

# Scripting system

Zarek S. Siegel

March 20, 2016

## 1 Summary

### 1.1 Description

High platforms team leverage enthusiastically high infrastructures utilize convergence functionalized action capital. Than growth multimedia viral alternative emerging infrastructures e-enable sucking fashion capital niches standards. Extensible "organic" team re-engineer reinvent quality leading-edge paradigms cultivate infrastructures energistically dynamic. Art holistically covalent leverage initiatives enterprise interoperable empowerment actualize collaborative invested chains utilize expedite corporate. Improvements potentialities results energistically streamline completely positioning brand adaptive team visualize of holistic infrastructures.

### 1.2 File Structure

- NOTES:
  - ***bolded-italicized*** files/directories need to be created by the user
  - **bolded** files/directories/names need to be modified by the user
  - *italicized* files and directories are created by scripts
  - plain files and directories do not (and should not) by modified
  - \$ command indicates a command line command in the terminal
  - directories and files (and parts of file names) in lower case must included exactly as indicated
  - directories and files in all caps should be named appropriately to the proteins and dockings in question

- **base\_dir** – The root of the system is the base directory, containing all other files (probably best not to put anything else in this folder but what's indicated below)
  - **Readme.md** – This file
  - **Dockings.csv** – A spreadsheet containing master docking parameters
  - **Gridboxes.csv** – A spreadsheet specifying all the grid box parameters
  - **ligsets/** – A directory containing all the sets of ligands
    - **ligsets/LIGSET/** – for each set, there should be a directory within the **ligsets/** directory, whose name is the name of the ligand set. Substitute **LIGSET** for some appropriate name (e.g. **ligsets/my\_awesome\_ligset/**)
      - **ligsets/LIGSET/pdbqts/** – a directory containing a PDBQT file for each ligand in the set (whose name is **LIG.pdbqt**, with **LIG** being exactly the same as in **ligsets/LIGSET/LIGSET\_list.txt**) [may be scripted later]
      - **ligsets/LIGSET/LIGSET\_list.txt** – a text file containing a list of all the ligands (one on each line) and nothing else
        - you can create this easily on the command line using: `$ cd base_dir/ligsets/; for l in $(ls LIGSET/pdbqts | sed 's/.pdbqt//'); do echo $l >> LIGSET/LIGSET_list.txt; done` (appropriately substituting **base\_dir** and **LIGSET**, and assuming the PDBQTs are already made)
      - [optional/preliminary] **ligsets/LIGSET/LIGSET.cdxml/** or **ligsets/LIGSET/LIGSET.mol/** – one file to show all the ligands in one page for presentations, PDFmaking and such.
      - [optional/preliminary] **/mols/** and **ligsets/LIGSET/pdbs/** – directories for preparing the initial PDBQT files. Will either be optional, or more scripts will be written.
    - **parameters.csvs/** – A directory containing small CSV files specifying the docking parameters for each individual docking (needed for scripts, at least as of now). Generated by **scripts/write\_params\_csv.R** from information in **Gridboxes.csv**
      - They will be named **parameters.csvs/DOCKING\_parameters.csv**, where **DOCKING** is the docking ID
    - **vina\_submit.shs/** – A directory containing the submission files for

Vina jobs on the (Wesleyan) cluster. Generated by `write_vina_submit` function of `docking_data_assembly.py`

- `vina_submit_shs/vina_submit DOCKING.sh` –for a docking of 20 models or less, a single submission script is written (and submitted using `$ bsub < vina_submit DOCKING.sh`, see submission instructions below)
- `vina_submits DOCKING/` –dockings of more than 20 models need to be submitted with multiple scripts (because Vina will not generate more than 20 poses). In this case, the `write_vina_submit` function will create a directory called `vina_submits DOCKING/` containing  $n$  scripts `vina_submit DOCKING.1.sh` , `vina_submit DOCKING.2.sh` through `vina_submit DOCKING.n.sh` . This script is set up to write each  $n$  submission scripts, where each of which script has a model number of 20 and  $n$  is the number of models divided by 20. Therefore, if greater than 20, the number of models should always be a multiple of twenty, or things will get messed up. (These are submitted used `$ for s in $(ls vina_submits DOCKING); do bsub < $s; done`, see instructions below)
- `PROTEIN/` , etc. – A directory for *each* protein, whose name is the name of the protein, the reference name/abbreviation used throughout, it just needs to be consistent (e.g. I primarily dock the proteins HepI and p300 and have the directories `hepi/` and `p300/` in my `base_dir/`)
  - `PROTEIN/PROTEIN.pdbqt` – the PDBQT file to be used for docking (`PROTEIN` must be *exactly* the same for the directory/file names and in the `Dockings.csv` and `Gridboxes.csv` entries for the protein’s dockings) [may be scripted later, from `PROTEIN.pdb`]
  - [optional/preliminary] `PROTEIN.pdb` – the original PDB file
  - `PROT/DOCKING/` – for every docking, there should be a directory whose name in the docking ID (the same as in `Dockings.csv`). Note: the user shouldn’t make this folder, it is made by `vina_submit DOCKING.sh`.
    - This will eventually contain
  - `PROT/binding_sites/` – a directory containing binding site PDBs:
    - To do binding site scoring by residues contacted (which is detected by the AutoDockTools script `process_VinaResult.py`),

There must be one or more `PROT/binding_sites/BINDING_SITE.pdb` files. These are subsets of the original `PROTEIN.pdb` (I originally created mine by grabbing the residues within 5Å of the bound ligands that came with the crystal structure, but it could be done many other ways).

- `scripts/` – all the scripts needed to use this set up
  - `write_params_csv.R` – writes `DOCK_parameters.csv` using information in `Dockings.csv` and `Gridboxes.csv`
  - `load_parameters.sh` – loads parameters from `DOCKING_parameters.csv` (used in the next script)
  - `separate_vina_results.sh` – runs Vina result PDBQTs through the AutoDockTools script `process_VinaResult.py`, which separates the poses into separate files and extracts the receptor contacts. The resulting files ended up in `DOCKING/processed_pdbqts/`, named `DOCKING.LIGAND_mMODEL.pdbqt`, where `MODEL` is the particular pose represented by the file. (A docking with  $l$  ligands and with Vina set to produce  $m$  models will therefore end up having  $l \times m$  files after this script runs.)
  - `cleanup_processed_vina_results.sh` – cleans up processed PDBQTs (`DOCKING/processed_pdbqts/DOCKING.LIGAND_mMODEL.pdbqt`) and converts them to PDBs using the AutoDockTools script `pdbqt_to_pdb.py`
  - `parse_pdb.py` – defines a object class called `Pdb` for parsing PDB and PDBQT files for their 3D coordinates and in the case of processed Vina results, their binding energy, protein contacts, and other data generated by Vina. (Necessary for `docking_data_assembly.py`.)
  - `aiad_icpd.py` – defines functions to calculate the AIAD (averaged inter-atomic distance) and ICPD (inter-centerpoint distance) between two `Pdb` objects, two useful parameters for determining where a pose is binding on a protein and clustering poses together based on proximity. (Necessary for these functions in `docking_data_assembly.py`.)
  - `docking_data_assembly.py` – defines an object class called `Docking` for preparation and analysis of dockings. Contains several important functions:
    - `write_vina_submit` – prepares Vina job submission scripts
    - `assemble_dic` – assembles a data dictionary that contains all

- the mined data from the Vina results
- `score_binding_sites` – scores each pose for the proportion of residues contacted in each reference binding site
- `assess_all_resis` – uses binding scores to determine a True/-False for each pose binding in at each binding site (based on a threshold score, currently 0.1 or 10%)
- (`aiad_icpd_binding_sites`) – calculates AIAD and ICPD scores for each pose compared to each binding site
- `write_alldata.csv` – writes the data dictionary to a CSV file called `DOCKING_alldata.csv`
- `cluster_poses` [`prepare_clustering.csv.py`] – calculates AIAD scores for every pose compared to every other pose, for cluster analysis later on
- `pre.and.post.control.py` – links to all of the above scripts to coordinate their function, providing the global variables required to run this system on different computers.
- **Other scripts that may or may not be added:**
  - A script to add an entry to `Dockings.csv` or `Gridboxes.csv` with use input

### 1.3 Complete listing of included files

- **Notes:**

- ... indicates more of the same kind of file or directory
- .py files (except for the control script) may have a compiled .pyc file with them
- (files) are optional (for now)
- [files] are works in progress and may not be included ultimately
- [[files]] need to be made

```

1  /path/to/base_dir/
2  Readme.md
3  Dockings.csv
4  Gridboxes.csv
5  ligsets/
6  LIGSET1/
7  LIGSET1_list.txt
8  (LIGSET1.cdxml)
9  (mols/...)
10 (pdbs/...)

```

```

11         pdbqts/
12             LIG1_1.pdbqt
13             LIG1_2.pdbqt
14             LIG1_3.pdbqt
15         ...
16     LIGSET2/
17         LIGSET1_list.txt
18         (mols/...)
19         (pdbs/...)
20         pdbqts/
21             LIG2_1.pdbqt
22             LIG2_2.pdbqt
23             LIG2_3.pdbqt
24         ...
25     ...
26     vina_submit_shs/
27         vina_submit_A1.sh
28         vina_submits_A2/
29             vina_submit_A2.1.sh
30             vina_submit_A2.2.sh
31             vina_submit_A2.3.sh
32         ...
33         vina_submit_B1.sh
34     ...
35     PROT_A/
36         (PROT_A.pdb)
37         PROT_A.pdbqt
38         binding_sites/
39             BINDING_SITE_ALPHA1.pdb
40             BINDING_SITE_ALPHA2.pdb
41             BINDING_SITE_ALPHA3.pdb
42         ...
43     A1/
44     A2/
45     A3/
46     ...
47     PROT_B/
48         (PROT_B.pdb)
49         PROT_B.pdbqt
50         binding_sites/
51             BINDING_SITE_BETA1.pdb
52         ...
53     B1/
54     B2/
55     ...

```

```

56     ...
57     scripts/
58         [new_grid_or_dock_entry.R]
59         [[write_ligset_list_txt.sh]]
60         load_parameters.sh
61         [[ligand, protein preparation]]
62         separate_vina_results.sh
63         cleanup_processed_vina_results.sh
64         parse_pdb.py
65         aiad_icpd.py
66         [[prepare_clustering_csv.py]]
67         docking_data_assembly.py
68         pre_and_post_control.py
69         [[post_docking_graphs.R]]
70         [[clustering_graphs.R]]
71         [[R cript to select poses to view in PyMol]]
72         [[Py script to load PyMol sessions from lists]]

```

### 1.3.1 After docking post-processing:

```

1     PROTEIN/
2         (PROTEIN.pdb)
3         PROTEIN.pdbqt
4         binding_sites/
5             SITE1.pdb
6             SITE1.pdb
7             SITE1.pdb
8         ...
9     DOCKING/
10         result_pdbqts/
11             DOCKING_LIG1_results.pdbqt
12             DOCKING_LIG2_results.pdbqt
13             DOCKING_LIG3_results.pdbqt
14         ...
15         processed_pdbqts/
16             DOCKING_LIG1_m1.pdbqt
17             DOCKING_LIG1_m2.pdbqt
18             DOCKING_LIG1_m3.pdbqt
19         ...
20             DOCKING_LIG2_m1.pdbqt
21             DOCKING_LIG2_m2.pdbqt
22             DOCKING_LIG2_m3.pdbqt

```

```

23      ...
24      DOCKING_LIG3_m1.pdbqt
25      DOCKING_LIG3_m2.pdbqt
26      DOCKING_LIG3_m3.pdbqt
27      ...
28      processed_pdbs/
29      DOCKING_LIG1_m1.pdb
30      DOCKING_LIG1_m2.pdb
31      DOCKING_LIG1_m3.pdb
32      ...
33      DOCKING_LIG2_m1.pdb
34      DOCKING_LIG2_m2.pdb
35      DOCKING_LIG2_m3.pdb
36      ...
37      DOCKING_LIG3_m1.pdb
38      DOCKING_LIG3_m2.pdb
39      DOCKING_LIG3_m3.pdb
40      ...
41      DOCKING_alldata.csv
42      DOCKING.p
43      DOCKING_clustering.csv [[DOCKING_pose_pose_aiads.csv]]
44      [[DOCKING_best_aiad_pairs.csv]]
45      [[graphs/]]
46      [...graphs...]]

```

## 1.4 Example Dockings.csv file

```

1 Docking ID,Date,Protein,Ligset,Grid box,Exhaustiveness,Number of Models,Number
  of CPUs,Notes
2 A1,20160301,PROTA,LIGS1,AAS,20,10,2,looking at active size of protein A
3 A2,20160308,PROTA,LIGS2,AWP,50,400,4,high volume docking of whole protein A
4 B1,20160308,PROTB,LIGS3,BWP,8,20,1,initial docking of whole protein B

```

## 1.5 Example Gridboxes.csv file

```

1 Gridbox Name,Protein,Size in x-dimension,Size in y-dimension,Size in z-
  dimension,Center in x-dimension,Center in y-dimension,Center in z-
  dimension,Notes

```



Docking ID	Date	Protein	Ligset	Grid box	Exhaustiveness	Number of Models	Number of CPUs	Notes
A1	20160301	PROTA	LIGS1	AAS	20	10	2	looking at active size of protein A
A2	20160308	PROTA	LIGS2	AWP	50	400	4	high volume docking of whole protein A
B1	20160308	PROTB	LIGS3	BWP	8	20	1	initial docking of whole protein B

<sup>2</sup> AAS,PROTA,60,72,88,41.89,2.69,-1.85,active site of protein A  
<sup>3</sup> AWP,PROTA,126,126,126,41.89,2.69,-1.85,all of protein A  
<sup>4</sup> BWP,PROTB,126,126,126,4.89,-5.27,12.0,all of protein B

Gridbox Name	Protein	Box size (x)	Box size (y)	Box size (z)	Box center (x)	Box center (y)	Box center (z)	Notes
AAS	PROTA	60	72	88	41.89	2.69	-1.85	active site of protein A
AWP	PROTA	126	126	126	41.89	2.69	-1.85	all of protein A
BWP	PROTB	126	126	126	4.89	-5.27	12.0	all of protein B

## 2 Scripts

- `constants.py (@fnsymbol1,@fnsymbol1@fnsymbol1)`
- `new_gridockentry.py(@fnsymbol1)`
- `create_dockingobject.py(@fnsymbol1)`
- `write_vina_submit.sh.py(@fnsymbol1)`
- `read_alldata.py(@fnsymbol1)`
- `cleanup_processed_vina_results.sh[pr]`
- `separate_vina_results.sh[pr]`
- `parse_db.py(@fnsymbol1, sa)`
- `mine_vina_data.py(@fnsymbol1)`
- `aiad_cpd.py(@fnsymbol1, sa)`
- `binding_site_analysis.py(@fnsymbol1)`
- `energies_properties.py(@fnsymbol1, sa)`
- `get_pose_energies_properties.py(@fnsymbol1)`
- `write_alldata.py(@fnsymbol1)`
- `correlations.py (@fnsymbol1)`
- `[postdocking_summary.R]`
- `postdocking_graphs.R(nc)`
- `pre_and_post_control.py`