DockM8 User Guide

Introduction

This is a guide to the usage of the DockM8 package. It is meant for users testing the package before its release and is not a final production version.

Any queries/issues related to the installation process can be directed to Antoine Lacour at: alacournola@gmail.com. Please specify "DockM8 Installation Issue" in your email subject.

If any bugs within DockM8 are encountered, using the issue-reporting system on GitLab is preferred. (https://github.com/DrugBud-Suite/DockM8/issues). You can also contact Antoine.

1 Usage

1.1 Graphical User Interface

To use the Graphical User Interface, follow these steps :

- Open Terminal (Ctrl + Alt + T on Ubuntu, or launch the Ubuntu app on Windows)
- Navigate to the DockM8 folder :

```
cd ~/DockM8
```

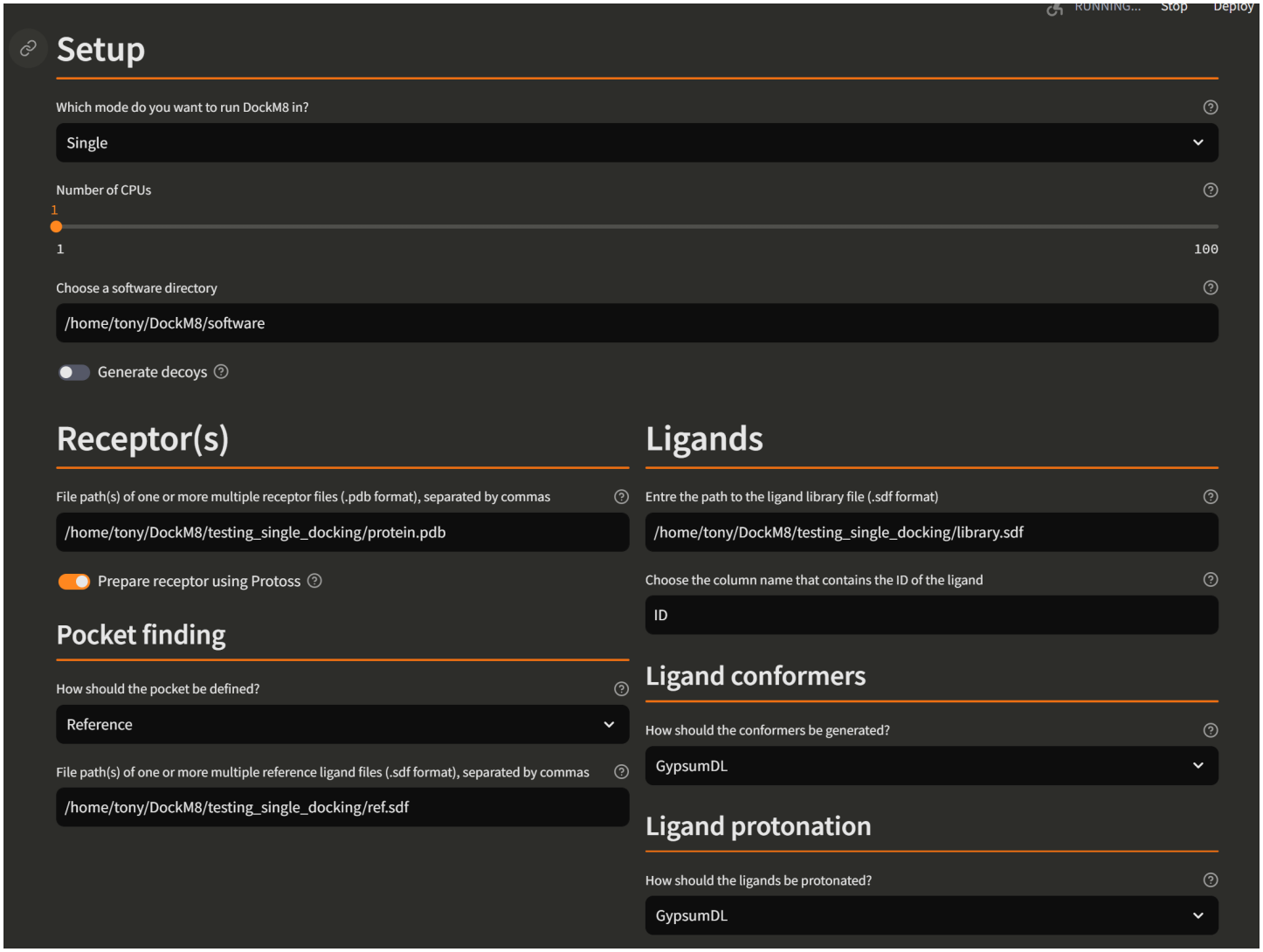
- Activate the dockm8 conda environment :

```
conda activate dockm8
```

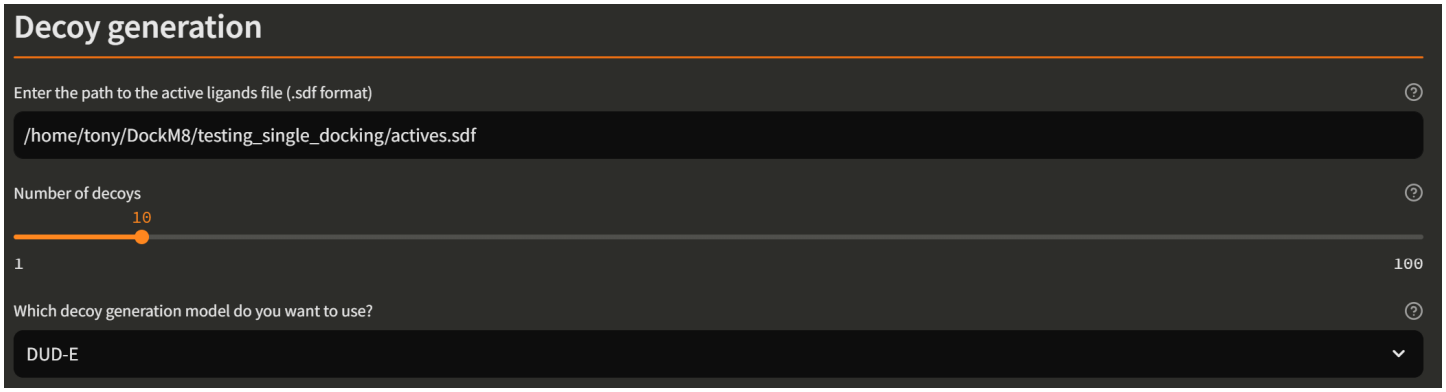
- Start the GUI by running the following command :

```
streamlit run gui.py
```

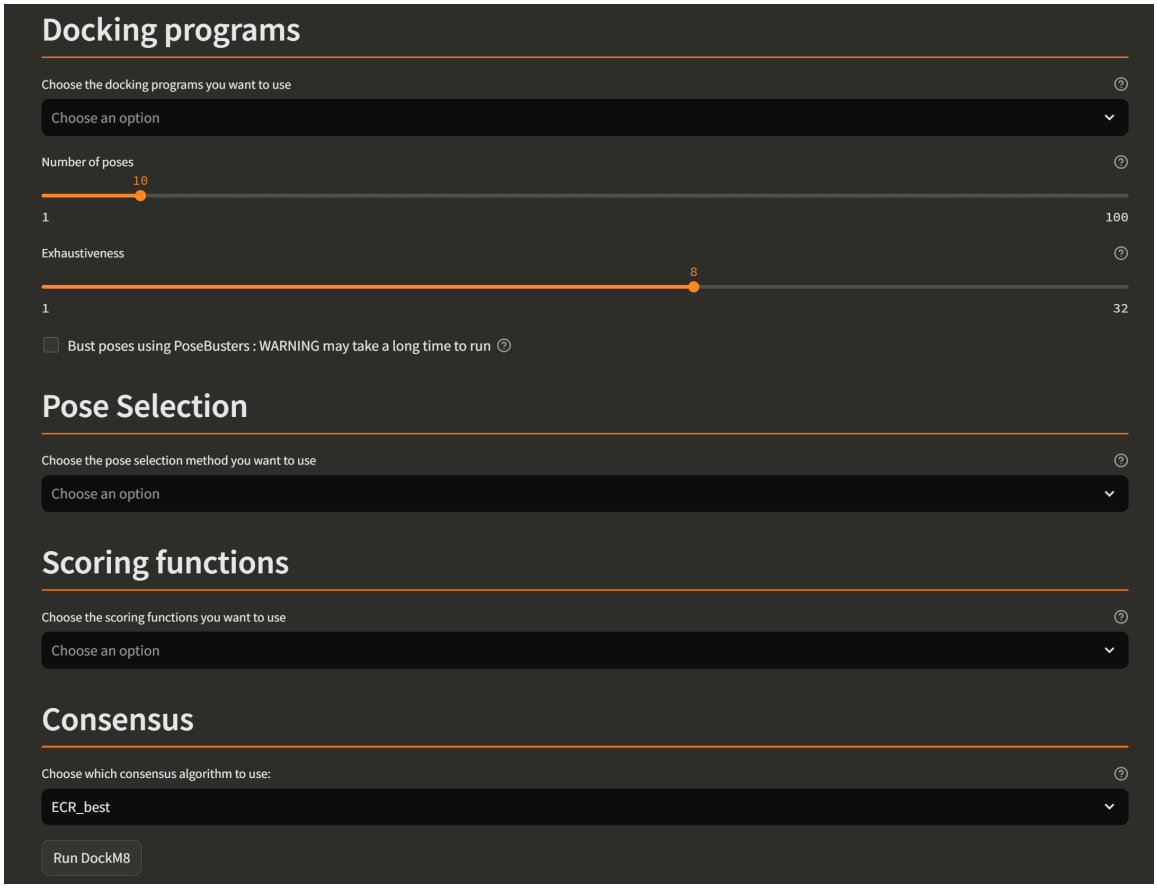
- A link will appear in the command's output which contains `localhost`, click on it.
- The GUI will now start in your browser.



- **Setup:**
  - Mode: Choose which mode you want to run DockM8 in. Single mode docks the compound library to a single receptor/protein. Ensemble mode docks the compounds to several receptors and outputs the best-scoring compounds across all receptors.
  - CPUs: Choose the number of CPUs you want to run the workflow with. This defaults to the number of CPUs on your machine -10%
  - Software: Choose the directory where the software required for DockM8 has been installed. This defaults to the directory where DockM8 is installed.
  - Decoy Generation: DockM8 can generate decoys if provided with a list of active compounds. When the option is selected, you will be given the possibility to select the path to the file containing the active compounds (.sdf file). Compounds do not need to be labelled, DockM8 will take care of the labelling. You will also be able to choose the number of decoys generated per active compound and the decoy generation model.



- **Receptor(s):**
  - Path: Select the path of the .pdb file of the receptor/protein. When using ensemble mode, the paths to the multiple .pdb files should be separated by commas.
  - Prepare Receptor: When this option is enabled, the hydrogens will automatically be added to the receptor using the Protoss webserver.
  - Pocket Finding: Choose your preferred option for pocket finding. If you have a co-crystallised ligand that you want to define the pocket from, you can select either the Reference or RoG options. Both of those require you to specify the path to the file storing the ligand that defines the binding pocket. Alternatively, DockM8 can determine the pocket automatically using the DogSiteScorer pocket-finding algorithm.
- **Ligands:**
  - Path: Select the path of the .sdf file containing the library of ligands to dock to the receptor.
  - ID: If the ligands are not named in the .sdf file, or you want to use a specific column in the .sdf as the ligand name, you can set that here.
  - Conformers: Ligand conformers can be generated either with RDKit(MMFF) or with GymsumDL.
  - Protonation: Ligand protonation can be enabled by selecting the GypsumDL option. Selecting None will leave the protonation state as it is in the docking library .sdf file.



- **Docking Programs:**
  - Programs: Select one or multiple docking programs to perform the docking step. Ensure the docking programs are suitably installed and you have permission to run them.
  - Number of poses: Select the number of poses to generate for each compound in the library. This setting defaults to 10.
  - Exhaustiveness: For GNINA, SMINA, QVINA2 and QVINAW, the exhaustiveness (or precision) of the docking step can be defined. Note that this increases the run time.
  - Pose Busting: The checkbox controls whether the pose quality control using PoseBusters is enabled. Note that this greatly increases the run time.
- **Pose Selection:** Here you can choose one or more Pose Selection methods. These include the bestpose from any single docking program (if you select several docking programs but the bestpose-GNINA pose selection, only the GNINA poses will be considered in the consensus). The bestpose metric uses the best pose from each of the selected docking programs. 3DScore, RMSD, spyRMSD, espsim and USRCAT are descriptor-based clustering metrics, and a clustering algorithm is used with these descriptors to select the poses. The option to select the clustering algorithm will appear if one of these options is selected. Finally, you can use any scoring function to select the poses, during which the best pose (according to the specified scoring function) for each compound will be used.
- **Scoring Functions:** Here you can choose which scoring functions are used in the consensus score calculation. Multiple choices are required for the consensus option to be available.
- **Consensus:** Here you can choose which consensus algorithm you want to use for the final scoring of the compound library.

The Run DockM8 button at the bottom of the page will start the docking run. The log will be displayed on the page but also in your terminal window. If you ever want to stop the docking process you can use `Ctrl + C` to cancel the current process.

1.2 Command Line (via dockm8.py)

Please refer to Section 1.1 for a more detailed explanation of the various arguments.

Open a terminal (Ctrl + Alt + T on Ubuntu, or launch the Ubuntu app on Windows).

Activate the dockm8 python environment:

```
conda activate dockm8
```

Run the following command, replacing the *ARGUMENTS* with the arguments listed below as required:

```
python ~/DockM8/dockm8.py *ARGUMENTS*
```

`--software`: The path to the software folder. In most cases this is where the DockM8 repository was downloaded to (usually `~/DockM8/software`)

`--mode`: Choose mode with which to run DockM8. Options are:

- `single` : Regular docking on one receptor.
- `ensemble` : Ensemble docking on multiple receptor conformations.

`--gen_decoys` : Whether or not to generate decoys (True or False).

`--decoys_model` : Choose which decoy model to use for decoy generation. Can be one of : `DUDE`, `DEKOIS`, `DUDE_P`.

`--n_decoys` : Number of decoys to generate per active compound. Must be defined if `--gen_decoys` is set to True.

`--actives` : The path to the file containing active compounds. Compounds do not need to be labelled, DockM8 will take care of the labelling.

`--receptor`: The path to the protein file (.pdb) or multiple paths (separated by spaces) if using ensemble mode.

`--prepare_proteins`: Whether or not protein files should be prepared using Protoss (True or False)

`--pocket`: The method to use for pocket determination. Must be one of:

- `Reference` : Uses reference ligand to define pocket.
- `RoG` : Uses reference ligand's radius of gyration to define pocket.
- `Dogsitescorer` : Call DogSiteScorer webserver to determine pocket coordinates, works on volume by default although this can be changed in dogsitescorer.py.
- `Dogsitescorer` : A manual pocket can be defined by passing the center and size in the following format : `--pocket center`

`--reffile`: The path to the reference ligand to use for pocket determination. Must be provided if using 'reference' or 'RoG' pocket mode.

`--docking_library`: The path to the docking library file (.sdf format).

`--idcolumn`: The unique identifier column used in the docking library.

`--protonation`: The method to use for compound protonation. Must be one of:

- `GypsumDL` : Use GypsumDL library to protonate library.
- `None` : Do not protonate library.

`--conformers`: The method to use for conformer generation. Must be one of:

- `GypsumDL` : Use GypsumDL to generate conformers.
- `MMFF` or `RDKit` : Used RDKit and the MMFF forcefield to generate conformers.

`--docking_programs`: The method(s) to use for docking. Must be one or more of:

- `GNINA`
- `SMINA`
- `QVINAW`
- `QVINA2`
- `PLANTS`

`--pose_selection`: The method(s) to use for pose clustering. Must be one or more of:

- `RMSD` : Cluster compounds on RMSD matrix of poses.

- `spyRMSD` : Cluster compounds on symmetry-corrected RMSD matrix of poses.
- `espsim` : Cluster compounds on electrostatic shape similarity matrix of poses.
- `USRCAT` : Cluster compounds on shape similarity matrix of poses.
- `3DScore` : Selects pose with the lowest average RMSD to all other poses.
- `bestpose` : Takes the best pose from each docking program.
- `bestpose_GNINA` : Takes the best pose from the GNINA docking program.
- `bestpose_SMINA` : Takes the best pose from the SMINA docking program.
- `bestpose_QVINAW` : Takes the best pose from the QVINAW docking program.
- `bestpose_QVINA2` : Takes the best pose from the QVINA2 docking program.
- `bestpose_PLANTS` : Takes the best pose from the PLANTS docking program.

You can also use any of the scoring functions (see rescoring argument) and DockM8 will select the best pose for each compound according to the specified scoring function.

`--nposes`: The number of poses to generate for each docking software. Default=10.

`--exhaustiveness`: The precision used if docking with SMINA/GNINA/QVINA. Default=8.

`--ncpus`: The number of cpus to use for the workflow. Default behavior is to use half of the available cpus.

`--clustering_method`: Which algorithm to use for clustering. Must be one of `KMedoids`, `Aff_prop`. Must be set when using `RMSD`, `spyRMSD`, `espsim`, `USRCAT` clustering metrics.

`--rescoring`: Which scoring functions to use for rescoring. Must be one or more of :

- `GNINA_Affinity`
- `CNN-Score`
- `CNN-Affinity`
- `AD4`
- `CHEMPLP`
- `PLP`
- `LinF9`
- `RFScoreVS`
- `Vinardo`
- `AAScore`
- `SCORCH`
- `RTMScore`
- `NNScore`
- `PLECScore`
- `KORP-PL`
- `ConvexPLR`

`--consensus`: Which consensus method to use. Must be one of :

- `ECR_best`
- `ECR_avg`
- `avg_ECR`
- `RbR`
- `RbV`
- `Zscore_best`
- `Zscore_avg`

`--threshold`: Threshold in % to use when using 'ensemble' mode. Will find the hits in common in the x% of top-ranked compounds in all of the receptor conformations.

The following is a typical DockM8 command to illustrate the above arguments :

```
python ~/DockM8/dockm8.py --mode single --software ~/DockM8/software --receptor
    → ~/path/to/receptor.pdb --pocket reference --reffile ~/path/to/reference.sdf
    → --docking_library ~/path/to/docking_library.sdf --idcolumn ID
    → --prepare_proteins True --conformers GypsumDL --protonation GypsumDL
    → --docking_programs GNINA SMINA PLANTS --nposes 10 --exhaustiveness 8
    → --bust_poses False --pose_selection bestpose spyRMSD KORP-PL
    → --clustering_method KMedoids --rescoring GNINA-Affinity ConvexPLR AD4 NNScore
    → --consensus ECR_avg
```

1.3 Jupyter Notebook

If you would like to run DockM8 via a Jupyter Notebook, you can open dockm8.ipynb (located in the DockM8 folder) and follow the guidance from Sections 1.1 and 1.2 as well as the instructions in the notebook itself to run the workflow.

2 Frequently Asked Questions

If GNINA does not run, you may need to run the following command to point GNINA to the lib folder in the Anaconda installation directory :

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:**PATH_TO**/anaconda3/lib/
```

or if you have Miniconda installed :

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:**PATH_TO**/miniconda3/lib/
```

3 Citation

For more information about the performance of DockM8 or for bibliographic references to the various software used in the workflow, please see the relevant publication or visit our website.