for systemType ‘protein’, ‘protein+ligand’, ‘host’, and ‘host+ligand’.

‘no’ This is the **default** for systemType ‘ligand’.

doLoadBalance

This keyword is only relevant for MPI multi-processor runs. If ‘yes’, the MPI process that finishes its assignment of searches first in each *‘vm2’* or *‘confsearch’* iteration signals all other processes to proceed when their current mode distort-minimize is complete. This results in some skipped searches, but improves load balancing considerably.

‘yes’ This is the **default** for systemType ‘protein’, ‘protein+ligand’, ‘host’, and ‘host+ligand’.

‘no’ This is the **default** for systemType ‘ligand’.

mixSearchBasis

This keyword and the following four related ones are only relevant for MPI multi-processor runs. Periodically, multiple conformers are used as a basis for independent (i.e. decoupled) conformational searching, with no communication between MPI processes. This adds diversity to the conformational search. The number of conformer starting structures equals the number of MPI processes. (see mixSearchPicks below).

Integer 0, 1 to 4	0	sets this option as off
	1	Use multiple conformers every call to the conformational search i.e. every vm2 or confsearch iteration.
	2	Use multiple conformers every second vm2/confsearch iteration. This is the **default**.
	3	Use multiple conformers every third vm2/confsearch iteration.
	4	Use multiple conformers every fourth vm2/confsearch iteration.

mixSearchIters

Relevant if concurrent conformer searching is on (i.e. if mixSearchBasis above is not 0). Sets the vm2/confsearch iteration above which concurrent searching is completely switched off. The **default** is 20.

mixSearchPicks

Controls how the group of conformers is selected for the 'mixSearchBasis' approach.

'inorder'	Select N conformers in order of their free energy as the set of conformers to search on, where N is the number of MPI processes.
'random1'	Select the first N/2 conformers in order, then pick an additional N/2 at random from all the remaining conformers.
'random2'	Select the first N/2 conformers in order, then pick an additional N/2 at random from the next poolSize – N/2 conformers in order of their free energy. See below for poolSize. This is the **default**.
'cluster'	Select the first N/2 conformers in order, then cluster the remaining conformers starting at N/2 + 1 with an RMSD cutoff of 0.5 Angstroms. Pick the lowest energy conformer of each cluster up to N MPI processes. If not enough clusters present select from the lowest energy conformer up again (to double search the low energy conformers).

doClusterBy

Controls whether clustering (mixSearchPicks 'cluster' option) is based on RMSDs of the whole molecule system or a component. For example, for a protein+ligand complex the clustering can be set as based solely on the ligand RMSDs.

'complex'	The **default** if molSystemType is 'protein+ligand' or 'host+ligand'.
'receptor'	The only option if molSystemType is 'protein' or 'host'. Can also be selected for 'protein+ligand' or 'host+ligand' runs.
'ligand'	The only option if molSystemType is 'ligand'. Can also be selected for 'protein+ligand' or 'host+ligand' runs.

poolSize

For mixSearchPicks option 'random2' option, sets the size of the pool of conformers that are picked from at random. The **default** is 64**.** For the first iteration of a VM2 run when starting conformers are read in (see Section 1.) the default is quadrupled to allow a more diverse search basis. For 'random1' and 'cluster' options it is hardwired as all available conformers; for option 'inorder' it

is hardwired as the number of MPI processes.

relaxNonDriverAtoms

If ‘yes’, when carrying out distortions along drivers, non-driver atoms are allowed to relax after each distortion step via a few geometry optimization cycles (driver atoms are kept fixed during these cycles). If ‘no’ is selected all non-driver atoms are kept fixed in space during distortions. Note that enforcing rigidity during driver distortions will speed up the search, but will invariably result in extremely high energies for small driver distortions limiting the conformational space sampled.

‘yes’ This is the **default**.

‘no’

Example usage 8

```
------------------------------------------------------------
#
confSearchStyle
vrigorous
#
maxSearches
200
#
numRlsearch
48
#
excludedAtomsFile
~/path/file_name_excluded_atoms.txt
#
mixSearchBasis
2
#
mixSearchPicks
random2
#
------------------------------------------------------------
```

==========

8. Custom Conformational Search Options.

Relevant for calcnType ‘vm2’ and ‘confsearch’.

Use these options when keyword confSearchStyle is set to ‘custom’.

Search

Choose the type of search to be carried out.

‘mode’ Initiates a search using distortions along mode based drivers followed by geometry optimization. The nature of the mode-based search can be further controlled by the options below. This is the **default.**

‘ligand’ Initiates a ligand based search where the ligand is translated, and/or rotated followed by a geometry optimzation of the system. The ligand based search can be further controlled by the options described below.

‘combined1’ Requests a mode based search followed immediately by a ligand based search.

‘combined2’ Requests a ligand based search followed immediately by a mode based search.

modeSearch

Choose the type of mode search to be carried out.

‘normal’ A standard mode search with distortions along drivers weighted according to mode coefficients. This is the **default**.

‘focused’ A more robust mode search with more focused and larger distortions. This style of mode search cannot be applied to ligand only systems.

‘combined1’ Requests a standard mode search directly followed by a robust mode search i.e. ‘normal’ then ‘focused’.

‘combined2’ Requests a robust mode search directly followed by a standard mode search i.e. ‘focused’ then ‘normal’.

mode

For a ‘normal’ search (see above), choose how to determine geometry displacements i.e. drivers.

‘single’ Use individual modes only. This is the **default**.

‘pair’ Use a linear combination of randomly chosen pairs of modes (generated on the fly).

‘combined1’ Carry out a ‘single’ mode search directly followed by a ‘pair’ mode search.

‘combined2’ Carry out a ‘pair' mode search directly followed by a 'single' mode search.

focusedSearch

For a ‘focused’ search (see above), choose ligand driven, receptor driven, or a combination of the two.

‘ligand’ Ligand driven focused search only. All receptor atom and any small ligand mode coefficients are zeroed out. Distortions are then focused on small groups of ligand atoms.

‘receptor’ Receptor driven focused search only. All ligand atom and any small receptor mode coefficients are zeroed out. Distortions are then focused on small groups of receptor atoms.

‘combined1’ Carry out a ‘ligand’ driven focused search directly followed by a ‘receptor’ driven focused search. This is the **default**.

‘combined2’ Carry out a ‘receptor’ driven focused search directly followed by a ‘ligand’ driven focused search.

ndrivers N Number of drivers N to select from the total available (only applicable to ‘single’ mode generated drivers).

-1 Select all available drivers i.e. N is set equal the total number of drivers generated. This the **default**.

drivers

Determines how the drivers are chosen or ordered.

‘largest’ Pick N drivers in order of the largest number of coefficients > |0.1|. This is the **default.**

‘random’ Randomly pick N drivers.

‘bottom’ Pick the N drivers with the smallest eigenvalues.

‘middle’ Pick N drivers from the middle range of eigenvalues.

‘top’ Pick the N drivers with the largest eigenvectors.

binRandomPairs

For searches with random pairs of modes if ‘yes’ the possible pair combination

are binned and the algorithm will pick equally from all the bins; if 'no' totally random pair combinations are used.

'yes' This is the **default** for host involved systems and ligand only systems.

'no' This is the **default** for protein involved systems.

modeDistMaxE

Specify the energy change cutoff for mode distortions. The **default** is 2000.0 (kcal/mol).

ligandSearch

Choose the type of ligand search to be carried out.

'systematic' Requests a systematic ligand search. Rotations of +/- ligandRotnMax/4, ligandRotnMax/2, and ligandRotnMax degrees (see ligandRotnMax, Section 7) and translations of +/- ligandTranMax/4, ligandTranMax/2, and ligandTranMax Angstoms (see ligandTranMax, Section 7) of the ligand about and along its principal axes are carried out in small steps. Between each step a few geometry relaxation steps are carried out for the receptor. Combined translation-rotations are also carried out giving a total of 80 searches per dimension searched. The number of dimensions searched is controlled by sligandSearch (see below). The preceding distances and angles are limits, and the rotation or translation is stopped at any step that results in an energy change greater than ligandDistMaxE (see below). After stopping each rotation or translation, a full geometry optimization is carried out.

'random' Requests a search involving random translations and rotations of the ligand along and about its principal axes. Rotation limits are +/- ligandRotnMax and translation limits are +/- ligandTranMax. The number of dimensions searched is controlled by rligandSearch (see below). Again, distortions are stopped if an energy change greater than ligandDistMaxE occurs. A geometry optimization is carried out after each distortion. The number of searches is controlled by numRlsearch (see Section 7 above).

'combined1' Requests a systematic ligand search directly followed by a random ligand search.

'combined2' Requests a random ligand search directly followed by a systematic ligand search.

sligandSearch

Number of dimensions in which to carry the systematic ligand search.

‘1d’ Rotation about the principal axis with the smallest principal moment of inertia, followed by full geometry optimization. Then translation along the same axis again followed by geometry optimization. Then translation-rotation along the same axis followed by geometry optimization. This is the **default**.

‘2d’ Carry out '1d' rotations as above, then do the same for the axis with the second largest principal moment of inertia. Then move onto the translations, then onto translation-rotations.

‘3d’ All principal axes are tried in the same manner as above.

rligandSearch

Number of dimensions in which to carry the random ligand search plus control of the procedure.

‘1d’ Random translations and rotations along and about the principal axis with the smallest principal moment of inertia, followed by full geometry optimization. This is the **default**.

‘2d’ Carry out ‘1d’ as above, then do the same for the axis with the second largest principal moment of inertia i.e. separate geometry optimization for each axis trans/rots.

‘3d’ All principal axes are tried in the same manner as above.

‘comb2d’ Combines the random translations and rotations along and about two principal axes *before* the geometry relaxation step.

‘comb3d’ Combines the random translations and rotations along all principal axes *before* the geometry relaxation step.

ligandDistMaxE

Specify the energy change cutoff for ligand rotation/translation distortions. The **default** is 10000.0 (kcal/mol).

Example usage 9
Custom search settings that reproduce the confSearchStyle setting ‘vrigorous’ described above in Section 7.

```
------------------------------------------------------------
#
Search
combined1
#
```

modeSearch
combined1
#
mode
combined1
#
sdriver
1
#
ndrivers
-1
#
drivers
bottom
#
modeDistMaxE
2000.0
#
ligandSearch
combined1
#
sligandSearch
3d
#
rligandSearch
comb3d
#
ligandDistMaxE
10000.0
#

==========

9. Options and Control of Spatial Boundary Based Conformer Rejection.

Relevant for calcnType ‘vm2’ and ‘confsearch’.

These options allow conformers that do not fit the users predetermined geometric criteria to be discarded during a conformational search. They allow, for example, protein-ligand conformations where the ligand may have left the region of the known binding pocket to be discarded, or for conformers in which explicit water molecules that move too far away from a known crystallographic position to be discarded. These region-based exclusions can be used in conjunction with or be replaced by energy-based constraints applied during geometry optimizations (see Section 14).

boxedAtoms

integer1 integer2 integer3 ….

An integer or list of integers that specifies an atom or atoms (other than ligand atoms) to apply a spherical boundary to; for example, an explicit water molecule oxygen atom. The center of geometry of the atoms in the list is only allowed to move in a sphere of specified dimension (see below), if it moves outside the sphere the conformation is rejected. Atoms on the list are also fixed in space during mode distortions. The reference center is defined by the input **.crd** coordinates of specified atoms. This option may be given up to twenty times i.e. the spherical box 'constraint' may be applied to twenty separate groups of atoms. Each spherical box may apply to a maximum of 200 atoms.

atomBoxSize

Specify the radius of the sphere that the 'boxedAtoms' center of coordinates must remain in. The **default** is 1.0 (Angstroms). If the 'boxedAtoms' center of coordinates moves outside this sphere the conformation is rejected.

ligandBoxSize

Specify the radius of the sphere in Angstroms that the ligand center of coordinates must remain in. If the ligand center of coordinates moves outside this spherical box the conformer is rejected. The reference center is defined by the input **.crd** coordinates of the ligand. To turn this filter off set as -1.0. The **default** is -1.0 (off) for molSystemType 'host+ligand'. For all other molSystemTypes, the **default** radius depends on the confSearchStyle: for 'custom', 'standard', and 'enhanced' it is 1.0 Angstroms; for 'rigorous' it is 2.0 Angstroms; for 'vrigorous' it is 4.0 Angstroms.

Example usage 10

```
----------------------------------------------------------
#
boxedAtoms
32 35
#
atomBoxSize
2.0
#
ligandBoxSize
2.0
#
----------------------------------------------------------
```

==========

10. Options for Free Energy Processing of Conformers.

Relevant for calcnType ‘vm2’ and ‘feprocess’.

modeScanning — Allows the mode scanning step in the calculation of the configuration integral to be turned on or off.

- ‘on’ — This is the **default**.
- ‘off’

temperature — Temperature in Kelvin used in the calculation of configurational integrals. The **default** is 300.00.

freeEnergyPreFactor

Control which atoms are used in the calculation of the free energy prefactor. Only relevant for protein involved calculations.

- ‘useLiveAtoms’ — Use only the ‘live’ atoms.
- ‘useRealAtoms’ — Use all ‘real’ atoms. This is the **default**.

Example usage 11

```
------------------------------------------------------------
#
modeScanning
off
#
temperature
273.15
#
------------------------------------------------------------
```

==========

11. Stereochemistry Checking and Enforcement Control.

Relevant for calcnType ‘vm2’, ‘confsearch’, ‘feprocess’, and ‘geomopt’.

maintainCisTrans

If ‘yes’ cis/trans arrangements across double bonds are enforced by rejecting conformers where isomerization has occurred; if set as ‘no’ cis/trans isomerization is allowed. Double bonds are as identified by the bond orders given in the input mol/sdf file; Cis/trans arrangements across double bonds are identified automatically.

‘yes’ This is the **default**.

‘no’

maintainParity

If ‘yes’ R/S stereocenters are enforced by rejecting conformers where stereoisomerization has occurred. If set as ‘no’ stereoisomerization is allowed. R/S stereocenters are as defined in the input mol/sdf file.

‘yes’ This is the **default**.

‘no’

maintainProteinPepBonds

Control the stereochemistry of protein peptide bonds by rejecting generated conformers that violate the chosen option.

‘asInput’ The stereochemistry of protein peptide bonds are maintained as they are in the user provided input structure. This is the **default**.

‘asTrans’ An attempt will be made to flip any cis protein peptide bonds found in the input structure and all peptide bonds will then be maintained as trans. **This option is not yet functional.**

‘no’ Protein peptide bond isomerization is allowed.

Example usage 12

```
----------------------------------------------------------
#
maintainCisTrans
yes
#
MaintainParity
yes
#
MaintainProteinPepBonds
asInput
#
----------------------------------------------------------
```

==========

12. Control of Filtering Out Conformer Repeats.

Relevant for calcnType ‘vm2’, ‘confsearch’, ‘feprocess’, ‘rmsd’, and ‘filter’.

These parameters set energy difference cutoffs and geometry RMSD cutoffs that control how similar two conformers have to be for one of them to be designated a repeat and discarded. Additionally, energy parameters that control the culling of ‘high energy’ conformers can be set.

preFilterCalcnType

Choose type of calculation to be carried out prior to filtering. Only relevant for calcnType ‘filter’.

‘geomopt’	Geometry optimization. This is the **default**.
‘energy+grad’	Single-point energy and gradient.
‘energy’	Single-point energy.
‘none’	No calculation before filtering.

pairCutoff1

Used in the filtering conformers either read in or resulting from a conformational search that *have not* undergone free energy processing. It is the bonded-term-energy difference below which a pair of conformers will be geometrically compared. The **default** for calcnType ‘vm2’ is 0.5 Kcal/mol; for calcnType’s ‘filter’, ‘rmsd’, ‘confsearch’, the **default** is 2.0 Kcal/mol.

pairCutoff2

Used in the filtering conformers either read in or resulting from a conformational search that *have* undergone free energy processing (relevant for calcnType’s ‘vm2’ and ‘feprocess’). It is the bonded-term-energy difference below which a pair of conformers will be geometrically compared. The **default** is 1.0 Kcal/mol.

pairRmsdCutoff1

Used in the filtering conformers either read in or resulting from a conformational search that *have not* undergone free energy processing. It is the geometric RMSD lower than which the conformer with the higher potential energy is discarded. The **default** for calcnType ‘vm2’ is 0.2 Angstroms; for calcnType’s ‘filter’, ‘rmsd’, and ‘confsearch’ the **default** is 0.3 Angstroms.

pairRmsdCutoff2 — Used in the filtering conformers either read in or resulting from a conformational search that *have* undergone free energy processing. It is the geometric RMSD lower than which the conformer with the higher free energy is discarded. The **default** is 0.3 Angstroms.

firstConfCullE — Energy cutoff used for initial culls e.g. the first 2 VM2 iterations. Depending on the calculation type and status, it is the conformer potential energy or free energy relative to the current lowest energy conformer at which all higher energy conformers are discarded. The **default** is 20.0 Kcal/mol except for calcnType's 'filter' and 'rmsd' when the **default** is 100.0 Kcal/mol.

ConfCullE — Standard energy cutoff used for culling high energy conformers. Depending on the calculation type and status, it is the conformer potential energy or free energy relative to the current lowest energy conformer at which all higher energy conformers are discarded. The **default** is 10.0 Kcal/mol except for calcnType's 'filter' and 'rmsd' when the **default** is 100.0 Kcal/mol.

displaceCurrentConfs

Only relevant for the molSystemType's 'protein' and 'protein+ligand'. If 'yes' during the filtering process a newly generated conformer found to be a repeat of a currently established conformer, which also has a lower energy (this energy difference will always be very small i.e. a fraction of a kcal/mol) will displace the currently established conformer. In some cases with this will lead to very small energy fluctuations between iterations and therefore very slow convergence, therefore the default is set as 'no'.

'yes'

'no' — This is the **default**.

Example usage 13

```
----------------------------------------------------------------
#
pairCutoff1
0.2
#
pairCutoff2
0.3
#
firstConfCullE
```

30.0
#
ConfCullE
20.0
#

==========

13. Options for Molecular Alignment and RMSD Calculation.

Relevant for calcnType ‘vm2’, ‘confsearch’, ‘feprocess’, ‘rmsd’, ‘filter’, and ‘geomopt’. For calcnType ‘rmsd’ a set of conformers must be read-in via the readInConfs keyword - see Section 1.

Currently alignment options are only relevant for molsystemType ‘ligand’, ‘host’, and ‘host+ligand’. For molsystemType ‘protein’ and ‘protein+ligand’ no alignment will be carried out regardless of user input as protein real-fixed atoms are already exactly aligned and provide the reference position and orientation for the whole system.

The alignment options allow the conformations produced during the course of a particular calculation to be superimposed on the input conformation for output. The default for the molsystemType’s listed above is for alignment to be turned on. Unless the user wants to specify the specific atoms to align, e.g. when there is a suitable ligand scaffold, the defaults picked by the program are usually appropriate.

preRmsdCalcnType

Choose type of calculation to be carried out prior to RMSD calculation. Only relevant for calcnType ‘rmsd’.

‘geomopt’	Geometry optimization. This is the **default**.
‘energy+grad’	Single-point energy and gradient.
‘energy’	Single-point energy.
‘none’	No calculation before filtering.

preRmsdFilter

If ‘yes’ filter the read-in conformers before calculation of RMSD. Only relevant for calcnType ‘rmsd’.

‘yes’

‘no’ This is the **default**.

rmsdAllPairsMethod

Choose symmetry aware method to calculate and output the RMSD between *all* pairs of conformers that remain after any filtering. Only relevant for calcnType ‘rmsd’.

‘symaware1’ Basic fast symmetry aware algorithm. This is the **default**.

‘symaware2’ More sophisticated and expensive symmetry aware algorithm – see *J. Chem. Inf. Comput. Sci.* **44**, 1301-1313 (2004). Not available for molSystem ‘protein’ and ‘protein+ligand’

‘none’ Only RMSDs between the Rank 1 conformer and the rest are calculated using the basic symmetry aware method.

confAlignment

‘none’ Turn alignment off.

‘receptor’ The **default** for molsystemType ‘host’ and ‘host+ligand’ runs.

‘ligand’ The **default** for molsystemType ‘ligand’ runs.

‘selectatoms’ Indicates that the user will provide specific atoms to use for alignment.

numAlignAtoms Number of atoms the user will provide for alignment.

N

atomsToAlign Integers identifying which atoms to align.

integer1 integer2 integer3 interger4 …

Example usage 14

```
#
confAlignment
selectatoms
#
numAlignAtoms
11
#
```

atomsToAlign
10 16 21 18 20 12 19 17 15 7 24
#

==========

14. Geometry Optimization Options and Control, Including Constraints.

Relevant for calcnType ‘vm2’, ‘confsearch’, ‘feprocess’, and ‘geomopt’.

The following control convergence criteria, geometry optimization methods, and maximum allowed geometry steps to achieve convergence.

maxAtomGrad	Standard convergence criterion. Used, for example, for calcnType ‘geomopt’ or ‘feprocess’ runs or for final geometries after mode distortion. It is the maximum absolute value gradient allowed of any individual mobile atom in the system. A second criterion is that the whole mobile system gradient RMSD must also be less than 1/3 of this parameter. The **default** is 0.001 (Kcal/mol)/Angstrom.
maxAtomGradLoose	Loose convergence criterion. Used, for example, for an initial geometry optimization after a mode distortion. It is the maximum absolute value gradient allowed of any individual mobile atom in the system. As above, the whole mobile system gradient RMSD must also be less than 1/3 of this parameter. The **default** is 0.01 (Kcal/mol)/Angstrom.
doPreoptSteps	Do some initial geometry steps before a first full geometry optimization is attempted. During pre-optimizations steps any atom gradients above 100.0 Kcal/mol/Angstrom or below -100.0 Kcal/mol/Angstrom are set to +/- 100.0 Kcal/mol/Angstrom are damped. This is useful for initial starting structures where there may be close contacts.
‘yes’	Turn this option on. This is the **default**.
‘no’	Turn this option off.
preoptMethod	Method to use for the pre-optimization geometry steps.
‘1’	Quasi-Newton geometry optimization algorithm.
‘2’	Conjugate-gradient geometry optimization algorithm. This is the **default**.

maxPreoptSteps	Maximum number of pre-optimization geometry steps. The **default** is 100.
geomoptMethod	Method to use for geometry optimization.
'1'	Quasi-Newton geometry optimization algorithm. This is the **default**.
'2'	Conjugate-gradient geometry optimization algorithm.
maxGeomoptSteps	Maximum number of geometry steps allowed for a geometry optimization. The **default** is 5000.
batchEnergyCutoff	This energy cutoff overrides the ConfCullE cutoffs in Section 12. The default is large so when the user supplies a wide range of conformers for geometry optimization less are discarded and can be examined via formatted output files. The **default** is 10000.0 Kcal/mol.

The following apply constraints to selected atoms in the system so they do not move far away from a desired position during a geometry optimization.

tetheredAtoms	File that identifies atoms in the system that will be tethered. Multiple groups can be defined with each group being subject to different constraints defined by the harmonic and polynomial tether related keywords that follow below. The file name is arbitrary. See Section XII for format specification.
	~/path/tethered_atoms_file.txt

tetherForceConstant

Specify a force constant if a harmonic constraint is required.

To specify a polynomial constraint the following three options with no blank lines are required to give the polynomial function E(dr) = A*(dr/R)**n.

---

tetherScalingFactor

Real number A

tetherDistance

Real number R

tetherOrder

Real number n

---

nfreezeAtoms — Number of 'live' atoms to freeze in space during a geometry optimization by simply zeroing out their gradient. Currently, it is recommended that this option is not used for calcnType 'vm2' or 'feprocess'.

freezeAtoms — List of integers that identify which atoms to freeze.

integer1 integer2 integer3 integer4 ….

Example usage 15

```
----------------------------------------------------------------
#
maxAtomGrad
0.001
#
maxAtomGradLoose
0.01
#
doPreoptSteps
yes
#
preoptMethod
2
#
maxPreoptSteps
400
#
geomoptMethod
1
#
maxGeomoptSteps
10000
#
tetheredAtoms
~/path/tethered_atoms_file.txt
#
# Constrained Group 1
#
tetherScalingFactor
```

100.0
tetherDistance
0.25
tetherOrder
12.0
#
 # Constrained Group 2
#
tetherScalingFactor
1.0
tetherDistance
0.5
teherOrder
12.0
#

==========

15. Molecular mechanics potential energy calculation: methods and usage control

level1mmMethod

Choose the method to treat mm solvation for energy derivative based calculations i.e. energy+grad calculations, geometry optimizations, and hessian calculations. Currently, straightforward use of the defaults is suggested. Control and selection of parameters for the methods themselves is described in Sections 16-19 below.

‘gb’ This is the **default**. Use a Generalized Born solvation method.

‘cd’ Use a constant dielectric solvation model.

‘dd’ Use distant dependent dielectric solvation model.

level2mmMethod

Choose the method to treat mm solvation for single-point energy corrections applied to, for example, any molecular geometries determined using level1mmMethod. For calcnType ‘energy’ and ‘energy+grad’ this single-point energy will be applied to the input structure(s). Control and selection of parameters for the methods themselves is described in Sections 16-19 below.

‘pbsa’ This is the **default**. Use the Poisson-Boltzmann Surface-Area (PBSA) solvation model.

‘none’ The PBSA energy correction will not be carried out. Only level 1 energies will be used.

allowZeroWaterLJ Controls whether Lennard-Jones parameters for water hydrogen atoms will be allowed to be zero – as they are in OPLS.

'yes' Zero value parameters are allowed.

'no' Zero value parameters are not allowed and are replaced with TIP3P parameters. This is the **default**.

allowZeroLJ Controls whether Lennard-Jones parameters for non-water hydrogen atoms will be allowed to be zero – as they are in OPLS for polar hydrogens.

'yes' Zero value parameters are allowed.

'no' Zero value parameters are not allowed and are replaced with:

CHARMM/Dreiding: $\varepsilon_i = -0.046$
$r_j^{min}/2 = 0.2245$

AMBER/GAFF: $\varepsilon_i = -0.0157$
$r_j^{min}/2 = 0.6$

OPLS: $\varepsilon_i = -0.03$
$r_j^{min}/2 = 0.2806$

This is the **default**.

mmAddFxdFxdConst

Controls whether the fixed-fixed real atom constant energy terms e.g. bond, angle, dihedral, improper, vdW, pure Coulomb (not GB solvation pairs) are calculated once at the start of a calculation and added as corrective constants throughout the calculation. Addition of these terms may facilitate energy comparisons with other programs.

'yes' Calculate the fixed-fixed constant energy terms. This is the **default**.

'no' Do not calculate the fixed-fixed terms.

Example usage 16

```
------------------------------------------------------------
#
level1mmMethod
gb
#
level2mmMethod
pbsa
#
```

==========

16. Molecular mechanics Generalized Born (GB) solvation model control

gbSolvationModel

Choose the particular GB model used.

'still97'	Use Still's analytical method for calculating the approximate Born radii for use in the GB solvation energy expression. See Qiu, Hollinger, and Still, J. Phys. Chem. A **1997**, 101, 3005-3014. This is the **default**.
'hawkins96'	Currently disabled due to ongoing reimplementation work.

still97ParamSet

Choose the P1-P5 scaling parameters for still97 GB solvation energy calculations.

'still'	Use the original scaling parameters from *J. Phys. Chem. A* **1997**, 101, 3005-3014. This is the **default**.
'gilson'	Use an alternative set of scaling parameters. See David, Luo, and Gilson, *J. Comput. Chem.* **2000**, 21, 295-309.

gbDielectricExt

External solvent dielectric used in the GB solvation model. The **default** value is 80.0, modeling bulk water.

gbDielectricInt

Internal (i.e. solute) dielectric used in the GB solvation model. The **default** value is 1.0.

gbCavityRadii

Choose the atomic cavity radii to use in the GB solvation model.

'halfRmin'	Use Rmin/2, where Rmin is the force field Lennard-Jones parameter, except for hydrogen atoms bonded to hetero atoms, which are set to 1.15 Å, and covalently bound fluorine atoms, which are set to 2.00 Å. This is the **default**, with the only exception being CHARMM combined with 'still97' *and* still97ParamSet option 'gilson' (see 'legacy' option below).

‘halfSigma’ Use σ/2, where σ is the force field Lennard-Jones parameter, except for hydrogen atoms bonded to hetero atoms, which are set to 1.15 Å, and covalently bound fluorine atoms, which are set to 2.00 Å.

‘bondi’ Use the Bondi van der Waals radii. See Bondi, A., *JPC* **1964,** *68,* 441.

‘mbondi’ Use the modified Bondi radii. See Rizzo, Aynechi, Case and Kuntz, *J. Chem. Theory Comput.* **2006,** *2,* 128-139.

‘legacy’ Use Rmin/2, where Rmin is the force field Lennard-Jones parameter, except for hydrogen atom radii, which are all set to 1.20 Å. This is the **default** for gbSolvationModel ‘still97’ and still97ParamSet ‘gilson’.
Note: These are the radii used in all preceding versions of the VM2 software package i.e. version 2.1 and earlier, regardless of the force field and model.

Example usage 17

```
----------------------------------------------------------------
#
gbSolvationModel
still97
#
still97ParamSet
still
#
gbCavityRadii
legacy
#
----------------------------------------------------------------
```

---

17. Molecular mechanics constant (CD) dielectric solvation model control

cdSolventDielectric

Solvent dielectric constant used in the constant dielectric solvation model ‘mm-cd’. The **default** value is 80.0.

==========

18. Molecular mechanics distance dependent (DD) dielectric solvation model control

ddCoefficient

Coefficient used in the distance dependent dielectric solvation model 'mm-dd'. The **default** value is 4.0 resulting in the so-called 1/4r method.

==========

19. Molecular mechanics Poisson Boltzmann Surface Area (PBSA) solvation model control

pbDielectricExt

External solvent dielectric used in the PBSA solvation model. The **default** value is 80.0 modeling bulk water.

pbDielectricInt

Internal (i.e. solute) dielectric used in the PBSA solvation model. The **default** value is 1.0.

pbsaCavityRadii

Choose the atomic cavity radii to use in the PBSA solvation model. Currently the same radii are used for calculation of the electrostatic solvation energy (PB) and the non-polar solvation energy (SA). **Note:** If the 'still97'/'gilson' GB solvation model is being used, to match GB and PBSA cavity radii the 'legacy' option below must be explicitly selected.

'halfRmin' Use Rmin/2, where Rmin is the force field Lennard-Jones parameter, except for hydrogen atoms bonded to hetero atoms, which are set to 1.15 Å, and covalently bound fluorine atoms, which are set to 2.00 Å. This is the **default**.

'halfSigma' Use σ/2, where σ is the force field Lennard-Jones parameter, except for hydrogen atoms bonded to hetero atoms, which are set to 1.15 Å, and covalently bound fluorine atoms, which are set to 2.00 Å.

'fitted' Use atomic cavity radii fitted to reproduce solvation energies determined using explicit TIP3P water molecules and the AMBER force field. See Tan, Yang, and Luo, *J. Phys. Chem. B* **2006,** *110,* 18680-18687. For GAFF atoms i.e. non-peptide atoms, 'mbondi' radii are used.

'bondi' Use the Bondi van der Waals radii. See Bondi, A., *JPC* **1964,** *68,* 441.

'mbondi' Use the modified Bondi radii. See Rizzo, Aynechi, Case and Kuntz, *J. Chem. Theory Comput.* **2006,** *2,* 128-139.

‘legacy’	Use Rmin/2, where Rmin is the force field Lennard-Jones parameter, except for hydrogen atom radii, which are all set to 1.20 Å. **Note:** These are the radii used in all preceding versions of the VM2 software package i.e. version 2.1 and earlier, regardless of the force field and model.

sasaProbeRadius

Set the solvent accessible surface area (SASA) probe radius. The **default** value is 1.4 Angstroms.

==========

IX. Ligand example

1. CHARMM pathway using Discovery Studio Visualizer (DSV)

1.1. Get mol2 data file for chosen molecule: ibuprofen

Step 1: Go to, for example, the ZINC database website http://zinc15.docking.org and perform a search for 'ibuprofen'.

Step 2: Placeholder

1.2. Load molecule into DSV

Step 1: Placeholder

2. CHARMM pathway using the web user interface CHARMMing

2.1. Get mol2 data file for chosen molecule: ibuprofen

Step 1: Go to, for example, the ZINC database website http://zinc15.docking.org and perform a search for 'ibuprofen'.

Step 2: Placeholder

2.2. Load the molecule

Step 1: Placeholder

X. Protein-ligand example: HIV-1 protease and 38 inhibitors

This is a full example of setup, execution of calculations, and collection of binding affinity results for a protein plus ligand series: the target protein is human HIV-1 protease and there are 38 ligands in the inhibitor series. (*1*)

NOTE: You will need a working installation of AmberTools with the $AMBERHOME environment variable set to carry out the full procedure as described below. Please see http://ambermd.org/ to download AmberTools and for its documentation. (You can skip the setup section by going straight to Section 2. and making use of the "-d reference" option, described in Sections 2.1.2. and 2.2.2.)

First, untar the examples file vcCompChem_2_8_examples.tar.bz2, which is provided with the package:

tar xvf vcCompChem_2_8_examples.tar.bz2

The main directory for this example is:

vcCompChem_2_8_examples /hiv1_protease_series_1/

it contains a readme file: README.hiv1p , which describes the overall process, stepping through the following three directories in turn

hiv1_protease_series_1/setup
hiv1_protease_series_1/run
hiv1_protease_series_1/results

An outline of each step now follows.

1. Setup

The procedure starts with setup, namely structure preparation, typing, charge assignment of the protein target molecule and ligand inhibitors, and assignment of mobile and fixed protein atoms.

1.1. Protein setup

The basis for this setup is the crystal structure of HIV-1 protease and the co-crystalized inhibitor AD-81. The PDB access code for this structure is 2I0D. The multiple aspects to consider when preparing a protein for molecular mechanics calculations starting from PDB coordinates are described in Section V 3.1. of this manual. Furthermore, the AMBER reference manual, available through the link given above, provides detailed advice for the use of AmberTools in this process - see the section titled “Preparing PDB Files”.

The files used for the following steps are found in the following subdirectory:

hiv1_protease_series_1/ setup/protein

1.1.1. Remove all hetatoms and water atoms except atom 1580

For this particular receptor and set of inhibitors, it is important to explicitly include one of the water molecules (atom number 1580) present in the 2I0D crystal structure. Therefore, edit the pdb file 2i0d.pdb deleting everything prior to the first ATOM entry, all HETATOM entries except for that of atom 1580, and everything except the END record after HETATOM 1580. Name the resulting file 2i0d_1580.pdb.

1.1.2. Extract the co-crystalized ligand

The co-crystalized ligand in 2I0D is used as a reference structure, so copy and edit the original 2i0d.pdb file, deleting all atoms except the AD-81 ligand atoms, and rename the file ad_81_from_2i0d.pdb .

1.1.3. Prepare the PDB file for tleap

Prepare the pdb file for tleap by running the script run_pdb4amber_1.sh, i.e.

```
./run_pdb4amber_1.sh >& run_pdb4amber_1.log &
```

This will produce the file 2i0d_1580_p4a.pdb as well as other files required by tleap.

1.1.4. Run tleap to assign parameters

Run tleap to assign parameters using the script run_tleap_2.sh.

```
./run_tleap_2.sh >& run_tleap_2.log &
```

This will produce .incpcrd, .prmtop, .mol2, and .pdb files. These will be named 2i0d_1580_p4a_tleap.*

1.1.5. Convert .prmtop and .inpcrd to .crd, .top, and .mol files

Run the VeraChem amber pathway conversion tool prm2top.pyc using the script run_prm2top_3.sh, i.e.

```
./run_prm2top_3.sh >& run_prm2top_3.log &
```

This will produce the files 2i0d_1580_p4a_tleap_vm2.[crd,top,mol] These are the files that will be used to run the VM2 calculations.

Compare your results with those provided in the ./reference subdirectory to ensure that the procedure was successful.

1.2. Ligand Setup

Some remaining protein setup steps require that the AD-81 ligand be already setup, so next, the full set of ligands are prepared and parameterized. The relevant subdirectories are:

```
hiv1_protease_series_1/setup/ligands/source_files
hiv1_protease_series_1/setup/ligands/vconf
hiv1_protease_series_1/setup/ligands/prepare_ligands
```

1.2.1. Initial 2D structures

Processing with AmberTools requires an input sdf file containing the ligands in 3D, with all hydrogens present and stereochemistry properly defined with parity values. For this example, the ligands were first drawn in 2D by a chemical draw program referencing

figures from the published experimental binding affinity article.(*1*) A 2D mol file was saved for each ligand.

These 2D structures can be found in the ./source_files subdirectory of ligands/. A simple python script (mol_2_sdf.py) is used to assemble them into a single sdf file called umass_1.sdf.

```
python mol_2_sdf.py -o umass_1.sdf
```

To process only a chosen subset of the prepared 2D structures a key file can be used that contains the names of the ligands, one on each line, to be processed e.g.

```
python mol_2_sdf.py -o umass_1.sdf -k ligand_key_5.txt
```

1.2.2. 2D to 3D conversion

VeraChem's Vconf program is used to convert these 2D structures to 3D. The relevant files are found in the vconf/ subdirectory. First, copy over the umass_1.sdf file generated by the last step, and then execute the run_vconf.sh script to carry out the conversion:

```
./run_vconf.sh &
```

The resulting 3D structures can be found in the file

```
hiv1_protease_series_1/setup/ligands/vconf/umass_1_vconf.sdf
```

You can compare your results against those provided in the reference/ subdirectory.

1.2.3. Generate partial charges and assign parameters to the ligands

Ambertools is used to assign bond, angle, torsion, and non-bonded Lennard-Jones parameters, while atom partial charges can be generated either by VeraChem's VCharge method or by AM1-BCC through AmberTools. The resulting prmtop and inpcrd files are then converted to the [crd,top,mol] file set used by VM2.

The prepareLigands.pyc script automates this process. First, go to the prepare_ligands directory

```
hiv1_protease_series_1/setup/ligands/prepare_ligands
```

then copy over the 3D sdf file

```
cp ../source_files/umass_1.sdf .
```

Then, to execute the script choosing VCharge partial atomic charges type:

```
./run_prepareLigands_vcharge.sh &
```

and to assign charge using AM1-BCC type:

./run_prepareLigands_am1-bcc.sh &

While VCharge takes less than a minute for the set of 28 ligands, generation of AM1-BCC partial charges requires a QM calculation, which can take a considerable amount of time, e.g., approximately 3 hours on a Xeon E5-2667, 3.2GHz cpu.

You can compare your results against those in the reference subdirectories.

1.3. Define fixed and mobile protein atoms

The choice of the included mobile and fixed protein atoms can have a significant impact on the final binding energy predictions produced by the VM2 method. VeraChem recommends inclusion of enough mobile atoms to capture relevant aspects such as loop movement on binding, while avoiding inclusion of large numbers of atoms as mobile, which are effectively spectators, so as to keep calculations manageable with respect to turnover times, and also minimize the occurrence of spurious minima that sometimes occur due to force field inadequacies.

A process for defining mobile and fixed atoms for subsequent free energy calculations is now described.

1.3.1. Generate co-crystalized ligand based AD-81 conformation

First, go to the directory

setup/define_fixed_and_mobile_atoms/1_gen_coxtal_ligand_conf

Next, generate a conformation of the co-crystalized ligand AD-81 to use as the reference coordinates to carve out the mobile and fixed atoms in subsequent steps. This is achieved by 'snapping' scaffold atoms from the AD-81 structure generated in Step 2 above, to the corresponding positions of the co-Xtal AD-81 scaffold atoms in the 2I0D PDB file i.e. scaffold atoms in the file ad_81_from_2i0d.pdb generated in Step 1.2.2

The required files are:

ad_81_pdbsnap_confs.inp	: VM2 input file
ad_81.crd	: coordinate file generated in **Section 1.2.3.**
ad_81.top	: topology/parameter file fin **Section 1.2.3.**
ad_81.mol	: mol file generated in **Section 1.2.3.**
ad_81_from_2i0d.pdb	: reference ad_81 coordinates from **Section 1.1.2.**

Generate the AD-81 conformations by typing:

./runvm2.bsh >& runvm2.log

The output of interest is the file:

ad_81.confsearch_rank1.crd

which contains the coordinates of lowest energy AD-81 conformer 'snapped' to the co-crystalized ligand scaffold atoms. The coordinate file is used in the next step.

1.3.2. Relax all hydrogen atoms in the system

To relieve close contacts that can occur on hydrogen atom placement, all hydrogen atom positions in the protein and AD-81 ligand are optimized according to the force field energy function.

Go to the directory

setup/define_fixed_and_mobile_atoms/2_opt_all_protein_h

then copy the file required from last step and rename it:

```
cp ../1_gen_coxtal_ligand_conf/ad_81.confsearch_rank1.crd ad_81_snap2pdb.crd
```

The required files for this step are:

```
2i0d_1580_p4a_tleap_hopt.inp      : VM2 package input file for H atom optimization
ad_81_from_2i0d.pdb               : reference ad_81 coordinates from Section 1.1.2.

2i0d_1580_p4a_tleap_vm2.crd          | Protein coordinates, parameters etc.
2i0d_1580_p4a_tleap_vm2.top       <--| generated by Section 1.1. above.
2i0d_1580_p4a_tleap_vm2.mol          | Copied directly from ./protein

ad_81_snap2pdb.crd                | ad_81_snap2pdb.crd is the just generated
ad_81.top                      <---| ad_81.confsearch_rank1.crd copied and
ad_81.mol                         | renamed. The top and mol files are as in 1.3.1.
```

Relax all hydrogen atom positions by typing:

./runvm2.bsh >& runvm2.log

The outputs of interest are the files

2i0d_1580_p4a_tleap_vm2.geomopt_rank1.crd
ad_81_snap2pdb.geomopt_rank1.crd

which contain the lowest energy coordinates of the protein and ligand AD-81 after hydrogen atom optimization. These coordinates are used in the next step.

1.3.3. Distance based generation of real/live set

Carve out a mobile and fixed set of protein atoms. VM2 uses so-called real and live

sets, where the 'real' set are all the atoms included in the calculation (mobile and fixed) and the 'live' set is the subset of the 'real' set that is mobile. In this step, the VM2 package is used to carve out a 'real' set that comprises all residues that have an atom within 7 Angstoms any atom of the supplied AD-81 ligand coordinates, and a 'live' set of all protein atoms within 5 Angstoms of any atom of the supplied AD-81 ligand coordinates.

Go to the directory

setup/define_fixed_and_mobile_atoms/ 3_dist_based_real_live_set

then copy and rename the required files from the last step:

cp ../2_opt_all_protein_h/2i0d_1580_p4a_tleap_vm2.geomopt_rank1.crd 2i0d_1580_p4a_tleap_vm2_opth.crd

cp ../2_opt_all_protein_h/ad_81_snap2pdb.geomopt_rank1.crd ad_81_snap2pdb_opth.crd

The required files for this step are:

2i0d_1580_p4a_tleap_genlivereal.inp <--- VM2 package input file for generation of 'real' atom set of all atoms within 7 Angstroms of any atom in the supplied AD-81 ligand crd, and a 'live' atom set within 5 Angstroms.

2i0d_1580_p4a_tleap_vm2_opth.crd | The crd file is the just generated
2i0d_1580_p4a_tleap_vm2.top <--| 2i0d_1580_p4a_tleap_vm2.geomopt_rank1.crd
2i0d_1580_p4a_tleap_vm2.mol | renamed. The top and mol are unchanged.

ad_81_snap2pdb_opth.crd | ad_81_snap2pdb_opth.crd is the just generated
ad_81.top <---| ad_81_snap2pdb.geomopt_rank1.crd from above
ad_81.mol | renamed. The top and mol files are unchanged.

Generate the real and live sets by typing:

./runvm2.bsh >& runvm2.log

The following output files allow you to visualize the ‘live’ set produced:

2i0d_1580_p4a_tleap_genlivereal.mol2 <--Load into visualizer to see live set produced.
2i0d_1580_p4a_tleap_genlivereal.pdb
2i0d_1580_p4a_tleap_genlivereal.sdf

To see the 'real' set of atoms defined in by these distance cutoffs, run the same calculation with the input file 2i0d_1580_p4a_tleap_genlivereal.inp changed to output 'real' atoms:

```
#
atomsToOutput
real
#
```

Generated output files required for running VM2:

2i0d_1580_p4a_tleap_vm2_opth_liverealatoms.txt	<---	This file contains the atom numbers of the live and real atoms generated by the applied distance cutoffs.

Once you are happy with the defined real/live sets copy the protein data files required for VM2 runs directly into the directory define_fixed_and_mobile_atoms/ i.e.

```
cp 2i0d_1580_p4a_tleap_vm2.mol ../.
cp 2i0d_1580_p4a_tleap_vm2_opth.crd ../.
cp 2i0d_1580_p4a_tleap_vm2.top ../.
cp 2i0d_1580_p4a_tleap_vm2_opth_liverealatoms.txt ../2i0d_5_7_live_real.txt
```

NOTE: mandatory renaming of 2i0d_1580_p4a_tleap_vm2_opth_liverealatoms.txt to include the text "live_real"

The setup stage is now complete.

2. Run Calculations

The next step is to run the protein-ligand, protein, and ligand, free energy calculations. The relevant directories and readme file are:

```
hiv1_protease_series_1/run/1_ligand_confgen
hiv1_protease_series_1/run/2_vm2_runs
hiv1_protease_series_1/run/README.runvm2
```

Optionally, ligand conformations can be pre-generated in /1_ligand_confgen and used to seed the VM2 calculations in /2_vm2_runs.

2.1. Generation of Ligand Starting Conformations

Two types of pre-generated ligand conformations can be utilized in this example. One is 'snapped' conformations, where atoms in each ligand common to a, for example, co-crystalized ligand are, with an applied guiding force, superimposed, while conformational space of the remaining atoms is sampled. The other is randomly orientated conformations of the ligand, suitable for when no pose information is known, only the location of the binding site.

2.1.1. Example run

Go to the directory

run/1_ligand_confgen

This directory contains a python script to generate run directories for conformer generation, and a python script to run the conformer generation calculations. Example usage is as follows:

python build_ligand_start_conf_dirs.py -t ad_81_from_2i0d.pdb

will first populate the directories

1_ligand_confgen/gen_ligand_start_confs_snap

1_ligand_confgen/gen_ligand_start_confs_rndm

with the required subdirectories, input files, and data files to run. Then the following command

python run_ligand_confs_gen.py -r slurm

will step through all these subdirectories, generating slurm scripts, and submitting the calculations to the batch queue.

Note: Requirements for this example run are:

ad_81_from_2i0d.pdb	<---	must be present in /setup/ligands/prepareLigands
scaffold_mapping_wkey.txt	<---	must be present in the current directory and contain the mapping of each ligand onto the reference ligand

2.1.2. Options available for building conformer generation directories

The python script build_ligand_start_conf_dirs.py can take a number of arguments for non-default control the source of the system data etc.:

-d or --data	reference	: Populate 'input_data' directory using the data in the setup 'reference' directories e.g. /setup/ligands/prepareLigands/reference, and subsequently build the run directories with this data.
	new	: Populate 'input_data' directory using the new data in the setup directories e.g. /setup/ligands/prepareLigands, and subsequently build the run directories with this data. (Default behavior.)
	reuse	: Reuse the data from an already populated

'input_data' directory.

-s or --startconfs random : Make a run directory for each ligand in the series for generation of ligand conformers in random orientations and with their center of geometry (COG) placed at a template ligand's COG.

snap : Make a run directory for each ligand in the series for generation of ligand conformers where scaffold atoms are 'snapped' to corresponding template ligand scaffold atoms (via applied harmonic potentials).

all : Make both of the above run directories. (Default behavior.)

-t or --template 'template_filename' : Name of file containing template ligand coordinates e.g. co-xtal ligand or previously docked ligand. Required unless '-d reuse' option set.

-c or --clear input : Delete the contents of 'input_data' directory.

rundirs : Delete the contents of the run directories 'gen_ligand_start_confs_rndm' and 'gen_ligand_start_confs_snap'.

all : Delete content from the 'input_data' directory and the run directories.

Example usage:

```
python build_ligand_start_conf_dirs.py -c rundirs -d reuse
```

This will clear the contents of previously generated run directories and use the data already present in ./input_data to regenerate the run directories i.e. data will not be taken from the setup directories in this case.

2.1.3. Options available for running conformer generation

The python script run_ligand_confs_gen.py can take a number of arguments:

-s or --startconfs random : Step through each ligand directory in /gen_ligand_start_confs_rndm and

		submit a calculation for generation of ligand conformers in random orientations and with their center of geometry (COG) placed at a template ligand's COG.
	snap	: Step through each ligand directory in gen_ligand_start_confs_snap and submit a calculation for generation of ligand conformers where scaffold atoms are 'snapped' to corresponding template ligand scaffold atoms (via applied harmonic potentials).
	all	: Carry out both sets of calculations. (Default behavior.)
-r or --runscript	bsh	: Generate and use bash shell scripts for submission of each calculation. (Default behavior.)
	csh	: Generate and use c-shell scripts for submission of each calculation.
	pbs	: Generate a pbs script for submission of each calculation to a queue.
	slurm	: Generate a slurm script for submission of each calculation to a queue.
-q or --partition	'queue name'	: For pbs and slurm run scripts, the name of the queue or partition if the default queue is not being used.
-p or --prepmode		: If present the run scripts are generated and placed in every directory, but the calculations are not submitted.

2.2. Protein-ligand calculations

Two main types of VM2 protein-ligand free energy calculation are available. One is regular VM2, which carries out iterative rounds of conformational searching until convergence; the other type carries out geometry optimizations of protein-ligand conformations constructed from ligand conformers read-in and processes them for free energy. The latter is much faster, but much less exhaustive in terms of sampling conformational space. In combination, there are three ways to seed these two VM2 calculation types with ligand conformers: multiple conformers with selected atoms ‘snapped’ to a reference ligand – see Section 2.1. above; multiple conformers randomly

orientated in space, but placed at the location of the binding site – see Section 2.1. above, and a single conformer, based on the position and geometry in which it was prepared originally. This provides for six different overall VM2 calculation schemes, which cover various types of use scenarios.

2.2.1. Example run

Go to the directory

run/2_vm2_runs

This directory contains a python script to generate run directories for protein-ligand VM2 free energy calculations, and a python script to step through the directories and run the calculations. Example usage is as follows:

python build_vm2_run_dirs.py -t ad_81_from_2i0d.pdb

will first populate the following six directories, which cover the calculation types described above, with the required subdirectories, input files, and data files to run.

/2_vm2_runs/fast_vm2_snap
/2_vm2_runs/fast_vm2_rndm
/2_vm2_runs/fast_vm2_single
/2_vm2_runs/vm2_snap
/2_vm2_runs/vm2_rndm
/2_vm2_runs/vm2_single

Note: For "_snap" and "_rndm" types, the corresponding pre-generation of ligand conformers – Section 2.1. - must already have occurred.

Then the following command:

python run_vm2_calculations.py -s snap -v fast -r slurm

will step through the subdirectories of /2_vm2_runs/fast_vm2_snap, generating slurm scripts, and submitting the calculations to the batch queue. Similarly, any of the other five calculations types may be run by setting the appropriate flags – see Section 2.2.2 below.

2.2.2. Options available for building VM2 directories

The python script build_vm2_run_dirs.py can take a number of arguments for non-default control of the source of the system data etc.:

```
-d or --data    reference    : Populate 'input_data' directory using the
                               data in the setup 'reference' directories
                               e.g. /setup/ligands/prepareLigands/reference and
                               /setup/define_fixed_and_mobile_atoms/reference,
                               and the ligand start conformer generation
                               reference directory /run/1_ligand_confgen/reference
```

and subsequently build the run directories
with this data.

new : Populate 'input_data' directory using the new data in the setup directories e.g. /setup/ligands/prepareLigands and /setup/define_fixed_and_mobile_atoms/ and the ligand start conformer generation directories /run/1_ligand_confgen/gen_ligand_start_confs_rndm and /run/1_ligand_confgen/gen_ligand_start_confs_snap and subsequently build the run directories with this data. (Default behavior.)

reuse : Reuse the data from an already populated 'input_data' directory.

-s or --startconfs random : Requests run directory set up for VM2 free energy calculations where randomly oriented ligand conformers are placed in the active site and are used to generate starting protein-ligand conformations.

snap : Requests run directory set up for VM2 free energy calculations where ligand conformers in which scaffold atoms have been 'snapped' to corresponding scaffold atoms of a template ligand (e.g. co-xtal ligand) are used to generate starting protein-ligand conformations.

single : Requests run directory set up for VM2 free energy calculations where a single ligand starting conformation and placement is used based on the supplied ligand .crd file coordinates. The placement can be adjusted if a template ligand is supplied and the place ligand flag set; see -t, --template and -p, --placelig below. Only used a non-adjusted ligand .crd if you prepared the ligand in a very good placement and pose in the receptor binding site.

all : Requests both types of directory to be set up. (Default behavior.)

-t or --template 'template_filename' : Name of file containing template ligand coordinates e.g. co-xtal ligand or previously docked ligand. Could simply be coordinates that signifiy the loacation of the binding site. Not required unless random start conformers are in use or the place ligand option just below is set.

-p or --placelig tcog : Place ligand .crd coordinates center of geometry at template ligand's center of geometry.

-c or --clear input : Delete the contents of 'input_data' directory.

rundirs : Delete the contents of the run directories.

all : Delete content from the 'input_data' directory and the run directories.

-v or --vm2type regular : Requests run directory set up for regular VM2 protein-ligand free energy calculations, which carry out extensive conformational searching.

fast : Requests run directory set up for fast VM2 protein-ligand free energy calculations, which calculate free energies via geometry optimizing protein-ligand conformations generated from read-in ligand conformers previously snapped to a template scaffold.

all : Requests set up for both types of VM2 calculation.

-k or --keyfile 'ligand_key_filename' : Name of text file containing the subset of ligands in the series - one on each line (see ligand_key_5.txt.)

2.2.3. Options available for running VM2 calculations

The python script run_ligand_confs_gen.py can take a number of arguments:

-s or --startconfs random : Requests that VM2 free energy calculations are run for the series where randomly oriented ligand conformers are placed in the active site and are used to generate starting protein-ligand conformations.

snap : Requests that VM2 free energy calculations are run for the series where ligand conformers in which scaffold atoms have been 'snapped' to corresponding scaffold atoms of a template ligand (e.g. co-xtal ligand) are used to generate starting protein-ligand conformations. (Default behavior.)

all : Requests both types of run be carried out.

-r or --runscript bsh : Generate and use bash shell scripts for submission of each calculation. (Default behavior.)

csh : Generate and use c-shell scripts for submission of each calculation.

pbs : Generate a pbs script for submission of each calculation to a queue.

slurm : Generate a slurm script for submission of each calculation to a queue.

-q or --partition 'queue name' : For pbs and slurm run scripts, the name of the queue or partition if the default queue is not being used.

-p or --prepmode : If present the run scripts are generated and placed in every directory, but the calculations are not submitted.

-v or --vm2type regular : Requests regular VM2 protein-ligand free energy calculations for the series, which carry out extensive conformational searching.

fast : Requests fast VM2 VM2 protein-ligand free energy calculations for the series, which calculate free energies via geometry optimizing protein-ligand conformations generated from read-in ligand conformers snapped to a template scaffold. (Default behavior.)

all : Requests both types of VM2 calculation are run for the series.

-i or --mpiprocs n (integer) : Sets the number of MPI processes to run. Currently all processes must run on the same node - though hand editing of run scripts can remove this restriction. The default is 8.

-g or --gpu : If present requests use of CUDA enabled VM2 executable.

-o or --ompthreads 1 : If -g not set results in MPI parallelism only. Enforced for ligand only runs.

2 : If set will result in MPI+OpenMP run (8 MPI processes (default), 2 OpenMP threads per process). If -g also set will result in MPI+OpenMP+CUDA parallelism.

4 : Same as previous, but 4 OpenMP threads.

```
-m or --molsystems    complexes+ligands      |
                                             |
                      complexes+protein      |
                                             |
                        protein+ligand       |
                                             |
                          complexes          |----> Run subset of the moleculer system
                                             |      types.
                                             |
                            ligands          |
                                             |
                            protein          |

                              all            : Default. Run ligands, complexes, and
                                               protein.
```

Example usage:

```
nohup python run_vm2_calculations.py -g -o 2
```

Run default fast-snap set of calculations (fast_vm2_snap directory) with 8 MPI process calculations for ligand calculations, but MPI+OpenMP+CUDA calculations for the complexes and the protein.

This run utilizes 8 MPI processes with 1 GPU per MPI process and 2 OpenMP threads per MPI process. It therefore requires 16 compute cores and 8 GPUs.

3. Results Collection

When the protein-ligand, protein, and ligand VM2 free energy calculations for the complete ligand series have completed, the binding free energies may then be calculated, and the formatted files, e.g., .mol2, .pdb, .sdf, containing the associated molecular structures collected.

The relevant directories and readme file are:

```
hiv1_protease_series_1/results
hiv1_protease_series_1/results/conformers
hiv1_protease_series_1/results/README.results
```

3.1. Generate binding free energy spreadsheets and collect conformer files

Go to the directory

```
hiv1_protease_series_1/results
```

To generate spreadsheets and collect molecule conformer files for the "fast_vm2_snap" calculations from Section 2.2.1 type:

```
python create_vm2_summaries.py -c fast_vm2_snap -n 2i0d -l ad_81
```

Requirements:

File containing experimental data: experimental_data.csv

The filename must contain "experimental_data".
The format is <proteinname_ligandname>, <value> e.g.

```
2i0d_ad_12,-9.367
2i0d_ad_17,-14.203
2i0d_ad_23,-11.559
2i0d_ad_24,-10.126
2i0d_ad_32,-10.337
2i0d_ad_33,-12.458
:
```

Output spreadsheets:

```
results/2i0d_fast_vm2_snap_complex.csv
results/2i0d_fast_vm2_snap_protein.csv
results/fast_vm2_snap_ligand.csv
results/2i0d_fast_vm2_snap_SUMMARY.csv
```

The last of these contains the binding free energies.

Output conformer files:

For the protein, each ligand, and each protein-ligand complex, formatted files (e.g. mol2, pdb, sdf, xyz) containing the lowest energy conformer, and the eight lowest energy conformers are written to:

```
results/conformers/fast_vm2_rndm/complexes
results/conformers/fast_vm2_rndm/ligands
results/conformers/fast_vm2_rndm/protein
```

3.2. Results generation options

For the script create_vm2_summaries.py the following two commandline arguments are mandatory with the following options:

-c or --calctype	fast_vm2_snap	: Identify the calculation type to collect and summarize run data for.
	fast_vm2_rndm	
	fast_vm2_single	

vm2_snap

vm2_rndm

vm2_single

-n or --receptorname : Provide the name of the receptor
e.g. for this case the protein
is named “2i0d”

There are two additional non mandatory arguments:

-d or --data new : Sets the source of the calculation
data to be extracted and summarized
as ../run/2_vm2_runs/fast_vm2_snap etc.
(Default behavior.)

reference : Sets the source of the calculation
data to be extracted and summarized
as ../run/2_vm2_runs/reference/fast_vm2_snap etc.

-l or --refligand : Provide the name of the reference
ligand to be used in relative binding
affinity calculation i.e. for Delta(DeltaG)
The default is no reference.

XI. Host-guest example

The following is an example of the Discovery Studio Visualizer route for a host molecule and xx guests (ligands).

XII. VeraChem file formats

1. VeraChem’s topology/parameter file (.top) format examples

The .top file format specification is described in detail in Section II. The following is a specific example for a small (ligand) molecule and the CHARMM force field – note the columns 8 and 9 in the atom block, which, specific to CHARMM, contain van der Waals parameters for 1-4 interactions.

```
        !NTITLE 1
!NATOM: 23
   1 C6R    12.01100   -0.11100   -0.05000    2.04000   -0.10000    1.76000
   2 C6R    12.01100   -0.11100   -0.05000    2.04000   -0.10000    1.76000
   3 C6R    12.01100   -0.11300   -0.05000    2.04000   -0.10000    1.76000
   4 C6R    12.01100   -0.11300   -0.05000    2.04000   -0.10000    1.76000
   5 C6R    12.01100   -0.01000   -0.05000    2.04000   -0.10000    1.76000
   6 C6R    12.01100    0.08800   -0.05000    2.04000   -0.10000    1.76000
   7 C      12.01100    0.59800   -0.14100    1.87000
   8 CT     12.01100   -0.25800   -0.09030    1.80000   -0.10000    1.75000
   9 CT     12.01100    0.03700   -0.09030    1.80000   -0.10000    1.75000
  10 NP     14.00670   -0.69000   -0.09000    1.83000   -0.10000    1.63000
  11 O      15.99940   -0.51600   -0.15910    1.55000   -0.20000    1.36000
  12 OS     15.99940   -0.35100   -0.15910    1.60000   -0.20000    1.36000
  13 HA      1.00800    0.10900   -0.04200    1.33000
  14 HA      1.00800    0.10900   -0.04200    1.33000
  15 HA      1.00800    0.10900   -0.04200    1.33000
  16 HA      1.00800    0.10900   -0.04200    1.33000
  17 HA      1.00800    0.09100   -0.04200    1.33000
  18 HA      1.00800    0.09100   -0.04200    1.33000
  19 HA      1.00800    0.09100   -0.04200    1.33000
  20 HA      1.00800    0.08700   -0.04200    1.33000
  21 HA      1.00800    0.08700   -0.04200    1.33000
  22 H       1.00800    0.33400   -0.04980    0.80000
  23 H       1.00800    0.33400   -0.04980    0.80000
!NBOND: 23
   1     3   880.000   1.38300 C6R  C6R
   1     5   880.000   1.38300 C6R  C6R
   1    13   740.000   1.08000 C6R  HA
   2     4   880.000   1.38300 C6R  C6R
   2     5   880.000   1.38300 C6R  C6R
   2    14   740.000   1.08000 C6R  HA
   3     6   880.000   1.38300 C6R  C6R
   3    15   740.000   1.08000 C6R  HA
   4     6   880.000   1.38300 C6R  C6R
   4    16   740.000   1.08000 C6R  HA
   5     7   772.000   1.46000 C6R  C
   6    10   780.000   1.35500 C6R  NP
   7    11  1280.000   1.22500 C    O
   7    12   700.000   1.31900 C    OS
   8     9   536.000   1.52900 CT   CT
   8    17   680.000   1.09000 CT   HA
   8    18   680.000   1.09000 CT   HA
   8    19   680.000   1.09000 CT   HA
   9    12   786.000   1.42000 CT   OS
   9    20   680.000   1.09000 CT   HA
   9    21   680.000   1.09000 CT   HA
  10    22   931.200   1.00000 NP   H
  10    23   931.200   1.00000 NP   H
!NTHETA: 37
   3     1     5   140.000  2.094395 C6R  C6R  C6R
   3     1    13    62.000  2.094395 C6R  C6R  HA
   5     1    13    62.000  2.094395 C6R  C6R  HA
   4     2     5   140.000  2.094395 C6R  C6R  C6R
   4     2    14    62.000  2.094395 C6R  C6R  HA
   5     2    14    62.000  2.094395 C6R  C6R  HA
   1     3     6   140.000  2.094395 C6R  C6R  C6R
   1     3    15    62.000  2.094395 C6R  C6R  HA
   6     3    15    62.000  2.094395 C6R  C6R  HA
   2     4     6   140.000  2.094395 C6R  C6R  C6R
   2     4    16    62.000  2.094395 C6R  C6R  HA
   6     4    16    62.000  2.094395 C6R  C6R  HA
   1     5     2   140.000  2.094395 C6R  C6R  C6R
   1     5     7   140.000  2.094395 C6R  C6R  C
   2     5     7   140.000  2.094395 C6R  C6R  C
   3     6     4   140.000  2.094395 C6R  C6R  C6R
```

```
     3     6    10   130.000  2.094395 C6R  C6R  NP
     4     6    10   130.000  2.094395 C6R  C6R  NP
     5     7    11   172.000  2.216568 C6R  C    O
     5     7    12   120.000  1.919862 C6R  C    OS
    11     7    12   162.000  2.171190 O    C    OS
     9     8    17    75.000  1.932079 CT   CT   HA
     9     8    18    75.000  1.932079 CT   CT   HA
     9     8    19    75.000  1.932079 CT   CT   HA
    17     8    18    66.000  1.881465 HA   CT   HA
    17     8    19    66.000  1.881465 HA   CT   HA
    18     8    19    66.000  1.881465 HA   CT   HA
     8     9    12   160.000  1.910612 CT   CT   OS
     8     9    20    75.000  1.932079 CT   CT   HA
     8     9    21    75.000  1.932079 CT   CT   HA
    12     9    20   118.000  1.889319 OS   CT   HA
    12     9    21   118.000  1.889319 OS   CT   HA
    20     9    21    66.000  1.881465 HA   CT   HA
     6    10    22    60.000  2.094395 C6R  NP   H
     6    10    23    60.000  2.094395 C6R  NP   H
    22    10    23    36.000  2.052507 H    NP   H
     7    12     9   166.000  2.022837 C    OS   CT
!NPHI: 46
     5     1     3     6     2.800   2.000   3.142 C6R  C6R  C6R  C6R
     5     1     3    15     3.000   2.000   3.142 C6R  C6R  C6R  HA
    13     1     3     6     3.000   2.000   3.142 C6R  C6R  C6R  HA
    13     1     3    15     2.500   2.000   3.142 HA   C6R  C6R  HA
     3     1     5     2     2.800   2.000   3.142 C6R  C6R  C6R  C6R
     3     1     5     7     3.100   2.000   3.142 X    C6R  C6R  X
    13     1     5     2     3.000   2.000   3.142 C6R  C6R  C6R  HA
    13     1     5     7     3.100   2.000   3.142 X    C6R  C6R  X
     5     2     4     6     2.800   2.000   3.142 C6R  C6R  C6R  C6R
     5     2     4    16     3.000   2.000   3.142 C6R  C6R  C6R  HA
    14     2     4     6     3.000   2.000   3.142 C6R  C6R  C6R  HA
    14     2     4    16     2.500   2.000   3.142 HA   C6R  C6R  HA
     4     2     5     1     2.800   2.000   3.142 C6R  C6R  C6R  C6R
     4     2     5     7     3.100   2.000   3.142 X    C6R  C6R  X
    14     2     5     1     3.000   2.000   3.142 C6R  C6R  C6R  HA
    14     2     5     7     3.100   2.000   3.142 X    C6R  C6R  X
     1     3     6     4     2.800   2.000   3.142 C6R  C6R  C6R  C6R
     1     3     6    10     3.100   2.000   3.142 X    C6R  C6R  X
    15     3     6     4     3.000   2.000   3.142 C6R  C6R  C6R  HA
    15     3     6    10     3.100   2.000   3.142 X    C6R  C6R  X
     2     4     6     3     2.800   2.000   3.142 C6R  C6R  C6R  C6R
     2     4     6    10     3.100   2.000   3.142 X    C6R  C6R  X
    16     4     6     3     3.000   2.000   3.142 C6R  C6R  C6R  HA
    16     4     6    10     3.100   2.000   3.142 X    C6R  C6R  X
     1     5     7    11     1.300   2.000   3.142 O    C    C6R  C6R
     1     5     7    12     0.500   2.000   3.142 X    C    C6R  X
     2     5     7    11     1.300   2.000   3.142 O    C    C6R  C6R
     2     5     7    12     0.500   2.000   3.142 X    C    C6R  X
     3     6    10    22     0.500   2.000   3.142 X    C6R  NP   X
     3     6    10    23     0.500   2.000   3.142 X    C6R  NP   X
     4     6    10    22     0.500   2.000   3.142 X    C6R  NP   X
     4     6    10    23     0.500   2.000   3.142 X    C6R  NP   X
     5     7    12     9     2.500   2.000   3.142 X    C    OS   X
    11     7    12     9     2.500   2.000   3.142 X    C    OS   X
    17     8     9    12     0.150   3.000   0.000 X    CT   CT   X
    17     8     9    20     0.150   3.000   0.000 X    CT   CT   X
    17     8     9    21     0.150   3.000   0.000 X    CT   CT   X
    18     8     9    12     0.150   3.000   0.000 X    CT   CT   X
    18     8     9    20     0.150   3.000   0.000 X    CT   CT   X
    18     8     9    21     0.150   3.000   0.000 X    CT   CT   X
    19     8     9    12     0.150   3.000   0.000 X    CT   CT   X
    19     8     9    20     0.150   3.000   0.000 X    CT   CT   X
    19     8     9    21     0.150   3.000   0.000 X    CT   CT   X
     8     9    12     7     0.100   3.000   0.000 CT   CT   OS   C
```

```
  20     9    12     7     0.330   3.000   3.142 X    CT   OS   X
  21     9    12     7     0.330   3.000   3.142 X    CT   OS   X
!NIMPHI: 8
   1     5    13     3   150.000   0.000   3.142 HA   X    X    C6R
   2     4    14     5   150.000   0.000   3.142 HA   X    X    C6R
   3     6    15     1   150.000   0.000   3.142 HA   X    X    C6R
   4     2    16     6   150.000   0.000   3.142 HA   X    X    C6R
   5     1     7     2   200.000   0.000   3.142 C    X    X    C6R
   6     3    10     4   180.000   0.000   3.142 C6R  X    X    NP
   7     5    12    11   294.000   0.000   3.142 C    X    X    O
  10    22    23     6   180.000   0.000   3.142 C6R  X    X    NP
!NBFIX: 0
!NFINAL: 6
        23         23         37         46          8 9999
!NDON:
```

1. Definition of protein real/live atom sets

The following provides the format for the file to identify the atoms to include in the calculation (real atoms), and which of these are mobile (live atoms). See Section VIII 2.

```
#total no. of atoms in the protein
numberOfAtoms
3137
#no of flexible atoms
numberOfLiveAtoms
645
#list of flexible atoms
listOfLiveAtoms
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
158
378
379
380
382
:
:
2943
2944
2955
3135
3136
3137
```

```
#number of real atoms
numberOfRealAtoms
2027
#list of real atoms
listOfRealAtoms
67
68
69
70
71
72
73
74
75
76
:
:
3096
3097
3098
3099
3135
3136
3137
#end
end
```

2. Identify constrained (tethered) atoms sets

The following provides the format for the file to identify sets of atoms to which constraints are applied. In case shown to sets of atoms are identified, one is a large set of 503 atoms, the other a small set of 4 atoms. The type and strength of the constraints applied are defined in the .inp file (see Section VIII 14.)

```
#Constrained atoms information
#numProtein
1
#numLigand
1

#proteinid
1

#setid
1
#numTetheredAtoms
503
#atomList
122
123
124
125
126
127
128
129
```

```
130
131
132
133
134
135
267
268
269
270
271
:
:
2654
2655
2656

#setid
2
#numTetheredAtoms
4
#atomList
2686
2689
2692
2695

#ligandid
1
#numTetheredAtoms
0
#atomList

#end
end
```

3. Identify atoms to exclude search drivers

The following provides the format for the file to identify atoms which if present in a particular search driver results in the exclusion of that driver in the conformational search. (see Section VIII 7.)

```
#Excluded Atoms information
#numProtein
1
#numLigand
1

#proteinid
1
#numExcludedAtoms
1066
#atomList
122
123
124
```

```
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
 :
 :
2682
2683
2686
2689
2692
2695

#ligandid
1
#numExcludedAtoms
0
#atomList


#end
end
```

XIII. Parallel processor performance

XIII. References

1. A. Ali *et al.*, Discovery of HIV-1 protease inhibitors with picomolar affinities incorporating N-aryl-oxazolidinone-5-carboxamides as novel P2 ligands. *J. Med. Chem.* **49**, 7342-7356 (2006).

XIV. Index