This example is a large scale, production quality LMOD conformational search on cyclosporine. The goal of this search is generating a representative ensemble of conformations for docking. Fast docking programs such as Autodock or Glide claim to be able to handle flexible ligands, but in fact their tolerance to flexibility is very low, moreover it is virtually impossible to dock flexible macrocyclic molecules such as cyclosporine, including ring flexibility. A more practical solution, is running (LMOD) conformational search to generate an ensemble of conformations and then screen them via ultra fast rigid body docking to select favorably scoring poses. In this particular example we used 1cwa.pdb where cyclosporine is docked to cyclophilin A. We start with an arbitrary conformation in csp.pdb and apply 5,000 LMOD iterations to generate an ensemble of about 1,800 distinct conformations. Via a user defined parameter (see below) the resulting ensemble is generated with a diversity constraint such that no two conformations are closer to each other than 2.0 Å all-atom superposition RMS in the set. Note the striking difference between the two pictures in cyclosporine_confs.png showing the superimposed ensemble of conformations generated by two similar searches. The left hand side is the result of a search using the previous version of LMOD (prior to AmberTools 16) and the right hand side is the result of the current search. The new version of LMOD gives the user a powerful control parameter to gauge the diversity of the conformations saved at the end of the search.

As a side note—this calculation is not included in the example—it would still be impractical to dock 1,800 conformations but there is a way to significantly lower the number of conformations to be docked, via cluster analysis. The basic idea is as follows. Let's say, we have 1,500-2,000 conformations of a molecule. Instead of docking all of them we can generate something like 50-100 clusters, and in the first iteration we only dock a single cluster representative (the closest conformation to the cluster centroid) per cluster. Then, assuming that docking will score best the closest conformations to the native (which of course would be unknown in the production project), we would re-dock all the members of the corresponding clusters whose representatives scored well. Of course, this is not exact but we have to assume that docking will generally score better the conformations closer to the (unknown) native than conformations farther away, otherwise the whole project is dead on arrival. Therefore, we can assume that out of the 50-100 clusters the first 5-10 contains most of the native like conformations, and re-docking them should give a representative ensemble of good docking poses. So, with this procedure the number of conformations to be docked can be kept within the range of a few hundred and that should be perfectly feasible computationally.

It is important to note that the gaff force field applied here to cyclosporine is rather generic and is not expected to render the native, bound conformation to be a particularly low energy (standalone) conformation. In fact, there is a striking difference between the near planar conformation seen in 1cwa.pdb and the low-energy, and very much folded conformations occupying the top part of the LMOD generated conformational ensemble in csp_lmod_conf_1452166579.pdb (see also csp_run_mpinab.log to examine the search log). The immense difficulty of this conformational search comes from the fact that because the number of conformations grows exponentially with the energy gap, and the native, near planar conformation is about 20 kcal/mol higher in energy than the global minimum, it is virtually impossible to find even a representative subset of all conformations within 20-25 kcal/mol above the global minimum. Fortunately, nonetheless, with the optimized LMOD settings used in the csp.nab script it is possible i) to make the LMOD search hover around 20 kcal/mol above the global minimum during the entire search, and ii) also to diversify the conformations to guarantee that every single pair of saved conformations will be displaced by at least some user defined (1.5-2 Å) all-atom superposition RMSD. With these settings—focusing on the high plateaus of the potential energy surface rather than the low lying valleys—it was possible to find conformations of cyclosporine ~1 Å RMSD from the native pose seen in 1cwa.pdb.

To run this job, one simply needs to compile csp.nab (which will read csp.pdb and csp.prmtop). It is highly recommended to compile it in parallel using mpinab, and run it it on 4-8 cores. 5,000 LMOD iterations with 50,000 total LMOD search steps took just about a day running on 8 cores of a 3.6 GHZ i7 processor. LMOD scales quite well thanks to the efficient force parallelization engine in AmberTools. The Antechamber work files utilized for generating charges and gaff parameters are also included in this example, and the procedure is as follows.

```
$ antechamber -i csp.pdb -fi pdb -o csp.mol2 -fo mol2 -c bcc
$ parmchk2 -i csp.mol2 -f mol2 -o frcmod
$ tleap
> source leaprc.gaff
> mods = loadAmberParams frcmod
> mol = loadMol2 csp.mol2
> set default PBRadii mbondi3        # this is necessary for using gb=8 GB model
> saveAmberParm mol prmtop inpcrd
> quit
$ $AMBERHOME/AmberTools/src/etc/lmodprmtop prmtop prmtop.lmod
$ mv prmtop.lmod csp.prmtop
```

The structure of a typical LMOD nab script and associated job logfile is fully explained in the main AmberTools documentation under NAB: Molecular Dynamics and Mechanics/Low-MODe (LMOD) optimization methods, here some specifics are pointed out.

1. It is good practice to use a hardware generated seed for the random number generator and save this seed value in the file names associated with an LMOD job. In the example script the number of seconds passed since zero hour, January first, 1970 is used for this purpose.
2. Noteworthy LMOD parameters: `lo.nmod=20` is the number of lowest frequency modes used, `lo.kmod=5` means that out of the 20, every new LMOD iteration explores 5 randomly selected modes (and follows them in both directions), and `lo.nrotran_dof=6` refers to the absence of any frozen (or tethered) atoms. `lo.energy_window=20` is set intentionally high to generate a variety of different conformations and not only focus on the lowest energy ones. `lo.conf_separation_rms=2.0` means that in the final set of LMOD generated conformations every single pair of them will be at least 2.0 Å superposition RMSD apart. This RMSD calculation includes all atoms (LMOD has no knowledge of atom types, etc.). This parameter is quite useful in controlling the diversity of low-energy conformations. `lo.nof_lmod_steps=0` instructs LMOD to try determining a barrier passing event automatically (see documentation). For example, something like this in the logfile
   ```
   5 /10  E = 209.357 ( 0.075)  Rg = 1.996  rmsd= 2.326  p= 1.0000
   5 / 4  E = 209.016 ( 0.081)  Rg = 1.865  rmsd= 1.040  p= 1.0000
   ```
   means that starting from a particular conformation, low-frequency mode #5 is explored in both directions. In one direction LMOD determined that after 10 LMOD steps (so-called zig-zag curvilinear perturbation, see documentation) a barrier was passed whereas the other direction required only 4 steps. After minimization, the resulting new conformations are, respectively, displaced by 2.326 and 1.040 Å RMSD from the same starting conformation. Rg is the radius of gyration and p is the Boltzmann probability of a conformation with respect to the starting conformation in that particular LMOD iteration.
3. `lo.mc_option=1` sets the Monte Carlo search to “Metropolis” which means that the

traditional Metropolis criterion is applied to the minimized energies. Note that the temperature is set to a very high value `lo.rtemp=3.0` which means that a high energy gap $E - E_0 = 3$ [kcal/mol] still has a significant Boltzmann probability of $e^{-1} \approx 37\%$.

With any questions or comments please contact me at istvan@kolossvary.hu or ikolossv@bu.edu.