# Getting Started with VM2

## VM2 Version 3.0

VeraChem LLC

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

**Contact:**

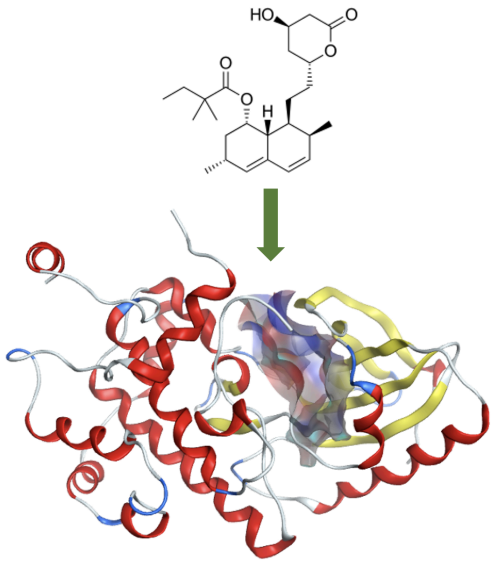
For information regarding VM2 software package licensing contact VeraChem LLC at sales@verachem.com

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

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

# 1 Welcome to VM2

Thank you for selecting VeraChem's VM2 software. VM2 is a comprehensive free energy package capable of generating fast and accurate predictions of binding affinity for protein-ligand and host-guest systems. A basic need in the small molecule drug discovery process is to determine how strongly a particular ligand or series of ligands bind to a protein active site and why.




VM2 has been designed and developed to provide computational and medicinal chemists with predicted binding affinity data across ligand series, for purposes of rank ordering, as well as for individual ligands for feedback on ligand design ideas. Binding free energies are computed directly by VM2 as the difference between the chemical potential of the complex and the sum of the chemical potentials of the ligand and the receptor alone.

$$\Delta G^o_{\text{bind}} = \mu^o_{\text{complex}} - (\mu^o_{\text{rec}} + \mu^o_{\text{lig}}) \tag{1}$$

The chemical potentials of the complex, receptor, and ligand are obtained by separate mining minima calculations, which search for low energy conformers and evaluate the contribution of each corresponding energy well to the system's total configuration integral Z, using a harmonic approximation plus numerical mode scanning to account for anharmonic effects:

$$Z_{\text{i}} = \int_{\text{well i}} e^{-(U(\mathbf{r})+W(\mathbf{r}))/RT} d\mathbf{r} \tag{2}$$

$$\mu^o = -RTln((8\pi^2)/C^o) \sum_{\text{i}} Z_{\text{i}}) \tag{3}$$

where R is the gas constant, T is the temperature, $C^o$ is standard concentration, U is the potential energy and W is the solvation energy. More information on the theoretical basis of VM2 and the software package, as well as results of benchmarking of VM2 against experiment, can be seen in this presentation and on VeraChem's website.

# 2 Requirements

## 2.1 Installed VM2 Package

To install VM2 follow the instructions in the file `VM2_3.0_package_installation.txt` provided with the software package; alternatively, the same instructions can be found in the VM2 "Quick Start: Installation" documentation. Be certain to check that the automated installation tests were successful before proceeding. Note that a common source of failure for the installation/installation tests is unset environment variables, so in the case of failed tests make sure the following are set in your `.bashrc` file:

```
export $VCHOME=/home/yourname/installdir/vcCompChem_3
export VM2HOME=$VCHOME

export VCPYTHON=$VCHOME/exe/vc_python
export VM2PYTHON=$VCPYTHON
```

If you are still unable to sucessfully install VM2, contact VeraChem at support@verachem.com

## 2.2 Other Software

As an option, VM2 can use AmberTools (Version 18+) to prepare and parameterize protein receptors and ligands. This requires that an installation of AmberTools be available and in the user's default path so it can be found at runtime by VM2. We recommend that AmberTools be installed using Conda and the resulting environment be activated prior to running VM2. Instructions for obtaining and installing AmberTools can be found here.

Note: An installation of AmberTools is currently recommended; however, the user has the option to provide VM2 with receptor and ligand forcefield parameter/topology files directly in the Amber prmtop format. In this case access to AmberTools from VM2 is not required.

## 2.3 Computational Resources

The minimum recommended resources are a Red Hat-based linux system (Version 7+) with 8 to 12 CPU cores (Intel or AMD) and approximately 24 GB of RAM (2 GB per CPU core).

To compute the binding affinity for a protein receptor and a series of ligands using the VM2 the total number of free energy (chemical potential) calculations required is equal to the number-of-ligands x 2 + 1. On typical Intel or AMD processors the calculations for the receptor alone and complexes will each take approximately 20-30 minutes and the ligand-only calculations will take approximately 5 minutes each. Because VM2 based binding affinities are generated from a set of entirely independent free energy calculations, access to a computer cluster, where multiple 12-core calculations can be run concurrently will greatly reduce the wall-clock time taken to calculate binding affinities for a receptor and a series of ligands. Use of a resource management/queue system such as SLURM or PBS is highly recommended.

# 3 VM2 Package Overview

The VM2 package is composed of a highly efficient, compiled language, free energy calculation engine that is parallelized on CPUs and GPUs, and a python based workflow capability, with associated molecular manipulation tools, which facilitates application of the calculation engine to receptor and ligand series binding affinity calculations. The VM2 workflow, given raw receptor and ligand series structure files (e.g. PDB for protein, SD files for ligands) automates the required setup (molecule preparation, ligand 2D to 3D, forcefield parameter assignment, partial atomic charge assignment), generates run directories, allows for submission of free energy calculations via a resource manager or on the user's local machine, and, on completion of calculations, collects and organizes receptor-ligand series calculated binding affinities and other energy data for analysis, as well molecular conformers in formatted structure files e.g. PDB, MOL2, SDF, XYZ.

# 4 VM2 Workflow

## 4.1 Example VM2 Workflows

The installed VM2 package includes VM2 workflow examples (or tutorials) for various types of protein-ligand series and host-guest series binding affinity calculations. See the following directories:

```
$VCHOME/tutorials/vm2/workflow/protein_ligand

$VCHOME/tutorials/vm2/workflow/host_guest
```

The example workflows contain detailed README files. It is highly recommended that as a first step the new VM2 user should run one or more of these example workflows.

### Protein-ligand Series Example Workflow

For an example of a protein-ligand series (HIV-1 protease + 5 ligands) workflow where there is a co-crystalized ligand available with the same scaffold as the ligand series, see the directories:

```
$VCHOME/tutorials/vm2/workflow/protein_ligand/vm2pkg_pl_hivp_umass5_workflows/vm2_coxtal_exa
mple

$VCHOME/tutorials/vm2/workflow/protein_ligand/vm2pkg_pl_hivp_umass5_workflows/rawdata_hivp_u
mass5_ad81template
```

The first directory 'vm2_coxtal_example' contains the scripts that invoke the workflow steps; the second directory 'rawdata_hivp_umass5_ad81template' contains the raw input data e.g. protein PDB file, ligand 2D mol files, etc. In the 'vm2_coxtal_example' directory see the text file README.coxtalvm2 for further details on how to run this protein-ligand VM2 workflow. **Note:** If run on a single 12-core node, this protein-ligand series example will take approximately two hours to complete; on two 12-core nodes via SLURM/PBS it will take approximately one hour to complete.

### Host-guest Series Example Workflow

For an example of a host-guest series (cucurbit[7]uril + 29 guests) workflow, where starting host-guest conformations are automatically generated by randomly rotating guest-only conformations and placing them into the host, see the directories:

```
$VCHOME/tutorials/vm2/workflow/host_guest/cb7_gilson_set/cb7_gilson_vm2

$VCHOME/tutorials/vm2/workflow/host_guest/cb7_gilson_set/rawdata_hg_cb7_gilson_set
```

The first directory 'cb7_gilson_vm2' contains the scripts for running the workflow steps, and the second directory 'rawdata_hg_cb7_gilson_set' contains the raw inputa data i.e. host and ligand SD/mol files. In the 'cb7_gilson_vm2' directory see the text file README.hgvm2 for more detail on how to run this VM2 host-guest workflow. **Note:** If run on a single 12-core node, this host-guest series example will take approximately four hours to complete; on tow 12-core nodes via SLURM/PBS it will take approximately two hours to complete.

## 4.2 VM2 Workflow Components

You may wish to use the example workflows provided as *templates* for your own protein-ligand series and/or host-guest series binding free energy calculations. Alternatively, you may want to use the workflow script builder tool provided with the VM2 package to generate workflows from scratch. Regardless, it is important to be aware of the structure of the VM2 workflow and the role of each of its components.

Given the required raw input data, e.g. protein PDB file, ligands SD file etc., the basic form of the workflow is 1) setup of molecular system, including forcefield parameter assignment, 2) pre-generation of ligand conformers, 3) generation of VM2 free energy calculation run directories, 4) submission of all VM2 free energy calculations, possibly through a resource manager, and 5) binding free energy results extraction into comma-separated-value (CSV) files and organization of generated molecular conformers in standard formatted files for visualization etc.

As seen in the example workflows, to carry out the full workflow just described above, the user can invoke the following five scripts, each corresponding to a workflow step:

```
./vc_workflow_1_setup.sh*
./vc_workflow_2_ligconfs.sh*
./vc_workflow_3_genrundirs.sh*
./vc_workflow_4_run.sh*
./vc_workflow_5_extract_results.sh*
```

The user can control aspects of each of these workflow steps by editing the workflow control parameter values in the file:

```
set_vc_workflow_control_vars.sh
```

Further details of required raw input data, workflow control parameters, workflow steps, and the workflow script builder tool now follow.

## Raw Input Data

To use VM2's workflow framework, the first requirement is the placement of formatted structure files for the molecular system in a simple directory structure. Below is an example for a protein-ligand system:

```
rawinputdatadir/protein
rawinputdatadir/ligands
rawinputdatadir/experiment
rawinputdatadir/template_ligand
```

Typically, a PDB file for the protein receptor is placed in the `/protein` directory and a set of MOL files or an SD file are/is placed in `/ligands` . If experimental binding affinities are available to compare with the VM2 predicted values, they can be placed in a simply formatted text file in `/experiment` . Finally, to generate good protein-ligand complex starting structures for the ligand series, a SDF or PDB file of a ligand in a reasonable bound conformation in the protein active site should be supplied in `/template_ligand`. Often the simplest route to obtaining a template ligand is to extract a ligand bound to your target protein from the crystal structure used to provide the protein PDB file, i.e., the a co-crystalized ligand.

## Control Parameters

Run parameters may be set by editing the file `set_vc_workflow_control_vars.sh` This allows control of many aspects of the VM2 binding free energy calculations, such as the path/name of the project directory, how many cpu cores to allocate to each calculation, how to assign partial atomic charges, how much of the protein should be mobile, and so on. The following are example parameters that set the resource mamager as SLURM, targeting the queue name 'all', and request a single node calculation, using 12 MPI processes for each calculation.

```
export VCPROJECTDIR=$PWD/yourvm2project
...
export QUEUETYPE='slurm'
export QUEUENAME='all'
...
export NUMNODES='1'
export NUMMPIPROCS='12'
export NUMOPENMPTHREADS='0'
export NUMGPUS='0'
...
```

Keyword definitions are documented in comment blocks within the `set_vc_workflow_control_vars.sh` file itself. Further examples are given in the sections directly below. A full list of available keywords (control variables) are listed at the end of this document here along with a brief description of each one's purpose.

## Molecular System Setup

The workflow setup invoked by `vc_workflow_1_setup.sh` carries out basic molecule preparation e.g. H-atom addition, ligand 2D to 3D conversion, forcefield atom typing/parameter assignment, assignment of partial atomic charges if necessary, as well as any requested optimization/relaxation of provided protein receptor structures. The setup step will be run entirely on the users local machine and will take several minutes to complete with the default settings. If a Conda installed AmberTools is being used, care must be taken that the appropriate Conda environment is set before proceeding.

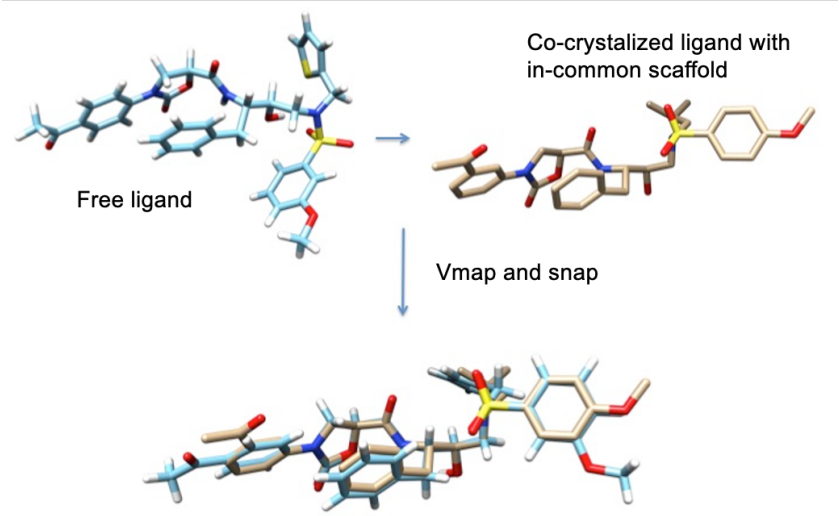
The following are examples of control parameters for the setup step, which set AmberTools as the protein and ligand preparation/typing/parameter assignment tool, the protein forcefield as 'ff14sb', the ligand forcefield as 'gaff2', VeraChem's Vconf tool for ligand 2D to 3D conversion, VeraChem's Vcharge tool for partial charge assignment, and so on:

```
export PROTEINFORCEFIELD='amber'
export PROTEINPREP='ambertools'
export PROTEINTYPER='ambertools'
export AMBERFORCEFIELD='ff14sb'
...
export REALATOMCUTOFF=6.0
export LIVEATOMCUTOFF=4.0
...
export LIGANDFORCEFIELD='gaff2'
export LIGANDPREP='ambertools'
export LIGANDTYPER='ambertools'
export LIGAND2D3D='vconf'
export LIGANDCHARGER='vcharge'
...
```

## Ligand Conformer Pre-generation

The script `vc_workflow_2_ligconfs.sh` invokes the pre-generation of starting conformers for the ligand series. If being used, one option for the ligand conformer pre-generation step is a 'map and snap' setting,

```
export MAPLIGTOTEMPLATE='vmap'
export PREGENLIGCONFS='snap'
```

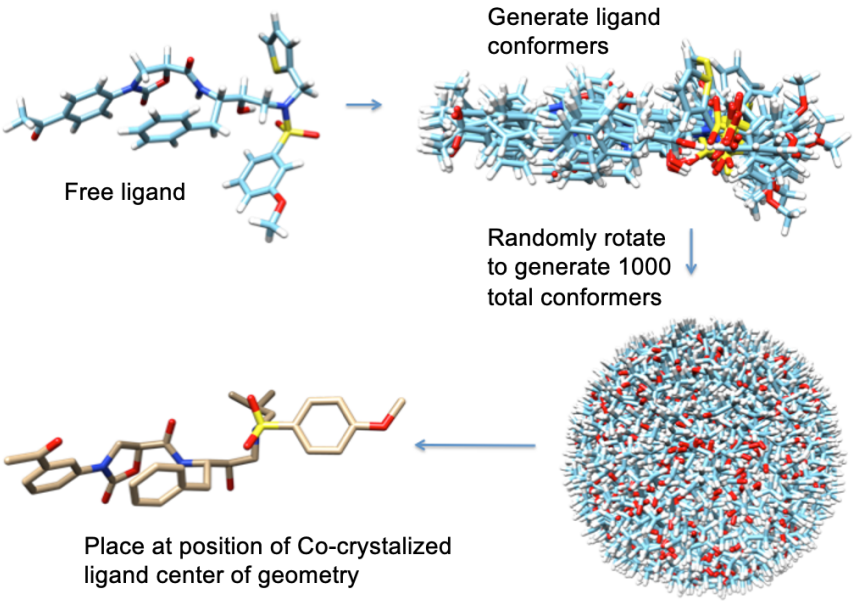



which uses VeraChem's Vmap tool for automatic mapping of common scaffold atoms on to a template scaffold e.g. co-crystalized ligand scaffold, followed by superposition of the common atoms with a guiding potential that 'snaps' the common scaffold into place. The results of a requested 'map and snap' step will be automatically extracted for use in later steps. They can be found in the directory:

```
/yourvm2project/run1/calculations/snap
```

The second option is the generation of a diverse set of conformers with which to seed the full VM2 conformational search when only the location of the active site is known through a co-crystalized ligand i.e. it does not share the same scaffold/general pose as the ligand series.

```
export MAPLIGTOTEMPLATE='none'
export PREGENLIGCONFS='random'
```



This is achieved through generation of ligand conformers, which are then randomly rotated around their center of geometry and placed at the center of geometry of the template ligand e.g. co-crystalized ligand. The results of a requested 'random' pre-generation of ligand conformers will, as in the 'map and snap' case, be automatically extracted for use in later steps. They can be found in the directory:

```
/yourvm2project/run1/calculations/random
```

The conformer pre-generation step takes approximately 30 seconds per ligand. It can be run locally or with faster turnaround on a cluster through a resource manager (i.e., SLURM or PBS).

**Note** that the ligand conformer pre-generation step can be skipped altogether, and the following parameter settings given in the control variable file,  if the user is supplying ligand series coordinates that are *already* good starting poses in the binding site.

```
export MAPLIGTOTEMPLATE='none'
export PREGENLIGCONFS='none'
```

## Generation of Run Directories

Running the script `vc_workflow_3_genrundirs.sh` builds and populates the directories in which the individual free energy (or chemical potential) calculations are cariied out. For a protein-ligand series workflow the directory hierarchy has the form:

```
/yourvm2project/run1/calculations/vm2/complexes
/yourvm2project/run1/calculations/vm2/complexes/proteinname/protein_name--ligand1
/yourvm2project/run1/calculations/vm2/complexes/proteinname/protein_name--ligand2
/yourvm2project/run1/calculations/vm2/complexes/proteinname/protein_name--ligand3
...
/yourvm2project/run1/calculations/vm2/ligands
/yourvm2project/run1/calculations/vm2/ligands/ligand1
/yourvm2project/run1/calculations/vm2/ligands/ligand2
/yourvm2project/run1/calculations/vm2/ligands/ligand3
...
/yourvm2project/run1/calculations/vm2/protein
/yourvm2project/run1/calculations/vm2/protein/protein_name
```

with each of these directories containing the required input files and a run script for either submission to a resource manager ( `runvm2.slurm` or `runvm2.bsh` ) or the local machine ( `runvm2.bsh` ). For example, the following is a typical set of required files for running an individual protein-ligand VM2 free energy (chemical potential) calculation:

```
protein_name.crd
protein_name.mol
protein_name.top                  <---- forcefield topology/parameters
ligand1.crd
ligand1.mol
ligand1.top                       <---- forcefield topology/parameters
ligand1_snap.xyz                  <---- pre-generated ligand conformers
protein_name_ligand1_vm2.inp      <---- calculation settings
protein_name_tethered_atoms.txt   <---- list of protein atoms to be constrained
runvm2.slurm*
```

## Calculation Submission

The script `vc_workflow_4_run.sh` initiates the submission of the individual free energy calculations, stepping through each run directory generated by `vc_workflow_3_genrundirs.sh` and executing, depending on the users choice, the `runvm2.slurm`, `runvm2.pbs` or `runvm2.bsh` script in each. This step comprises the vast majority of the computational effort in the VM2 workflow. For the default settings, the time for an individual protein-ligand or protein-only free energy calculation, using 12 CPU cores, for example, is roughly 20-30 minutes, and is approximately 5 minutes for a ligand-only run.

**Note** that every VM2 free energy calculation in this step of the workflow is entirely independent; therefore, very fast throughput of protein-ligand or host-guest series binding free energy calculations can be achieved by running even on a modestly sized cluster. For example, on a cluster with 72 available CPU cores (e.g. 6x12-core nodes), a throughput of **12 to 18 protein-ligand free energies per wall clock hour** can be achieved.

Calculations submitted through a resource manager should be monitored by the usual commands e.g. `squeue` for the SLURM resource manager and `qstat` for the PBS resource manager. It is also advisable that some of the run directories be examined and output files checked soon after submission to confirm functioning calculations.

## Results Extraction

After all the submitted individual free energy calculations, i.e., complexes, ligands, and receptor, are completed, on running the script `vc_workflow_5_extract_results.sh` the results will be collected, summarized, and placed in the directory:

```
/yourvm2project/run1/results
```

The receptor-ligand series binding free energies and their energy components can be found in a calculation summary CSV file, along with the corresponding experimental binding affinities, if they were provided by the user in the raw input data directory. In addition, the free energies and component energies of the separate complexes, ligands, and receptor are available in CSV files, e.g., for protein-ligand:

```
/yourvm2project/run1/results/protein_name_vm2_SUMMARY.csv   <---- Binding free energies

/yourvm2project/run1/results/protein_name_complex_vm2.csv   <---- Free energies of complexes
/yourvm2project/run1/results/protein_name_protein_vm2.csv   <---- Free energy of protein
/yourvm2project/run1/results/vm2_ligand.csv                 <---- Free energies of ligands
```

Conformers generated for the complexes, ligands, and receptor during the VM2 calculation are also presented:in a number of file formats, allowing for easy viewing in a third party tool of your choice (e.g. MOE, VMD, Chimera, etc.) For protein-ligand series calculations, they can be found in the following directories:

```
/yourvm2project/run1/results/conformers/complexes
/yourvm2project/run1/results/conformers/protein
/yourvm2project/run1/results/conformers/ligands
```

## 4.3 VM2 Workflow Script Generator

The installed VM2 package includes a command line question and answer 'wizard' application that provides a straightforward route for generation of the VM2 workflow scripts for a given use case; i.e., chosen molecular system, type of calculation, type of resource manager, number of compute cores to use, forcefields to use, system setup choices, etc. This VM2 workflow script generator 'wizard' application is called `generate_vm2_workflow_scripts.sh` and can be found in the `$VCHOME/exe` directory of the VM2 installation. Prior to running the script generator it is recommended that the raw input data directory structure be set up and populated.

On running the script generator via the command `$VCHOME/exe/generate_vm2_workflow_scripts.sh` the following message and initial question will appear:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Welcome! VeraChem VM2 Workflow Scripting Wizarding shall now begin ...

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Answer a few questions about your molecular system and desired VM2 workflow.
A set of shell scripts for running your workflow steps will then be generated.

**Note**  This helper tool is most effective if a raw input data directory
location is supplied when asked for, as this enables additional validation of
the proposed workflow settings, and it also reduces/removes the need to edit
the generated scripts. (Though, this tool can certainly be used to generate
a set of template scripts for later editing if desired by the user.)

Before proceeding, do you want to see the general form of the raw input data
directory structure? Y/N
```

Simply answer each new question to move onto the next section. Once all the wizard question/answer sections are completed, it will automatically generate the workflow control file and five workflow scripts discussed above:

```
set_vc_workflow_control_vars.sh
vc_workflow_1_setup.sh*
vc_workflow_2_ligconfs.sh*
vc_workflow_3_genrundirs.sh*
vc_workflow_4_run.sh*
vc_workflow_5_extract_results.sh*
```

To make changes later or adjust settings not covered by the wizard question/answers you can directly edit the parameter control file `set_vc_workflow_control_vars.sh`. See the control parameter reference section for details of available parameter settings. Also, the generated workflow control file and scripts can be copied and edited to repurpose them for use with other systems.

# 5 Best Practices

Not surprisingly, much like other free energy software/methodologies, the achievement of accurate binding free energy predictions using VM2 has a significant dependence on the quality of the setup of the receptor and ligand molecular models, as well as selection of calculation settings that are suited to the molecular system. The following sections suggest basic best practices regarding molecular system input data, setup for calculations, common issues that may affect accuracy, and possible tuning of calculation settings.

## 5.1 Input Data

### Protein receptor

Construction of molecular models for VM2 requires good receptor and ligand structures. For protein receptors this means a good quality (high resolution) X-ray crystal structure in the standard PDB format. If structures from the official Protein Data Bank are being used i.e. not in-house generated structures, the various repositories e.g. RCSB PDB, PDBe, and PDBj provide validation scores and analysis tools. Alternatively, re-refined structures may be considered e.g. PDB-REDO.

It is currently ***required*** that some basic editing of the PDB file be carried out before supplying it to the VM2 workflow raw input data directory. For example, any cofactors or non-metal ions should be removed, and likewise any water molecules should be removed, unless they play a role in ligand binding, in which case they can be left in the PDB file and included explicitly in VM2 setup and subsequent calculations. Co-crystalized ligands should also be removed from the protein PDB file. In general, the AmberTools manual section "Preparing PDB Files" describes well the necessary hand-edits to PDB files that will allow subsequent workflow automated protein model setup to succeed.

### Template ligand

If the protein PDB file contains a co-crystalized ligand that is similar in structure (i.e., shares a common scaffold or substructure) to the ligand or ligand series being considered (i.e. the target ligand(s)), it is highly recommended that it be extracted into its own PDB file and supplied to the VM2 workflow raw input data directory as a template ligand on which the target ligand(s) can be superposed, unless the user already has a suitable template from another source. Even if the co-crystalized ligand does not share a common substructure with the target ligand(s) it can still serve as template ligand that simply identifies the location of the binding site, if an alternative template is not available.

### Host receptor molecules/ligand series molecules

For host molecules or target ligand(s), the input data requirement is 2D or 3D structures in MOL or SD file format, with explicit bond orders, formal charges, and stereochemistry defined. Example 3rd-party tools to generate SD files from 2D drawings, SMILE strings, etc., are MOE, RDkit, DataWarrier, and there are many others. VeraChem offers a 2D drawing program, Vdraw , which is valence/stereochemistry aware and can seamlessly send 2D drawings for 2D to 3D conversion and output to SD files via VeraChem's Vconf application. An additional vital aspect of the host or ligand molecules supplied to the VM2 workflow is their protonation states. Currently the VM2 workflow itself cannot assign protonation states, so they must be assigned (possibly with the help of 3rd-party software) and set beforehand e.g. by formal charge placement and bond order assignment and/or hydrogen addition in the SD file supplied to the workflow.

### Experimental binding affinities

The VM2 workflow, if provided experimental binding affinity data, will, at the results extraction step, output these experimental affinities directly beside the associated calculated binding free energies, facilitating performance assessment via statistical analysis. When initiating a project applying VM2 to a protein target, if there is access to experimental binding affinity data for at least 10 ligands (preferably more) related to the eventual ligand target set (e.g., common substructure or scaffold), some initial validation/tuning runs are recommended to allow for possible optimization of the model and calculation settings for the best accuracy and efficiency.

## 5.2 System Setups

Given the required basic raw input data files just described, the VM2 workflow proceeds with further setup tasks to provide the final molecular models used in the VM2 free energy calculations. These tasks can be grouped into two stages. **Stage One** includes clean-up of problematic structures, normal structure preparation e.g. hydrogen atom addition, and forcefield atom typing/parameter assignment, and partial atomic charge assignment. **Stage Two** tasks are VM2 calculation related setup tasks, including generation of real and live atom sets, relaxation of protein structure around the binding site, possible ligand 2D to 3D conversion, and organization of the resulting molecular models in a directory structure for later use. The workflow can be set to carry out these tasks automatically by use of a combination of AmberTools and VeraChem software tools.

### AmberTools modules utilized

AmberTools modules are utilized exclusively in Stage One setup tasks:

- *pdb4amber* : clean up PDB structure and format for subsequent Amber typing and parameter assignment
- *tleap* : carry out Amber forcefield typing and parameter assignment
- *antechamber* : ligand atom typing and GAFF/GAFF2 parameter assignment; optional am1bcc partial atomic charge assignment

### VeraChem python-based setup tools

Various VeraChem software modules/tools are utilized in both Stage One and Stage Two setup tasks. The following are python based tools:

- *prm2top* : convert prtmtop files to VeraChem formatted files
- *vconf* : ligand 2D to 3D conversion
- *vcharge* : module for fast generation of ligand series partial atomic charges

- *vmap* : perform substructure matching

### VeraChem compiled language modules

In addition, the VM2 python workflow code interfaces with the VM2 free energy calculation engine, which has various submodules used for the following setup tasks:

- *snap* : snap target ligand scaffold atoms to template ligand scaffold atoms
- *geomoptHatoms* : relax protein hydrogen atoms
- *geomopt* : relax protein atoms in and around binding site
- *constructLiveReal* : carve out distance based real and live atom sets

### Setup outside the workflow

If the user prefers they can carry some or all of the Stage One tasks *outside* the workflow, read in the resulting pre-prepared files (Amber prmtop, incpcrd, and other standard formats) and proceed directly with Stage two. This is an example workflow where all Stage One protein and ligand preparation, including forcefield typing and parameter assignment, occurs outside the VM2 workflow:

```
$VCHOME/tutorials/vm2/workflow/protein_ligand/vm2pkg_pl_cmet_workflows
```

This option provides tremendous flexibility to the user, who can always use the available fully automated workflow, but can choose to use any 3rd-party tools they wish to trouble shoot/prepare their structures, etc., if they prefer.

## 5.3 Trouble shooting

There are a number of issues that are quite commonly found in PDB files as well as small molecule data files (e.g. MOL/SD files) that may lead either to outright failure of the VM2 setup stages to complete or to suboptimal performance of VM2 binding free energy calculations when compared to experiment. The following are issues to watch for with suggested approaches to address them.

### Protein: PDB file clean up/structure preparation

Protein PDB files, regardless of source, usually need some clean up before they can be used as a basis for any type of molecular modeling. If the fully automated VM2 workflow is being run with AmberTools for structure preparation etc. the *pdb4amber* tool will attempt to carry out necessary clean up, it is good practice, though, even after successful completion of setup stages in the workflow, to still check log files and prepared structures for problems, for example the following files will contain useful information:

```
/yourvm2project/setup/PrepareProteinStage1/run_pdb4amber.out
/yourvm2project/setup/PrepareProteinStage1/reduce_info.log
/yourvm2project/setup/PrepareProteinStage1/run_tleap.out
/yourvm2project/setup/PrepareProteinStage1/leap.log
```

Specific problems to look for and issues that may effect the quality of final binding free energy results are:

**Residue naming**

See the section "Residue naming conventions" in the AmberTools manual for details of expected naming when PDB files are being processed by *tleap* for atom typing and parameter assignment. Particular cases to look for are histidine naming that should reflect protonation state (**HIS**, **HID**, **HIP**); bridging cystines should be named **CYX**; neutral versions of the normally charged **ASP, GLU**, and **LYS** should be named **ASH**, **GLH**, and **LYN**, respectively; and neutralizing N- and C- terminal capping groups are named **ACE** (COCH3), **NHE** (CONH2) and **NME** (CONHCH3). Incorrect or inconsistent naming will likely cause failure of structure preparation and forcefield typing/parameter assignment. If *pdb4amber* is being utilized in the VM2 workflow, it will attempt to provide correct naming e.g. for two **CYS** residues with sulfur SG-SG atoms < 2.5 Angstrom apart found in the input PDB it will rename them to **CYX**, and for histidine groups it will assign the correct name ((**HIS**, **HID**, or **HIP**)) according to explicit hydrogens if they are present, but if they are not it will rename as **HIE**.

**Missing residue atoms**

If residue heavy atoms, partial side chains, or complete side chains are found to be missing they can either be added in their ideal geometry, or through rotamer libraries, to the system before typing and parameter assignment. If, however, they are very far away from the binding site or potentially will not even be present in the real/live set, they can be capped as GLY. The idealized geometry of any added side chain close to the active site should be relaxed sometime during the setup process. The *pdb4amber* tool will list missing heavy atoms, and the *tleap* does have some capacity to add missing atoms/side chains. If on inspection of *tleap* output and the newly modified protein structure, missing atoms problems persist, it is advisable to address this outside the workflow either with AmberTools directly (which exposes more options to the user) or with another 3rd-party software that can add missing heavy atoms/ side chains, e.g., MOE, Chimera, etc.

**Non-standard (modified) amino acids**

Any modified amino acid residues should typed and parameterized if 'official' parameters exist. If 'official' parameters do not exist and the residue is far from the active site it may be simply swapped out for a standard residue; however, if the residue is close to the active site it should be typed and charged using a generalized scheme. Again, the *pdb4amber* and *tleap* outputs should be examined and if necessary AmberTools should be used externally to the workflow to better utilize existent parameters (see the section 'Modified amino acids and nucleotides' in the AmberTools manual), or carry out non-standard parameter assignment etc.

**Metals ions and other non-standard residues**

It is recommended that non-metal ions and non-standard residues, such as ligands and co-factors, be removed from PDB files before running VM2 workflow based setup. If a metal ion exists directly in the active site of the protein and is involved in ligand binding, it is advised that, currently, VM2 should not be used to compute binding free energies (though development work in this direction is planned). For systems with metal ions far away from the active site or even in the region of the active, but not involved in ligand binding, it may be adequate to allow the default *pdb4amber*/*tleap* typing and parameter assignment, but more sophisticated treatments such as generation of quantum mechanics based parameters for the metal ion and surrounding residues would be preferable. (Automation of such schemes is a planned development.)

**Chain breaks**

If a chain break (gap) occurs near the binding site or in a loop that plays an important role then the missing residues should be added. Addition of missing loops will probably be best approached outside the workflow with 3rd-party tools. If the chain break is very far away from the binding site and likely to be fixed in space or even not present in the real/live set, then the termini of the break can be capped. The *pdb4amber* tool will identify chain breaks in its output, allowing manual addition of TER records in the PDB and then re-running of the workflow step will result in introduction of charged capping groups, N (NH3+) or C (COO-), by *tleap* . This will suffice for regions dropped out of the real set for VM2 calculations, and may suffice for termini inside the real set but far away from the active site, though addition of neutral caps (ACE or NME) would be preferable.

**Protein stereochemistry**

Stereochemistry errors that can occasionally occur in PDB files are *cis* peptide bond arrangements and incorrect alpha-carbon parity. It is preferable these issues are detected and fixed during the preparation stage. This requires manipulation of the PDB geometry coordinates. Structure visualization/analysis/manipulation tools (MOE, Chimera) should be used to correct stereochemistry errors, especially those in or close to the binding site.

**Specific protein residue protonation states**

There are five ionizable protein residue types: ASP, GLU, HIS, TYR, LYS. Their states are known at standard physiological pH 7.4, but the local pH in a binding pocket can vary according to the local structure. Also, it is possible that a ligand on binding will induce a change in the protonation state of binding pocket residues. Users may want to employ 3rd-party software that attempts to predict protein system protonation states by specific residue e.g. Propka. Currently, this should be done outside the VM2 workflow. Care should be taken to confirm to residue and atom naming conventions if submitting resulting adjusted PDB files to the workflow to complete the setup.

## Host and ligand molecules

Host molecules and especially ligand molecules provide further challenges with respect to preparation, typing, and parameter assignment. In contrast to protein systems, where constituent amino acids are specifically characterized and preparation tools can take advantage of this prior knowledge, generalized approaches must be used. Generalized preparation steps include hydrogen addition, protonation state assignment, bond order recognition, assignment of stereochemistry, and generalized typing and parameter assignment. As described above, the VM2 workflow expects host and ligand input data files in MOL/SD file format, with protonation states, bond orders, and stereochemistry already defined. Given this, the workflow setup invokes the *antechamber* tool to carry out further preparation tasks e.g. hydrogen atom addition if hydrogen atom not already present, before carrying out atom typing and GAFF/GAFF2 parameter assignment, and, if requested, *am1bcc* partial atomic charge assignment. The faster VeraChem *vcharge* tool is used by default for partial atomic charge assignment. Successful completion of *antechamber* and *vcharge* based setup steps relies heavily on correctly formatted definitions of bond orders, stereochemistry, and protonation states in supplied MOL/Sd files; furthermore, ultimately, chemically correct (or at least chemical sensible) assignments of these properties are vital to achieving accurate VM2 binding free energy predictions:

**Bond orders**

Bond orders must be supplied in host/ligand MOL/SD files as single, double, or triple bonds, i.e., Bond Block values of 1, 2 or 3.  Setting bond orders as aromatic (Bond Block value 4) will result in failed ligand setup. Furthermore, it is important to assign chemically 'correct' bond orders as they can effect force field parameter assignment, and in addition the VM2 algorithms make use of bond order information when performing conformational searches and other procedures.

**Stereochemistry**

The stereochemistry of the host/ligand molecules supplied through the MOL/SD file is propagated through to the VM2 calculations, which unless specifically requested not to, will maintain that stereochemistry throughout conformational searches etc. rejecting conformers that violate it. As protein-ligand binding affinities can have a strong dependence on ligand stereochemistry, it is important that correct stereochemistries are supplied from the start at the setup stage.

**Protonation states**

Protonation states can have a very large effect on predicted protein-ligand and host-guest binding free energies and their accuracies compared to experiment. If pKa values are known for the ligand under consideration (or for a similar chemical motif) then this may be adequate for good choices. If no pKa values are available, the use of third-party software for prediction of protonation states should be considered.

# 5.4 Molecular model and calculation considerations

## Protein receptor model

**Explicit solvent**

Protein PDB files usually include water molecules, with the water oxygen atoms indicating their positions. As already stated above, all water molecules that are not identified as playing a role in the binding of the co-crystalized ligand and potential target ligand series, should be removed from the PDB file before it is supplied to the VM2 workflow setup. Water molecules that are identified as important can be included in the VM2 calculation by simply leaving them in the PDB file supplied for setup, though a constraint should be applied to the water oxygen atom - see below for details.

Although the inclusion of explicit water may turn out to be required in order to achieve accurate calculated binding free energies for particular protein targets and ligand series, it is often far from obvious by initial visual inspection of structures. Users may want to utilize 3rd-party software that may algorithmically identify important water molecules in the binding site, for example Grid Inhomogeneous Solvation Theory (GIST), available in *cpptraj*, which is part of AmberTools. A more heuristic approach is to examine multiple X-ray crystal structures, with similar co-crystalized ligands (i.e. common scaffold) to the ligand target series, and look for conserved water molecules in the active site.

**Choice of protein real/live set**

The choice of protein atoms to include in the VM2 calculations (real atoms) and which of these should be mobile (live atoms) can have a large effect on the accuracy of VM2 predicted binding free energies. Enough protein atoms should be live so that the protein environment can adjust appropriately when the binding ligand is present. Furthermore, if the system under consideration contains a flexible loop that can affect the active site, as it the case for kinase systems, this loop should be mobile. Of course, the larger the real/live set is, the more computationally demanding each VM2 search step will be. In addition, the search space rapidly increases too, which can mean more conformational searching is required to achieve free energy convergence. So choosing a real/live set that is larger than needed should also be avoided.

The real/live atom set is defined by distance to a reference set of atoms in the binding site. The reference can be a bound molecule (e.g. co-crystalized ligand PDB coordinates), a group of molecules (e.g. a whole ligand series, already well placed and with good poses, possibly from docking), or a ligand superimposed on another smaller template with a common substructure/scaffold. All these exists as options in the VM2 workflow. A key consideration is that if the ligand series various in size, the defined real/live set must accommodate the whole range of sizes up to the largest ligand in the series.

Currently, the recommended starting real/live atom distance cutoffs are 6 Angstroms (real) and 4 Angstroms (live). On completion of the workflow step one, these atoms sets are available in PDB and mol2 formats, and so can be visualized and assessed. One possible issue to look for when there are large flexible ligands, is potential for part of the ligand to move behind the defined real set. In this case the real set can be increased, while initially retaining the same sized live set e.g. 8 Angstroms (real)/6 Angstroms (live).

**Relaxation of protein atoms in and around the binding site**

The relaxation of atoms in and around the binding site from their positions in the original PDB X-ray crystal structure can be important as it can relieve strain from near contacts and/or strain present because for the applied forcefield/energy model the PDB atoms are not at their minimum energy positions. In the workflow setup, if not turned off by the user, relaxation of hydrogen atom positions only is carried out by default in a region 12 Angstroms from the reference. Next, all protein atoms are relaxed in a region, by default, 6 Angstroms from the reference. The workflow allows for a bound ligand to be present during these relaxations, with the motivation for this being the avoidance of unwanted movement of protein atoms into parts of the binding site that may hinder ligand binding.

## Host and ligand models

**Partial atomic charge assignment**

The workflow offers a choice of VeraChem's electronegativity equalization method Vcharge for host and ligand partial atomic charge assignment, or the AM1-BCC method as implemented in the *antechamber* tool in AmberTools. In extensive benchmarking of protein-ligand and host-guest systems, the use of partial atomic charges from VeraChem's Vcharge tool did not affect the correlation of predicted binding free energies with experiment, but it is much faster. Therefore, the recommended default charging method in the VM2 workflow is Vcharge.

### VM2 calculation settings

The underlying VM2 calculation engine allows for fine control over many aspects of the VM2 algorithms. Currently the VM2 workflow exposes only a few of these: the level of conformational searching and the dielectric used for the continuum solvation models.

#### Conformational searching

The workflow default level of conformational searching - see options and default in the **Calculation engine control** section of the workflow control parameter values reference below - has been found reliable for protein-ligand series calculations where the ligand series share a common scaffold with an available co-crystalized ligand or where good starting poses are available for the ligand series, and also where the suggested 6 Angstrom (real)/4 Angstrom (live) atom sets have been used. For larger real/live sets than this it becomes more likely that higher levels of conformational searching will be required for reliable convergence. This is also the case where a template ligand in the right confirmation/pose with a common scaffold to the target ligand series is not available, and randomly oriented ligand conformers are placed in the binding site to generate starting conformers to seed subsequent conformational searching. See the following example workflow, which greatly benefits from increased conformational searching beyond the default settings:

```
$VCHOME/tutorials/vm2/workflow/protein_ligand/vm2pkg_pl_hivp_umass38_workflows/vm2_ukwnpose_example

$VCHOME/tutorials/vm2/workflow/protein_ligand/vm2pkg_pl_hivp_umass38_workflows/rawdata_hivp_umass38_xk2template
```

#### Continuum solvation dielectric

The Generalized Born (GB) and Poisson Boltzmann Surface Area (PBSA) continuum models are used to provide solvation energies during VM2 calculations. Testing to date has not shown that varying from the default dielectric constants (internal 1.0; external 80.0) can improve the correlation of VM2 predicted binding free energies with experiment.

## 5.5 Validation/Tuning

Upon the basis of VeraChem's own extensive VM2 benchmarking studies, for new systems it is highly recommended that if experimental binding affinity data is available for a set of ligands related to the target ligand series, validation/tuning studies should be carried out to establish model and calculation settings most favorable to producing good correlations with experiment. Some suggested workflow variations and steps to achieve validation/tuning are:

- Start with 6 Angstrom (real)/4 Angstrom (live) set calculation, no explicit water molecules, and default conformational search level.
- If water molecules conserved across X-ray crystal structures, include them a re-run the above workflow, if improved correlation with experiment retain the explicit waters in subsequent validation runs.
- Expand real/live set to 7 Angstroms/5 Angstroms. If improved correlation shown increase real/live to 8 Angstroms/6 Angstroms, but also increase level of conformational searching.

- Assess results across the workflow variations with respect to correlation with experiment and choose model/calculation settings to apply to target ligand series.

# 6 Further Workflow Details

For the most part the VM2 workflow is automated to the point where only keywords/control parameters need to be set. Currently, though, for a few capabilities additional details/formatted files need to be supplied by the user.

## 6.1 Explicit Water Molecules

To include explicit water molecules in protein-ligand VM2 calculations they should be included in the protein PDB file that is supplied to the workflow. For example, in the protein-ligand workflow tutorial referred to above in Section 4.1, the PDB file supplied in the raw input data protein directory

```
rawdata_hivp_umass5_ad81template/protein/2i0d_1580.pdb
```

a single water molecule oxygen atom is present at the end:

```
...
ATOM   1501  CE2 PHE B  99      30.167  22.169  29.272  1.00 46.75           C
ATOM   1502  CZ  PHE B  99      29.238  23.190  29.224  1.00 46.64           C
ATOM   1503  OXT PHE B  99      28.098  19.116  32.273  1.00 45.39           O
TER    1504      PHE B  99
HETATM 1580  O   HOH A 511      19.943  32.396  11.709  1.00 20.12           O  <-- water
END
```

and will be processed in workflow setup steps and included in the VM2 calculations. It is important that included explicit water molecules remain about in the same region across protein-only and protein-ligand complex conformers, so a constraint on the water oxygen atom is recommended - see next section.

## 6.2 Adding Geometric Constraints

Currently a harmonic constraint will be automatically setup and applied to a receptor atom (or atoms) in the workflow if a formatted text file, which must contain `tethered_atoms.txt` in its name, is included in the raw input data directory e.g.

```
rawdata_hivp_umass5_ad81template/protein/2i0d_tethered_atoms.txt
```

Below is the format of the file, with ...

```
#Tethered Atoms information
#numProtein
```

```
1
#numLigand
1

#proteinid
1

#setid
1
# water oxygen         <---- Comment only, can edit
#numTetheredAtoms
1
#atomList
3135                   <---- Atom index in the prepared protein

#ligandid
1
#numTetheredAtoms
0
#atomList

#end
end
```

For an example of host molecule constraint file see, for example, see the raw input data hosts directory for the host-guest series tutorial mentioned in Section 4.1:

```
rawdata_hg_cb7_gilson_set/hosts/cb7_tethered_atoms.txt
```

## 6.3 Experimental Data File Format

As described above, experimental binding affinity data can, if the user wishes, be supplied to the VM2 workflow, and then on completion of the full binding free energy calculation workflow, it is included in the same CSV file that summarizes the calculated binding free energies, and component energies, for the receptor-ligand series. This facilitates assessment of VM2 accuracy for the given molecular system, setup, and run parameters. The experimental data is supplied to the workflow through a formatted text file in the raw data directory:

```
rawinputdatadir/experiment/experimental_data.txt
```

(**Note**: the file name must contain `experimental_data.txt`, but the name can be extended as desired by the user e.g. `system_name_experimental_data.txt`)

The formatting of the file for a receptor and a series of ligands is a row for each complex containing the receptor name (or substring of the receptor name) linked by a double dash to the ligand name (or unique substring of the ligand name), comma separated with the experimental binding affinity in kcal/mol. For a protein-ligand series the basic form, then, is as follows:

```
protein_name--ligand1, affinity1
protein_name--ligand2, affinity2
protein_name--ligand3, affinity3
...
```

The workflow examples (tutorials) provide actual examples. For example, see the file `experimantal_data.txt` in:

```
$VCHOME/tutorials/vm2/workflow/protein_ligand/vm2pkg_pl_hivp_umass5_workflows/rawdata_hivp_umass5_ad81template/experiment
```

For two different host molecules and a common set of three ligands (guests):

```
host1_name--ligand1, affinity1
host1_name--ligand2, affinity2
host1_name--ligand3, affinity3
host2_name--ligand1, affinity4
host2_name--ligand2, affinity5
host2_name--ligand3, affinity6
```

Again an actual example can be seen in the workflow tutorials. See the file `sampl5_oa_experimental_data.txt` in:

```
$VCHOME/tutorials/vm2/workflow/host_guest/Sampl5/rawdata_sampl5_oa/experiment
```

# 7 Control Parameters Reference

**Usage:** Set the control parameter environment variables (left hand column of the reference tables below) in the file set_vc_workflow_control_vars.sh using the 'export' command e.g.,

```
export MOLSYSTEMTYPE='protein+ligand'

export CALCNTYPE='vm2'

export VCWORKFLOWEXE=$VCHOME/exe/VM2.pyc

...
```

Defaults refered to in the reference tables are the underlying default values set by the VM2 workflow script generator.

| Molecular system | Options | Description of resulting action |
|---|---|---|
| | | |
| **MOLSYSTEMTYPE** | **protein+ligand** | Calculate protein-ligand complex binding free energies/energies |
| | **host+ligand** | Calculate host-guest complex binding free energies/energies |
| | **ligand** | Calculate ligand-only free energies/energies |

| Calculation type | Options | Description of resulting action |
|---|---|---|
| | | |
| **CALCNTYPE** | **vm2** | 2nd generation mining minima free energy calculation, which carries out an exhaustive conformational search, calculates configuration integrals for individual conformers, and carries out a Boltzmann weighting to give a total free energy. |
| | **feprocess** | Process a set of already generated molecular conformers for total free energy. |
| | **confsearch** | Carry out a conformational search providing potential energies of the resulting conformers. |

| Workflow exe, data, and project dirs | Options | Description of resulting action |
|---|---|---|
| | | |
| **VCWORKFLOWEXE** | */path* | Full path and name of python workflow exe. Default is **$VCHOME**/exe/VM2.pyc |
| **VCRAWDATADIR** | */path* | Full path of directory containing raw data such as: protein PDB file, ligand mol files, experimental data file, ... etc. |
| **VCPROJECTDIR** | *./name* | Project directory to be used/created e.g. $PWD/hivp_umass5_vm2_coxtal |

| Calculation resources | Options | Description of resulting action |
|---|---|---|
| | | |
| **VCCUSTOMEXE** | */path* | VeraChem calculation engine exe custom location: full path to directory. (Optional, rarely needed) |
| | | |
| **QUEUETYPE** | **slurm** | Use a SLURM resource manager. This is the **default**. |
| | **pbs** | Use a PBS resource manager. |
| | **bsh** | Generate bash run script and run jobs sequentially on local machine. |
| | | |
| **QUEUENAME** | *name* | Set the queue/partition *name* to use for pbs or slurm jobs. For your system's default queue/partition set as '**default**'. |
| **NUMNODES** | *n* | Number of compute nodes to use for receptor (protein or host) involved calculations. Default is **1**. |
| **NUMMPIPROCS** | *n* | Total number of MPI processes to use for receptor involved calculations. Default is **12**. |
| **NUMOPENMPTHREADS** | *n* | Number of OpenMP threads to use per MPI process. If set to 0 a pure MPI run is performed. Default is **0**. |
| **NUMGPUS** | *n* | Total number of GPUs to use for receptor-included runs. Default is **0**. |
| **NUMLIGNODES** | *n* | Number of compute nodes to use for ligand calculations. Default is **1**. |
| **NUMLIGMPIPROCS** | *n* | Total number of MPI processes to use for ligand calculations. Default is **12**. |
| **NUMLIGOPENMPTHREADS** | *n* | Number of OpenMP threads to use per MPI process. If set to 0 a pure MPI run is performed. Default is **0**. |
| **NUMLIGGPUS** | *n* | Total number of GPUs to use for receptor runs. GPU use for ligand calculations not recommended. Default is **0**. |

| Protein setup | Options | Description of resulting action |
|---|---|---|
| | | |

| | | |
|---|---|---|
| **PROTEINFILENAME** | *name*.**pdb** or *name*.**prmtop** | Protein PDB (or .prmtop) file *name* present in **$VCRAWDATADIR**/protein. This is required. |
| **PROTEINREFLIGAND** | *name* or substring of *name* | Ligand *name* or substring of ligand *name* chosen from the raw data **$VCRAWDATADIR**/ligands directory. If given, this ligand will be superimposed on the template ligand in **$VCRAWDATADIR**/template_ligand and be will be present during any requested protein relaxation during setup. It must be given if **$PREGENLIGCONFS** is set as '**snap**' - see ligand setup below. |
| **PROTEINFORCEFIELD** | **amber** | Currently only the default forcefield '**amber**' is available in the workflow. |
| | | |
| **PROTEINPREP** | **ambertools** | Use AmberTools for basic protein preparation. |
| | **none** | Pre-prepared protein parameter/topology files must be supplied in **$VCRAWDATADIR**/protein. Namely, .prmtop, .inpcrd, .mol2, and .pdb files. Also, the .prmtop file name must be set through **$PROTEINFILENAME** (see above) and the other files must share its base name. |
| | | |
| **PROTEINTYPER** | **ambertools** | Use AmberTools for protein typing/parameter assignment. |
| | **none** | Pre-prepared protein parameter/topology files must be supplied in **$VCRAWDATADIR**/protein. See **PROTEINPREP** entry above. |
| | | |
| **AMBERFORCEFIELD** | **ff99sb** | Choose the AMBER99SB forcefield parmeter set. |
| | **ff14sb** | Choose the AMBER14SB forcefield parameter set. |
| | **none** | No specific choice of Amber forcefeld. Used when pre-prepared Amber parameter/topology files supplied in **$VCRAWDATADIR**/protein. See **PROTEINPREP** entry above. |
| | | |
| **LIVEREALREFCHOICE** | **proteinreflig** | As a reference to carve out the set of protein atoms to be included (real) in the VM2 calculations and included plus mobile (live), use the ligand set by **$PROTEINREFLIGAND** - see above. |
| | **file** | Indicates a file path to a reference molecule/structure to be used for real/live atom definition will be given by setting **LIVEREALREFFILE** - see below. |
| | **dir** | Requests that the union of all atomic coordinates of all structure files (MOL, SDF) present in **$VCRAWDATADIR**/live_real_ref (if it exists) be used as the reference. |
| | **template** | Use the template provided in **$VCRAWDATADIR**/template_ligand as the reference. |
| | | |

| | | |
|---|---|---|
| **LIVEREALREFFILE** | */path* | If **$LIVEREALREFCHOICE**='file' a full path to the file must be given. |
| **REALATOMCUTOFF** | *real value* | Cutoff distance to any **$LIVEREALREFCHOICE** atom that defines which protein atoms are included in the VM2 calculations. A suggested value to start at is **6.0** Angstroms. |
| **LIVEATOMCUTOFF** | *real value* | Cutoff distance to any **$LIVEREALREFCHOICE** atom that defines which protein atoms are live i.e. which atoms are treated as mobile. A suggested value to start at is **4.0** Angstroms. |
| **COMPLETELIVERES** | **on/off** | If set as **on** residues that contain live atoms are made fully live i.e. the remaining non-live atoms in the residue are automatically added to the live atom list. Default is **off**. |
| **MINLIVEFORCMPLT** | *n* | The minimum number of live atoms in a residue for it to be completed. Default is **5**. |
| **PROTEINOPTH** | **on/off** | If set as **on** as part of the protein setup optimize the positions of H atoms in the protein. Default is **on**. |
| **PROTEINOPTHCUTOFF** | *real value* | Cutoff distance to any **$LIVEREALREFCHOICE** atom that defines which H atoms will be geometry optimized during the protein setup. The default is **12.0** Angstroms. (A size no less than **$REALATOMCUTOFF + 2.0** is recommended.) |
| **PROTEINRELAX** | **on/off** | If set as **on** as part of the protein setup optimize the positions of cutoff defined atoms (see just below) in the protein. Default is **on**. |
| **PROTEINRELAXCUTOFF** | *real value* | Cutoff distance to any **$LIVEREALREFCHOICE** atom that defines which protein atoms will be optimized (relaxed) during the protein setup. The default is **6.0** Angstroms. (A size no less than **$LIVEATOMCUTOFF + 2.0** is recommended.) |

| Ligand setup | Options | Description of resulting action |
|---|---|---|
| | | |
| **LIGANDFORCEFIELD** | **gaff** | Use the general AMBER force field (GAFF) for ligands. |
| | **gaff2** | Use the second generation of the general AMBER force filed (GAFF2) for ligands. |
| | | |
| **LIGANDPREP** | **ambertools** | Use AmberTools for basic ligand preparation. |
| | **none** | Pre-prepared ligand parameter/topology files must be supplied in **$VCRAWDATADIR**/ligands. Namely, .prmtop, .inpcrd, .mol2 and .mol files for each ligand. |
| | | |
| **LIGANDTYPER** | **ambertools** | Use AmberTools for ligand typing/parameter assignment. |
| | **none** | Pre-prepared ligand parameter/topology files must be supplied in |

| | | |
|---|---|---|
| | | **$VCRAWDATADIR**/ligands. Namely, .prmtop, .inpcrd, .mol2 and .mol files for each ligand. |
| | | |
| **LIGAND2D3D** | **vconf** | Generates good **3D** ligand structures when supplied **2D** structures in MOL files or SD file in **$VCRAWDATADIR**/ligands. |
| | **none** | Set to '**none**' when all ligands in **$VCRAWDATADIR**/ligands are already good **3D** structures. |
| | | |
| **LIGANDCHARGER** | **vcharge** | Use VeraChem's Vcharge tool to generate partial atomic charges for ligands. |
| | **am1bcc** | Use the AmberTools am1bcc charging capabilty. |
| | **none** | Pre-prepared ligand parameter/topology files must be supplied in **$VCRAWDATADIR**/ligands. Namely, .prmtop, .inpcrd, .mol2 and .mol files for each ligand. |
| | | |
| **MAPLIGTOTEMPLATE** | **vmap** | Use VeraChem's Vmap tool to map input ligand atoms to the equivalent atoms in the supplied template in **$VCRAWDATADIR**/template_ligand. This allows ligand common scaffold matching and superposition. |
| | **none** | No mapping to the template structure atoms is carried out. |
| | | |
| **LIGSDFFROMINPCRD** | **-ligsdffrominpcrd** | Turns on overwriting of SDF coordinates using .inpcrd coordinates. Only use for direct from .prmtop, .inpcrd, etc. gaff runs or if .inpcrd file with coordinates you want to use is available. |
| | *blank string* | Given as: export **LIGSDFFROMINPCRD**=' '. No overwriting of input SDF coordinates. |
| | | |
| **PREGENLIGCONFS** | **snap** | Forces superposition of ligand atoms onto those in common in the template in **$VCRAWDATADIR**/template_ligand. |
| | **random** | Randomly orientated ligand conformers generated for placement at center of geometry of template. |
| | **none** | No pre-generation of ligand conformers is carried out. |
| | | |
| **USETEMPLATESDF** | **-usetemplatesdf** | If **$PROTEINREFLIGAND** use its template 'snapped' coordinates SDF for all other maps and snaps. |
| | *blank string* | Given as: export **USETEMPLATESDF**=' '. Always use the template in **$VCRAWDATADIR**/template_ligand for maps and snaps. |

| Calculation engine control | Options | Description of resulting action |
|---|---|---|
| | | |
| **MPISEARCHSTYLE** | **uncoupled** | All MPI processes carry out independent searches and only communicate their lowest energy conformers at the end of a search. |
| | **coupled** | Each MPI process communicates its lowest energy conformer and the lowest energy coordinates are then used by all processes as a new search basis. |
| | **mixed** | Uncoupled searches are performed on even VM2 iterations and coupled searches on odd ones. |
| | | |
| **MPIMAXUCPLDITERS** | *n* | Maximum number of iterations with '**uncoupled**' or '**mixed**' set. After it is reached the search style is reset to '**coupled**'. For **$MPISEARCHSTYLE** = '**uncoupled**' a value of **10** is recommended, otherwise **20** is recommended for '**mixed**'. |
| | | |
| **CONFSEARCHSTYLE** | **enhanced** | Single mode-distort-minimze and random pair-mode-distort minimize conformational searches are carried out for receptor-involved VM2 calculations e.g. protein, protein-ligand, host, and host-guest. |
| | **rigorous** | Adds ligand and protein focused mode searching to '**enhanced**'. This is the **default**. |
| | **vrigorous** | Further adds ligand rotation/translation searching. |
| | | |
| **MAXCONFSEARCHES** | *n* | Maximum number of conformational searches per search-type (e.g. single-mode, pair-mode) per VM2 iteration for receptor-involved VM2 calculations e.g. protein, protein-ligand, host, and host-guest. The default is **100**. |
| | | |
| **LIGCONFSEARCHSTYLE** | **minimal** | Single mode-distort-minimize conformational searches are carried out. |
| | **enhanced** | Single mode-distort-minimze and random pair-mode-distort minimize conformational searches are carried out. This is the **default**. |
| | **rigorous** | Adds focused mode searching to '**enhanced**'. |

| | | |
|---|---|---|
| | | |
| **MAXLIGCONFSEARCHES** | | Maximum number of conformational searches per search-type (e.g. single-mode, pair-mode) per VM2 iteration for ligand-only VM2 calculations. The default is **100**. |

| **Calculation engine solvation** | **Options** | **Description of resulting action** |
|---|---|---|
| | | |
| **DIELECINTERNAL** | *real value* | Internal dielectric used for GB and PBSA continuum solvation models. Default is **1.0** |
| **DIELECEXTERNAL** | *real value* | External dielectric used for GB and PBSA continuum solvation models. Default is **80.0** |

| **Experimental data** | **Options** | **Description of resulting action** |
|---|---|---|
| | | |
| **EXPERIMENT** | **-experiment** | |
| | *blank string* | Given as: export EXPERIMENT=' '. No comparsion to experiment will be made. |

| Results extraction | Options | Description of resulting action |
|---|---|---|
| | | |
| **EXTRACTFOR** | **humans** | Extract energy data and conformer structure formatted files for human manipulation i.e. manual loading of .csv's into Excel, sdf's, mol2's, pdb's into molecular viewers. |
| | **perconfpp** | Same as for '**humans**' setting but also generates extra data for possible post-processing runs. |
| | **moe** | Extract data for parsing and 'automated' manipulation/analysis by MOE. |
| | | |
| **MAXCONFSTOEX** | *n* | Maximum number of conformers per complex, ligand, and receptor calculation to extract from the run directories to the results directory. The default is **8**. |
| **POSTPROCCONFLMIT** | *n* | Maximum number of conformers to include in any post-processing. The default is **4**. |