How to dock molecules with softwares from Schrödinger Inc.

Schrodinger Inc. provides a package with different options for docking small molecules to proteins. I will try to describe them briefly here, without rephrasing what is explained in the original articles or in the Schrödinger documentation. If you install Maestro (see below) it comes with the full documentation and I will write here in which file to look. This documentation is well written and I have found in it a lot of the information I was looking for. The present tutorial was first written for people in the Chemistry and Chemical Biology Department at Harvard University, so I sometimes refer to information specific for this environment; however, most of the information is general.

The main software for docking is called Glide. 3 levels of precisions are available: HTVS (High-Throughput Virtual Screening), SP (Standard Precision), XP (Extra Precision). During the docking (which performs "a systematic search of the conformational, orientational, and positional space of the docked ligand"), a score is calculated which is similar to a binding free energy in kcal/mol. The three precisions differ by the scoring function and also by how they dock the ligands. The first one takes less than 1 second per structure, the second one around 30 seconds per structure and the last one roughly 5 minutes. When I am dealing with large libraries (more than 1 million structures), I start with the HTVS precision, take the 15% bests which is then used as input for the SP and then take the 2% bests of SP for docking with XP. The values of 15% and 2% have to be adapted depending on the library size and how long you are willing to wait. If the library is made of 10,000 structures or less I would directly use the XP procedure to avoid false-positive and false-negative results (with 10 jobs with 1000 structures for example, each job will take 5,000 minutes i.e. 3,5 days).

More accurate procedures are also available. They both use Glide, and also Jaguar for the first one and Prime for the second one:

- In the QPLD procedure (Quantum Mechanic Polarized Ligand Docking), each structure is docked (with XP or SP), then the charges of the ligand in the field of the protein are computed with QM (at a level of theory similar to B3LYP/6-31G*) and the ligand is re-docked. The poses and scores should thus be slightly better than with only XP. It takes ~1 hour per structure. I have found that the improvement is usually small, and I would not recommend spending a lot of time with it.
- In the IFD procedure (Induced Fit Docking), each structure is docked (with XP or SP). The van der Waals radii are then reduced and/or highly flexible side chains of the protein around the ligand are removed. The ligands are then re-docked: this allows having multiple poses. The protein structure around the ligand is then predicted by reorienting nearby side chains. Finally, the protein is minimized and the ligands are re-re-docked. This protocol allows taking into account the protein flexibility but is very long (at least 10 hours per structure). Another way to take into account the protein flexibility is to perform the docking in different grids made with different conformations of the same protein, and calculate the average score or keep only the lowest score.

For docking, two files are needed: a protein grid and a ligand library. Tools are provided by Schrödinger Inc. for preparing both. One important tool in the Schrödinger suite is Maestro. It can be downloaded as part as an Academic Campaign (http://www.schrodinger.com/freemaestro/) and is also useful for post-processing. However, for preparing the ligands or the protein other softwares from Schrödinger Inc. are needed, and they are only installed on Odyssey since they are commercial. Two options can be used:

- Log into Odyssey and type "*module load hpc/schrodinger-2014-3*" and then "*maestro*".

Maestro can then be used for preparing almost all the input files, and jobs can even be run from it. This procedure is very slow because Maestro is a complicated graphical interface, and all the information between your screen and Odyssey must be exchanged through internet.

- Install in your computer the Schrodinger suite (not the academic version but the full suite; choose the same version as the one you want to use in Odyssey). In your ~/.bashrc file, add: "export SCHROD_LICENSE_FILE=27003@rclic1.rc.fas.harvard.edu". Then turn on the Odyssey VPN. Every program can then be ran locally (using the resources of your computer), while using the licenses from Odyssey. This is very useful for preparing proteins for example.

Usually I use Maestro only to prepare the protein. For docking, I prefer to prepare a template for the input file (which is always a text file) and then modify it and run it with bash on Odyssey.

Before making the grid, the protein must be prepared (proteins from the Protein DataBank sometimes miss some atoms for example). The protein preparation workflow is explained briefly in the file *protein_preparation.pdf* from the Maestro directory and more details are in the *protein_prep.pdf* file. The idea of the workflow is: 1) load a structure; 2) preprocess it by adding hydrogens, missing atoms or residues; 3) correct the problems; 4) minimize the structure. Generating the receptor grid is also made from Maestro. How to do it is explained in the *glide_docking.pdf* file in the Glide directory (this sheet is very useful) and also in the *glide_user_manual.pdf* or *glide_quick_start.pdf* files (docs/glide directory in the Maestro installation folder). If the protein structure you have has already a ligand in the active site and you want to dock other ligands in the same place, the procedure is very easy (you just need to click in the ligand and choose your options). Otherwise you need to know the cartesian coordinates of the center of the active site. The grid output file is a .zip file. The full process for preparing the protein and then the grid takes no more than 10 minutes when you run Maestro locally.

Then, you need a ligand library. How to prepare it depends and what you want to dock and what you already have. Schrödinger Inc. provides a software called ligprep for preparing the ligands. It can read a lot of input file formats and convert them to 3D, add the missing hydrogens, adapt the ionization state and minimize it. Even if you have a library with 3D structures (downloaded from internet for example), it is recommended to use ligprep at least to optimize the structures with its own level of theory. Ligprep works roughly at the speed of one molecule per second. Ligprep can be used directly from Maestro (see *ligand_preparation.pdf* from the Ligprep directory). If you run it from bash, the file *ligprep_command.pdf* sums up the commands. You should also look at the *ligprep_user_manual.pdf* file (docs/ligprep directory in the Maestro installation folder). From one input structure, ligprep can produce several outputs which are different stereoisomers, tautomers, ionization states... How many structures are made is written at the end of the .log output file. You can decide how much you want to expand states; I mainly used the meta-options for ligprep (see the end of page 4 of *ligprep_command.pdf*) such as "*ligprep -WAIT -expand_itc -ismi File.smi -omae File.mae"*.

All the softwares from Schrödinger Inc. work with their own jobcontrol center which I don't really like. I use the *-WAIT* option which says to ligprep to wait the end of the job before running the next command in the bash script. If you don't do it, the job will be running on the Odyssey server but you won't see it with *sacct* and you will need to control your jobs with *jobcontrol -list*. *-expand_itc* is one of the meta-options. If you can afford it, I recommend to use it (it "aggressively expand states"). If your library is too big you can use *-vary_itc*. If you only want to optimize the structure, instead of using *-expand_itc* you can use *-R b*. *-ismi* says that the input file File.smi is in the SMILES format, and *-omae* that the output file File.mae is in MAE format (this format is the one mainly used by all the Schrödinger Inc. softwares). A general advice I have is to split all your jobs. It is better to run 20 ligprep jobs –each one with 1000 input structures– than one with 20,000 structures or 200 jobs with 10,000 structures than 20 with 100,000 structures. The reason is that errors occur, and restarting failed

jobs does not always work. If you split your input files, you can in the same submission file ask for different ligprep jobs; it works well. You can then merge the output files in a unique file and split it back if needed.

Once you have the grid and the ligands, you can start the docking. The easiest way of doing it is by directly using glide. The command is: *glide -WAIT InputFile.in*. Again, I use the *-WAIT* option. Glide only needs an input file which looks like this:

```
EPIK_PENALTIES NO
USECOMPMAE YES
WRITEREPT YES
NREPORT 15000
RINGCONFCUT 2.500000
PRECISION HTVS
GRIDFILE glide-grid.zip
LIGANDFILE File.mae
```

More options are available, and details of these options can be found in the files *glide_user_manual.pdf* or *glide_quick_start.pdf* (docs/glide directory in the Maestro installation folder). This input file can be prepared from Maestro (Applications/Glide/Ligand Docking...) where you can choose your option, and then click on "Write". You can find the explanations of all keywords in the *glide_user_manual.pdf* file, on p104. For GRIDFILE and LIGANDFILE, you can give absolute or relative path. You will have several output files, the interesting ones are *InputFile_pv.maegz* and *InputFile.rept*. The first one can be opened by Maestro to see the poses of the ligands. It is in fact a .mae file which is zipped. So if you want to read it, you can rename it to *InputFile_pv.mae.gz* and then gunzip it. The .rept file is a text file which is written only if you have asked WRITEREPT YES in the input file and summarizes the results with the scores. If you have splitted your docking jobs and want to merge them, you can use the following command in which the 20 best poses will be written in NewFile.rept and NewFile_pv.maegz:
$SCHRODINGER/utilities/glide_merge -r NewFile.rept -o NewFile_pv.maegz -n 20 OldFile_pv.maegz

If you don't have a lot of ligands, you can use the precedent procedure directly with the XP precision. In such cases, the option WRITE_XP_DESC YES is interesting. However if you have a lot of ligands, you will use first HTVS, then SP, then XP (or only SP then XP). The workflow will be slightly different because you want to be able to extract the best HTVS poses to use them as input for the SP passage. The output file _pv.maegz can theoretically be directly used as an input file for the next step (the structure of the file is the same, it is a .mae file). However this is not recommended by Schrödinger Inc.: the ligand geometries in the output file are different from the one in the input file and are thus not optimized. The torsions, bond-angles, bond-lengths are different and this can lead to errors (see http://www.schrodinger.com/kb/1034). So what you want is to extract some specific ligands from the input file. This can be done in two ways.

- You can do it manually if the compounds are identified by unique titles. If the titles are not always different, there is a utility called unique_names to solve this. I highly recommend that you check that all your input ligands have a different name before starting the docking. The first time I worked with Glide I didn't check that and it lead to a lot of troubles later. The problem of names occurs mainly when ligprep produces different isomers from the same input file: they can all have the same name. In the following, *(...)* has to be replaced by either *$SCHRODINGER* or */n/sw/schrodinger-2014-3/*
  1) Create a file with the 1000 ligands (e.g.) you want to keep from HTVS_pv.maegz:
     (...)/utilities/maesubset -n 1:1001 HTVS_pv.maegz > Best-HTVS_pv.mae

2) Extract the titles from these 1000 docking results and write them in a file titles.txt:
(...)/utilities/proplister -p title Best-HTVS_pv.maegz > titles.txt
3) Extract from the input file the ligands whose names are in the file titles.txt:
(...)/utilities/maesubset -t titles.txt HTVS-Input.maegz > Best-HTVS_Original.mae

- The other possibility is to use the Virtual Screening Workflow (VSW). This procedure allows doing everything with the same job. With a unique input, you can do in a row the docking with HTVS, extracting the best poses, docking with SP, extracting the best poses and docking with XP. The command is *vsw -WAIT File.inp*. Input files can be made directly from Maestro but some options will have to be written manually. The VSW is very powerful, but it can be tricky to use (see *vsw_user_manual.pdf* from the docs/vsw directory in the Maestro installation folder). I didn't use it this way (HTVS-SP-XP), but it is useful for docking and then extracting at a given precision.

As I have already mentioned, other workflows are possible such as QPLD or IFD. Both types of inputs are similar to the VSW input files. I won't describe them here. See the documentation *qpld_user_manual.pdf* and *inducedfit_user_manual.pdf* (from the docs/qpld and docs/inducedfit directories in the Maestro installation folder). Since these procedures are more complicated, they are well explained in the Schrödinger documentation. The commands for each procedure are *qpld -WAIT File.inp* and *ifd -SUBHOST ${HOSTNAME} -NGLIDECPU 1 -NPRIMECPU 1 -WAIT File.inp*.

The Schrödinger suite of softwares is expensive, and Harvard has only 25 licenses. If you want to know how many licenses are still available, you can do *lsload -l > list.txt* and then open the *list.txt* file (be sure to disable text wrapping). You will then be able to see how many licenses are still available for each software. When you submit a job, if no more licenses are available it will directly fail. There should be a control system which first checks if some licenses are available, and if not waits for new licenses to be available; however it has never worked for me. I thus use the trick below: in my bash script I make a loop which checks if the output file has been written. As long as there is no output, you try to run glide after waiting some time (5 minutes, e.g.):

```
while [ ! -s ${File}.rept ]; do
        glide -WAIT ${File}.in
        sleep 300s
done
```