Preparing large MD simulations with VMD

4 min. read · View original

One of the unique capabilities of VMD as compared with other molecular visualization and analysis packages is its ability to operate on very large biomolecular complexes containing hundreds of millions of atoms. VMD contains many built-in tools to assist with common structure building tasks such as assignment of MD force field parameters and topology, adding solvent, placing ions, and writing out the completed structure into fileis using molecular file formats that don't have various limitations that interfere with large structures.

While it is common practice to use the built-in graphical interfaces in VMD to perform structure building tasks on molecular systems of moderate size (1 million atoms or less), it is often beneficial, and sometimes crucial, to use script-based approaches when working on very large structures with tens to hundreds of millions of atoms, to overcome limitations that may exist in the easy-to-use graphical tools. While we work on addressing the limitations of the graphical tools in future releases of VMD, this tutorial should serve as a guide to assist people in using the tools as they exist today.

Solvating very large systems

The VMD "solvate" plugin was developed a long time ago before it became practical to simulate hundreds of millions of atoms. While it is possible to use the built-in plugin to solvate such large systems, the performance of the code is very low for large systems, when it is used in the traditional way, because it hasn't (yet) been rewritten to use new high-performance structure building features of recent versions of VMD and psfgen. In the short term, the basic steps outlined below can provide a significant performance increase over the normal solvation procedure, reducing a multi-day solvation run (for 100M atoms) down to just a few minutes, by using advanced features of psfgen and VMD that have not yet been incorporated into the graphical structure building tools.

The steps below are exemplary, and you will want to ensure that you add the appropriate script logic to trim the solvent box down to the exact size you actually need. Similarly, if you prefer to have the final solvated system centered on the origin, these are trivial steps you can easily insert into the outline below. The point of this script is to provide an example that demonstrates a high performance approach for efficiently solvating 100M atom complexes.

You must supply solute.js which should contain your proteins, nucleic acids, etc. prior to running these steps.

Molecular structures can be converted from other file formats to the .js binary structure/trajectory file format by loading all input files into VMD first, and then doing these steps to emit a .js file from a loaded molecule in VMD:

```
set sel [atomselect top all]
$sel writejs solute.js
$sel delete
```

Using the psfgen binary shipped with NAMD

Below are exemplary steps for using the standalone psfgen binary that is distributed with NAMD 2.9/2.10, which is always compiled with support for molfile plugins enabled.

```
# Create a 100nm x 100nm x 100nm water box (with no solute
added yet...)
# If we use the VMD solvate template water box, we need
about 16 copies
# in each direction.  If you use a different template water
box, or your
# VMD is installed in a different directory, you will need
to modify these
# filenames appropriately.  The number of copies along each
axis and
# the offset for each tile should match the water box
dimensions.
# run this in a Unix shell:
/usr/local/bin/psfgen << END
readpsf
/usr/local/lib/vmd/plugins/noarch/tcl/solvate1.6/wat.psf
coordpdb
/usr/local/lib/vmd/plugins/noarch/tcl/solvate1.6/wat.pdb
writemol js solventpad.js 16 {67.16 0 0} 16 {0 67.16 0} 16
{0 0 67.16}
END

# post-process the resulting merged solvent to remove out-
of-bounds
# waters, and fix up residue names, etc.
# run this in VMD:
mol new solventpad.js
set sel [atomselect top all]
$sel set resid [$sel get residue]
$sel delete
set good [atomselect top "not same residue as ((x < 0) or
(y < 0) or (z < 0) or (x > 1000) or (y > 1000) or (z >
1000))"]

# Here is where we would normally perform additional steps
such as
# re-centering the solvent box to the origin, if
necessary...
# for example:  $good moveby -500 -500 -500

$good writejs solvent.js
```

```
exit


# Load the solute and solvent, and merge them into a single
structure,
# and write out a new .js file containing the combined
(clashing)
# structure.
# run this in a Unix shell:
/usr/local/bin/psfgen << END
readmol js solute.js
readmol js solvent.js
writemol js clashes.js
END


# post-process the resulting merged solvent/solute
structure to remove
# all of the clashes.
# run this in VMD:
mol new clashes.js
set good [atomselect top "not same residue as (within 2.4
of not segname \"W.*\")"]
$good writejs final.js
$good writenamdbin final.coor
exit
```