Supplementary Information for

VinaMPI: Facilitating Multiple Receptor High-Throughput Virtual Docking on High Performance Computers

S. Ellingson; J.C. Smith, J. Baudry
UT/ORNL Center for Molecular Biophysics

VinaMPI Tutorial

Compiling VinaMPI (these directions were used on the Kraken supercomputer)

1. Download the revised Autodock Vina code and VinaMPI (install directory = $INSTALL_DIR)

   ~> cd $INSTALL_DIR
   ~> wget cmb.ornl.gov/~sek/autodock_vina_1_1_2.tgz
   ~> wget cmb.ornl.gov/~sek/VinaMPIv2.tgz
   ~> tar xzvf autodock_vina_1_1_2.tgz
   ~> tar xzvf VinaMPIv2.tgz

2. Compile the Autodock Vina code

   ~> cd autodock_vina_1_1_2_rev/autodock_vina_1_1_2/build/linux/release/
   ~> module swap PrgEnv-pgi PrgEnv-gnu
   ~> module load boost
   ~> make clean
   ~> make depend (this step may give warnings but still be ok)
   ~> make

3. Direct VinaMPI to the Autodock Vina code and compile VinaMPI

   ~> cd $INSTALL_DIR/VinaMPIv2_new2/
   ~> vi vinampi.work.c

   go to line 279 and change the path to Autodock Vina to
   $INSTALL_DIR/autodock_vina_1_1_2_rev/autodock_vina_1_1_2/build/linux/release/vina
   save and quit vi

   ~> make

File Preparation

1. Changes have been made to a few ADT scripts in order to prepare the input file needed by VinaMPI and additional scripts written to facilitate the process. Download files and copy them to the ADT directories on your computer.

   ~> wget cmb.ornl.gov/~sek/predocking.tgz
   ~> tar xzvf predocking.tgz
   ~> mv MoleculePreparation2.py
   /Library/MGLTools/1.5.4/MGLToolsPckgs/AutodockTools/
   ~> mv prepare_ligand5.py
   /Library/MGLTools/1.5.4/MGLToolsPckgs/AutodockTools/Utilities24

   Make sure predockingflexv4.py correctly points to prepare_ligands5.py and the parentDirectory points to the directory containing your input .mol2 file and lines 30 and 31 have the correct string denoting a new molecule in the .mol2 file.

2. Starting with a .mol2 file containing all of the ligands for your screen call the predockingflexv4.py scripts. As a parameter it takes the name of the .mol2 file.

   ~> python predockingflexv4.py input.mol2

   This will create all the ligand input .pdbqt files for the screen and a file ligands.txt which contains all the ligand names, number of rotatable bonds, and number of atoms.

3. Call the preprocess_ligandsv3.py to sort the ligands.txt file and create the input file needed for VinaMPI (ligand_sort.txt).

   ~> python preprocess_ligandsv3.py

4. Make sure that prep_receptors.py points to prepare_receptor4.py (ADT scripts) and parentDirectory points to directory in which you want the receptor .pdbqt files to be created. Then call the script from within the directory containing the .pdb files for all of the receptors.

   ~> python prep_receptors.py

5. If doing ensemble docking where you are specifying the center of the docking box based on one atom in the protein and your different receptor files are different conformations of the same protein, you can use the make_rec_param.py script and give it the atom and size of box side in Å (if cubic) as parameters. If every receptor is a unique protein, the receptor input file must be created manually by the guidelines in the manuscript.

   ~> python make_rec_param.py VAL_281_CG1 30

6. Once all of the files are in place, the screen can be started with the following command

   aprun -n 84672 VinaMPI receptors.txt ligands_sort.txt 4 98164 receptors ligands

   where –n specifies the number of processors to use (flags would be changed here if using threading as well), VinaMPI is the path to the program, receptors.txt is the path to the receptor parameter file, ligands_sort.txt is the path to the ligand parameter file, 4 is the number of receptors in the screen, 98164 is the number of ligands in the screen, receptors is the directory containing all of the receptor .pdbqt input files, and ligands is the directory containing all of the ligand .pdbqt input files. The output files will be written in a directory named out created in the directory in which the program is called.