Report

Diem-Trang Pham

Summer 2017

This report summarizes my work during the summer.

1 Requirements

- Python 2.7
- Libraries: Biopython 1.69, matplotlib 2.0.2

For simplicity, you can use Anaconda (https://docs.continuum.io/) to install required libraries.

2 Model generator

2.1 Interaction filter: strip.py

strip.py is a python version of strip.pl and has similar input parameters.

Input: sif file, format-specific list of amino acid residues (comma separated, no space)
Output: csv file or txt file which contains models generated from all possible combinations of interactions
Example:

./strip.py pdb1pwc_h.sif A:400:_:PNM
or
./strip.py pdb1pwc_h.sif A:400:_:PNM,A:62:_:SER,A:298:_:HIS

Output can be csv file or txt file. The only difference between these two output file formats is that CSV is easier to read by human, but it will be more difficult for the computer program to read as an input.

Output file format can be change in function Strip().

2.2 Models analysis: analyze_models.py

This script takes output txt file from strip.py as input. It cannot take csv file.
This script analyzes all output models from strip.py

Input: output txt file from strip.py
Output: Output includes figures and files:

- Figures:
  - all_sets.png: Count for number of models by number of interactions.
  - neighbor.png: Residue frequency in all models.
  - residue_in_unique_sets.png: Residue frequency in unique models.
  - residue_in_unique_sets_sorted.png: Residue frequency in unique models, sorted in descending order of frequency.
  - unique_sets_freq.png: Frequency of unique models.
- Files:
  - id_list.csv: list of all residue IDs and names.
  - edges_by_unique_sets.txt: groups of interactions by each unique models.
  - unique_sets_freq.txt: frequency of unique models, ordered by frequency.

Example:

./analyze_models.py pdb1pwc_h-400_py.txt

2.3 PDB generator from unique models: PDBgenerator.py

This script takes unique_sets_freq.txt file from analyze_models.py as an input and generates PDB files using unique models.

Input: output unique_sets_freq.txt file file from strip.py, PDB_file_to_trim
Output: PDB files
Example:

./PDBgenerator.py unique_sets_freq.txt 1pwc.pdb

2.4 PDB generator from SIF and residue list: makePDBfromUniqueModel.py

This script automatically run all steps from 2.1, 2.2 to 2.3.

Input: sif file, format-specific list of amino acid residues, PDB file
Output: csv file or txt file which contains models generated from all possible combinations of interactions
Example:

./makePDBfromUniqueModel.py pdb1pwc_h.sif A:400:_:PNM 1pwc.pdb

or

./makePDBfromUniqueModel.py pdb1pwc_h.sif A:400:_:PNM,A:62:_:SER,A:298:_:HIS 1pwc.pdb

2.5 PDB generator from Probe: probe2PDB.py

This script generates PDb file from Probe file and list of amino acid residues as the basis for the interactions.

Input: Probe file, pdb to trim, list of amino acid residues (comma separated)
Output: PDB file
Example:

./probe2PDB.py 1PWC_h.probe 1pwc.pdb A:400:_:PNM

or

./probe2PDB.py 1PWC_h.probe 1pwc.pdb 400

or

./probe2PDB.py 1PWC_h.probe 1pwc.pdb A:400:_:PNM,A:62:_:SER,A:298:_:HIS

or

./probe2PDB.py 1PWC_h.probe 1pwc.pdb 400,62,298

3 SIF generator

There are two approaches to generate PDB:

- Directly from probe file.
- Similar to RINerator (http://rinalyzer.de/rinerator.php): use both PDB and proble file to select atom by Biopython function named getAtom(), using its alternative location.

Both approaches has differences compare to Sif file generated by RINerator.

- Interaction types is sorted by order: mc, sc, ligand, solvent; interaction types in RINerator Sif file are mixed.
- Redundant interactions are removed.

Example: Redundant interactions are
A:6:_:PRO cnt:sc_solvent A:1017:_:HOH
A:6:_:PRO cnt:solvent_sc A:1017:_:HOH
A:1017:_:HOH cnt:sc_solvent A:6:_:PRO
A:1017:_:HOH cnt:solvent_sc A:6:_:PRO
In our Sif file, only one interaction is kept; depends on which interaction is processed first from probe file.

3.1 SIF generator from probe file: probe2Sif.py

This script takes a probe file as an input and generates SIF file.

Input: probe file
Require: res_mcatoms_dic.py should be placed in the same directory with probe2Sif.py
Output: Sif file
Example: ./probe2Sif.py 1pwc_h.probe

3.2 SIF generator from pdb and probe

This approach reused part of source code from RINerator in trimming probe. There are two steps in the script: trim probe and generate Sif file from probe.

Similar to RINerator, all input file locations must be specified in a job script.

Input: similar input as RINerator program. Inputs are specified inside job script.
Output: a new trimmed probe and SIF file.
Example: ./3bwm_job.py