OptMAVEn 2.0

1. **Start\_Experiment.py**
   1. **EXPERIMENT.Experiment.user\_creation**
      1. Input antigen molecules and select residues to exclude
      2. Input CHARMM topology, parameter, and solvation files
      3. **EXPERIMENT.input\_validation**
         1. mkdir experiment[“Folder”]
         2. cd experiment[“Folder”]
         3. cp CHARMM topology and parameter files from OptMAVEn2.0/input\_files to cwd
         4. **EXPERIMENT.Experiment.appropriateness**
            1. Output antigen to experiment[“Folder”]/Molecule$Ag.pdb
            2. **VMD.generate\_PSF**

**make\_antigen\_psf.tcl**

Use Molecule$Ag.pdb and topology files in experiment[“Folder”] to add missing atoms to Molecule$Ag.pdb and generate Molecule$Ag.pdb and Molecule$Ag.psf in experiment[“Folder”]; return a MOLECULES.Molecule of Molecule$Ag.pdb with added coordinates, but do not transform any existing coordinates.

* + - 1. rm all files in cwd
      2. mkdir cwd/input\_files
      3. cd input\_files
      4. cp CHARMM topology and parameter files from OptMAVEn2.0/input\_files to cwd
      5. cd experiment[“Folder”]
      6. mkdir cwd/results
      7. mkdir cwd/structures
      8. cd structures
      9. Output antigen structure to structures/Molecule$Ag.pdb
    1. Input epitope positions and OptMAVEn grid
  1. cd experiment[“Folder”]
  2. Output experiment to Experiment\_Details.txt
  3. python OptMAVEn2.0/programs/OptMAVEn2.0.py

1. **Optmaven2.0.py**
   1. **OPTMAVEN2.initialize\_antigen**
      1. **VMD.generate\_PSFs**
         1. **make\_antigen\_psf.tcl**
            1. Use experiment[“Folder”]/structures/Molecule$Ag.pdb and experiment[“Folder”]/input\_files/$topology.rtf to generate experiment[“Folder”]/structures/Molecule$Ag.pdb and experiment[“Folder”]/structures/Molecule$Ag.psf
      2. **VMD.relax\_molecules**
         1. Use OptMAVEn2.0/input\_files/namd\_relaxation\_base.conf to generate a NAMD configuration file experiment[“Folder”]/input\_files/namd\_relaxation.conf for the relaxation that uses experiment[“Folder”]/structures/Molecule$Ag.pdb, experiment[“Folder”]/structures/Molecule$Ag.psf, experiment[“Folder”]/input\_files/$parameters.prm, and output prefix experiment[“Folder”]/structures/Molecule$Ag
         2. Run NAMD to relax the antigen and generate output files with the extensions .coor, .vel, .xsc, and .xst after the output prefix.
         3. mv experiment[“Folder”]/structures/Molecule$Ag.coor experiment[“Folder”]/structures/Molecule$Ag.pdb
         4. rm the .vel, .xsc, and .xst files
      3. **VMD.initial\_antigen\_positions**
         1. **initial\_antigen\_position.tcl**
            1. Use experiment[“Folder”]/structures/Molecule$Ag.pdb and experiment[“Folder”]/Experiment\_Details.txt to generate a file of the antigen with the epitope centered at the origin and pointing towards the negative z axis.
   2. **OPTMAVEN2.cull\_clashes**
      1. **VMD.cull\_clashes**
         1. **cull\_clashes.tcl**
            1. Use experiment[“Folder”]/structures/Molecule$Ag.pdb, OptMAVEn2.0/input\_files/MoleculeH.pdb, OptMAVEn2.0/input\_files/MoleculeK.pdb, and experiment[“Folder”]/Experiment\_Details.txt to generate a file experiment[“Folder”]/input\_files/positions.dat of the positions of the antigen that avoid framework clashes.
   3. **experiment.incr\_OptMAVEn\_status**
      1. Output “1” to experiment[“Folder”]/status
   4. **OPTMAVEN2.interaction\_energies**
      1. **VMD.MAPs\_interaction\_energies**
         1. mkdir experiment[“Folder”]/energies
         2. mkdir experiment[“Folder”]/energies/Molecule$Ag
         3. mkdir experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart
         4. qsub experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart/$MAPsPart.sh
2. **$MAPsPart.sh**
   1. cd experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart
   2. **VMD.prepare\_antigen\_part**
      1. **merge\_antigen\_part.tcl**
         1. Use the antigen experiment[“Folder”]/structures/Molecule$Ag.pdb and MAPs part OptMAVEn2.0/databases/MAPs/$partType/$MAPsPart.pdb to generate combined antigen-part coordinate and structure files experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart/Molecule$Ag.pdb and experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart/Molecule$Ag.psf, respectively.
   3. **VMD.MAPs\_interaction\_energy**
      1. **interaction\_energies.tcl**
         1. Use VMD’s NAMDEnergy plugin to calculate the interaction energy between the antigen and MAPs part whose coordinates and structure are given in experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart/Molecule$Ag.pdb and experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart/Molecule$Ag.psf, respectively, in all antigen positions given in experiment[“Folder”]/input\_files/positions.dat, also using the epitope given in experiment[“Folder”]/Experiment\_Details.txt and parameters given in OptMAVEn2.0/input\_files/$parameters.prm, and generating the file of interaction energies experiment[“Folder”]/energies/Molecule$Ag/$MAPsPart/energies.dat.
   4. cd experiment[“Folder”]
   5. python OptMAVEn2.0/programs/OptMAVEn2.0.py
3. **Optmaven2.0.py**
   1. **OPTMAVEN2.regroup\_energies**
      1. Read the positions from experiment[“Folder”]/input\_files/positions.dat
      2. For each molecule Molecule$Ag, for each MAPs part $part, read the energy for each position $pos from the file experiment[“Folder”]/energies/Molecule$Ag/$part/energies.dat, and then, for each position, write the interaction energy between the antigen and each part to a file called experiment[“Folder”]/energies/Molecule$Ag/$pos.dat
   2. **experiment.incr\_OptMAVEn\_status**
      1. Output “2” to experiment[“Folder”]/status
   3. **OPTMAVEN2.select\_all\_parts**
      1. For each position file experiment[“Folder”]/energies/Molecule$Ag/$pos.dat, create a script, experiment[“Folder”]/energies/Molecule$Ag/$pos.sh, that will select the optimal set of parts for that position.
      2. qsub experiment[“Folder”]/energies/Molecule$Ag/$pos.sh
4. **$pos.sh**
   1. python OptMAVEn2.0/modules/select\_parts.py experiment[“Folder”]/energies/Molecule$Ag/$pos.dat
   2. cd experiment[“Folder”]
   3. python OptMAVEn2.0/programs/Optmaven2.0.py
5. **select\_parts.py**
   1. Get the name of the file of all of the interaction energies at the position $pos from the first argument.
   2. Use OPTMAVEN2.select\_parts\_cplex to select the five most optimal sets of parts from among the parts and energies in $pos
   3. Output the five sets of parts to experiment[“Folder”]/energies/Molecule$Ag/parts\_$pos.csv
6. **Optmaven2.0.py**
   1. Combine all of the experiment[“Folder”]/energies/Molecule$Ag/parts\_$pos.csv files into a single file, experiment[“Folder”]/energies/Molecule$Ag/parts\_all.csv, that lists all of the positions and the five sets of parts chosen for each position.