

The LMOD optimization code has been updated with major improvements and new features including more accurate flexible docking, the option to visualize normal modes, utilization of random mixtures of low-frequency modes, and the option to work with a range of modes anywhere in the spectrum and not just the lowest frequency modes. The latter is particularly useful for docking where the modes relevant to binding a ligand molecule are usually not the lowest frequency modes. The interface of the new LMOD has not changed, everything works exactly the same way as in Amber18 and earlier versions, a few parameters simply have additional options as documented below. The new features are demonstrated with production quality examples.

1. Flexible docking has been improved under the hood by changing some of the internal logic of the work flow.
2. It is paramount to minimize a molecular structure to a very low gradient ( $\text{RMS} < 1\text{e-}08$  kJ/mol/Å) to avoid unnatural vibrational modes, however, such rigor is not necessary for minimizing the stored low-energy conformations. This distinction was not possible in the previous version of LMOD where a single parameter controlled both convergence criteria and resulted in wasting substantial CPU time in over-minimizing low-energy conformations. In the new LMOD one can specify the gradient RMS limit separately for the first minimization, which determines the quality of the modes, and for the minimization of the conformers. In the sander interface this means that the *drms* parameter in the *&cntrl* block refers to the minimization of the input structure for which the normal modes are computed, and can be set to  $\sim 1\text{e-}10$ . The *lmod\_minimize\_grms* parameter in the *&lmod* block refers to the convergence criterion to minimize low-energy conformations, and it is generally enough to set it to  $\sim 1\text{e-}01$ . Similarly, in the nab interface the stringent convergence criterion is set by the *xo.grms\_tol* parameter, and the loose criterion is set by *lo.minim\_grms*.
3. Arguably the most useful, new LMOD feature is visualization of the normal modes. Since the very essence of LMOD conformational search is utilizing the most efficient normal modes, visualization makes it possible to select an ideal range of modes (see 5.) instead of blindly relying on the set of the lowest frequency modes. In essence LMOD generates a quasi-trajectory representing smooth vibrational motion, and Pymol, VMD, or any other molecular graphics software can be used to animate the normal modes. The smoothness of the modes comes from being curvilinear as opposed to simple linear extrapolations of the vibrational eigenvectors. The curvilinear representation utilizes the ZIG-ZAG algorithm as described in section “LMOD procedure” (37.4.2 in the Amber18 manual and 39.4.2 in the Amber20 manual). Visualization can be requested by setting LMOD verbosity to *lmod\_verbosity* = 5 in the sander interface and *lo.print\_level* = 5 in the nab interface. Examples are provided for both interfaces in `$AMBERHOME/AmberTools/examples/nab/` (along with numerous other LMOD examples) in the `lmod_vib_anim/` subdirectory. The example directory has a detailed README file with information how to run these calculations. The nab script is completely general and after compiling it the executable can be used via command line arguments for any system. There is also a Pymol script that provides additional rendering in Pymol. The `sander/` subdirectory has a sander example, but it is lacking the rendering options available through nab objects.
4. Another new feature in LMOD is employing mixed modes. This option is completely automated and does not require any user intervention. At the end of every single LMOD iteration, after exploring all the selected pure modes, a random mixture/superposition of these modes is also explored, in both directions. Since the direction of any given mode is

indeterminate, a random mixture means superposing all modes included in a particular iteration such that each individual mode is multiplied by randomly picked +1 or -1. Mixed modes often find interesting, new conformations. Mixed modes are shown as mode number zero in the screen output such as the one shown in section “Sample LMOD program” (37.4.6 in the Amber18 manual and 39.4.6 in the Amber20 manual). Also note that mode visualization in 3. always includes the superposition of all modes included as the last, pseudo-mode.

5. Finally, the user has the option of exploring a range or band of modes instead of simply using the  $N$  lowest frequency modes. This is especially useful for flexible docking calculations, or in any situation where the modes relevant to a particular conformational change are not the lowest frequency ones. For example, given a protein complex, mode visualization will vividly show that the lowest frequency modes involve large scale motions where the individual molecules move together and these modes are irrelevant regarding the binding/unbinding. Similarly, the lowest frequency modes of a protein-ligand complex typically show the ligand as a rigid body, sort of surfing the waves of the large scale domain motions of the protein, and have nothing to do with binding poses. Mode visualization can help identify the first relevant mode  $k$ , and then LMOD can be instructed to explore modes  $k$  to  $N$  instead of 1 to  $N$ . This is simply done by an extended use of the *number\_free\_rotrans\_modes* (sander) or the *lo.nrotran\_dof* (nab) parameters. This parameter is normally set to 6 to skip the trivial ro-translational modes, but the new LMOD allows to set it to any non-negative integer value to skip an arbitrary number of modes, thus focusing on higher frequency vibrations. (It is instructive to also set it to zero once, to actually see that ro-translational modes are not necessarily pure translations or rotations, but can be superpositions such as when a nut is sliding up and down on the bolt along the spiral thread.)