Compute pKa wrapper

The purpose of the wrapper is to compute pKa values on all titration sites of a protein. The user specifies target protein by providing the PDBID at the last of command line. The wrapper will then return the corresponding pKa values by calling PB solvers for computing free energies. In the file, we provide some details of the code.

1 Procedure

1. Download the PDB file based on the input PDBID. Store the interior part - lines with the beginning "ATOM" into a PDB data structure.
2. Read the well-formatted usrdata.in file and store in a map.
3. Search all the titratable sites (RSDID: ARG, ASP, GLU, HIS, LYS, TYR, CYS) in the PDB file.
4. Change all the titratable sites to its deprotonated status name, ARG → AR0 , ASH → ASP, GLH → GLU, HIP → HIE, LYS → LYN, TYR → TYM, CYM → CYX (this should be checked with extra condition) in PDB. With the prepared PDB file, we can generate an all-mute PQR file with the help of PDB2PQR. Change the input file (usrdata.in) of solver correspondingly, call the solver and fetch the free energy of all-mute state.
5. Compute the intrinsic pKa of each titratable site by the equation,

$$\mathrm{pK_a} = \mathrm{pK_a^0} - \frac{1}{\mathrm{RT}\ln 10}[\triangle \mathrm{G_{ele}}(\mathrm{A_p} - \mathrm{A_pH}) - \triangle \mathrm{G_{ele}}(\mathrm{A_s} - \mathrm{A_sH})]. \tag{1}$$

$\mathrm{pK_a^0}$ is the model $\mathrm{pK_a}$, the values are given in the following, R is the gas constant and T is the temperature in Kelvin. R is related to the Boltzmann constant, $\mathrm{k_B}$, and the Avogadro constant, $\mathrm{N_A}$,

ARG	12.0
ASP	4.0
CYS	9.5
GLU	4.4
HIS	6.3
LYS	10.4
TYR	9.6

Table 1: model pKa value for titratable sites.

$$\mathrm{R} = \mathrm{k_B} \cdot \mathrm{N_A} \approx 8.31\mathrm{J}/(\mathrm{mol} \cdot \mathrm{K}), \tag{2}$$

where $\mathrm{k_B} = 1.3806 \cdot 10^{-23}\mathrm{J/K}$, $\mathrm{N_A} = 6.02 \cdot 10^{23}/\mathrm{mol}$.
T is set to be the room temperature which is around 300K, then RT turns to be,

$$\mathrm{RT} \approx 8.31 \cdot 300\mathrm{J/mol} \approx 2.5\mathrm{kJ/mol} = (2.5/4.182)\mathrm{kCal/mol}. \tag{3}$$

$\triangle\mathrm{G_{ele}}(\mathrm{A_p} - \mathrm{A_pH})$ is the difference of the free energy of protonated protein and the all mute protein. The protonated protein contains one titratable amino acid with one more charge than the all mute protein. $\triangle\mathrm{G_{ele}}(\mathrm{A_s} - \mathrm{A_sH})$ is the difference of the free energy of protonated amino acid alone and the mute amino acid alone. The protonated amino acid is a part of the protonated protein according to what titratable site the intrinsic pKa compute for. The mute amino acid alone is the corresponding part in the all mute protein.
6. Next to prepare the site-site interaction energy, $\triangle\mathrm{G_{ij}}$, denoting the free energy of the protein in which only have two protonated titratable amino acid and mute the rest. Compute $\triangle\mathrm{G_{ij}}$ by modifying the QPR file and call solver.
7. Use the intrinsic pKa, $\mathrm{pK_i^0}$, and site-site interaction energy, $\triangle\mathrm{G_{ij}}$, corresponding with a given pH value to calculate the transfer energy for each status, $\triangle\mathrm{G}(\mathrm{A} \to \mathrm{A}(\theta); \mathrm{pH})$. The status, θ, is determined by combination of different titratable sites. Due to the multiplication principle, the total number of status is 2^{N}, where N is the total number of titratable sites. The formula is given by,

$$\triangle\mathrm{G}(\mathrm{A} \to \mathrm{A}(\theta); \mathrm{pH}) = -\mathrm{RT}\ln 10 \sum_{\mathrm{i}} \theta_{\mathrm{i}}(\mathrm{pK_i^0} - \mathrm{pH}) + \frac{1}{2}\sum_{\mathrm{i}} \theta_{\mathrm{i}} \sum_{\mathrm{j}\neq\mathrm{i}} \theta_{\mathrm{j}} \triangle\mathrm{G_{ij}}. \tag{4}$$

8. Preprocess with the transfer energy for each status, $\triangle\mathrm{G}(\mathrm{A} \to \mathrm{A}(\theta); \mathrm{pH})$, we can substitute it into an equation and check the result relative to $\frac{1}{2}$ with a given error tolerance. If the absolute difference between the result and $\frac{1}{2}$ is less than the error tolerance, then we think the pKa value of certain titratable site is satisfying. The equation is given by,

$$< \theta_{\mathrm{i}}, \mathrm{pH} >= \frac{\sum_\theta \theta_{\mathrm{i}} \mathrm{e}^{-\triangle\mathrm{G}(\mathrm{A}\to\mathrm{A}(\theta);\mathrm{pH})/\mathrm{RT}}}{\sum_\theta \mathrm{e}^{-\triangle\mathrm{G}(\mathrm{A}\to\mathrm{A}(\theta);\mathrm{pH})/\mathrm{RT},}} \tag{5}$$

where the θ_{i} represents one titratable site in the protein, $\theta_{\mathrm{i}} = 1$ when the status have the titratable site protonated, otherwise $\theta_{\mathrm{i}} = 0$.

2 Code Details

The input command to run is

$$\text{python wrapper\_pKa.py PDBID.} \tag{6}$$

The PDBID indicates the abbreviation of a protein that the user has interest in. For example,

$$\text{python wrapper\_pKa.py 4pti,} \tag{7}$$

means the user has interest in pKa values of all the titratable site of 4pti protein.

The wrapper is written in python and has two structs, several methods including main, downloadPBD, writePBD, writeUseData, writePQR, fetchEnergy, run_script, run_with_pqr, intristicPKA, run_intristic, site_site_interaction, interaction_Energy, compute_statue_energy, and is_pKa_okay.

Two data structures to store a line of PBD and PQR. They are similar and we take a line of PQR as an example,

ATOM 5 N PRO 1 12.31600 -13.58000 12.82200 -0.07000 1.85000

The first column is the same for all the rows. The second column is the index of the atom. The third column indicates the element type of the atom. The fourth column is the abbreviation of the amino acid residue to which the atom belongs. The fifth column is the index for the amino acid. The next three columns give the x, y, z coordinates of the atom. The ninth column is the atomic charge and the last column is the atomic radius.

The downloadPBD method reads the online and splits the content into three parts based on the specific PBDID given. The first part is header - the content before the first line beginning with 'ATOM...'. The second is pdb_list - all the lines beginning with 'ATOM...', spliting each line into a PDB data structure and putting them into a list. The third part is trailor - the content after the last line beginning with 'ATOM...'.

The writeUsrdata method rewrites out the usrdata.in file by given content map.

The writePDB and writePQR methods write out a new PDB and PQR file to the input path. If the file exists, doing nothing and return.

The fetchEnergy method has two functions. One is to judge whether the output file from TABI is complete. The other is to search and pass the solvation energy back to main method.

The run_script and run_with_pqr method combine the previous methods together to prepare the input file to the solver, run the solver, and get the results from the solver.

The intristicPKA method is to compute the intrinsic pKa value based on given titratable site using the equation (1) above. The run_intristic method is to call the intristicPKA function

the number of titratable site's times to get the list of intrinsic pKa.
The site_site_interaction method prepares the PQR file based on certain titratable site pairs (s1,s2) and computes the corresponding site-site Energy by calling the solver.
The interaction_Energy loops and calls the site_site_interaction, store the results into a matrix.
Given specific pH value, the compute_statue_energy method calculates all the energy transfers (total number = 2^N), where N is the number of titratable sites. Then, it stores them in a map with keys. The keys are in a binary number in which 1 in i^{th} position represents the i^{th} titratable site is porotonated and 0 means deprotonated. The order of titratable sites is fixed as the site_site list generated before computing calling the function. The referring equation is equation (4)
The is_pKa_okay method computes the numerator of the equation (5) and evaluate the result by substituting with the preprocessed invariable denominator and judge the answer relative to 1/2 with err_tol(error tolerance). If the result is satisfied with the standard, then keep it.
The main method combines all the methods together and computes the pK_a value.