1 pKa computation using TABI solver

1.1 pKa computing procedure

Following the discussion in the previous section, we use the programing language Python to write a wrapper to pipeline the pKa computation in four steps as 1) changing the protonation states on PDB and then on PQR files, with each PQR file representing a needed charge distribution in the pK_a computation, 2) calling MIBPB solver for solving electrostatics potential and free energies for all PQR files, 3) calculating the intrinsic pKa values, and 4) finally titrating the final pK_a values with energies including site-site interactions. The procedure implemented with the Python wrapper is given as the following.

Step 1: Prepare the protein structure and protonation states
In this step, we prepare the structure and charge distribution of the protein. The protein structure is obtained from the Protein Data Bank (PDB) (www.pdb.org). The charge distribution is then produced by PDB2PQR [1] with user chosen forcefields. Before calling the PDB2PQR program, the protonation states are configured by using different residue names as specified below in Table 1. For a protein with N_t titration sites, we will need $1+2N_t+\frac{1}{2}(N_t^2-N_t)$ charge distributions in the form of PQR files as specified below.

Table 1: The protonation states of titratable amino acids: each residue has a name for unprotonated state and a name for protonated states; total charge of the residue in a state is included in the parentheses.

standard resid.	**ARG**	**ASP**	**GLU**	**HIS**	**LYS**	**TYR**	**CYS**
unprot.(chg)	AR0 (+1)	ASP(-1)	GLU(-1)	HIE(0)	LYN(0)	TYM(-1)	CYM(-1)
protonated(chg)	ARG(+2)	ASH(0)	GLH(0)	HIP(+1)	LYS(+1)	TYR(0)	CYX(0)

(1) one PQR file with all titration sites unprotonated, keeping the background charge;
(2) $2N_t$ PQR files having all titration sites unprotonated but one protonated with or without the background charge; those with background charges on are used for calculating the intrinsic pK_a;
(3) $\frac{1}{2}(N_t^2-N_t)$ PQR files with ith and jth titration sites protonated only; all bakground charges are set to zero.

Step 2: Call TABI solver for solving electrostatics
This step calculates the electrostatic free energies for all titration states represented by different PQR files. Each PQR file with different charge distribution changes the RHS of the PB equation. The PB equation can then be accurately and conveniently solved using the TABI solver [2]. In calling the TABI solver, the user can specify solver and PB model related parameters such as dielectric constants, ion concentration, mesh size, etc. in the *usrdata.in* file. After this step, we received electrostatic free energies from all charge distributions. These energies will be used to calculate pK_a as explained in the next two steps. Our work follows the procedure as described in Ho's thesis [3].

Step 3: Compute the intrinsic pK_a

The intrinsic pK_a for the ith titratable site can be computed by Eq. (1.1) [3],

$$\mathrm{p}K_{\mathrm{a},i}^{intr} = \mathrm{p}K_{\mathrm{a},i}^{0} - \frac{1}{RT\ln 10}[\Delta G_{\mathrm{ele}}(\mathrm{A_p} \to \mathrm{A_pH}) - \Delta G_{\mathrm{ele}}(\mathrm{A_s} \to \mathrm{A_sH})] \tag{1.1}$$

where the protein environment has only the fixed background charges, i.e. with all titration sites unprotonated. In this equation, $\Delta G_{\mathrm{ele}}(\mathrm{A_p} \to \mathrm{A_pH})$ is the difference of the free energy between protein with ith titration site protonated and protein with all titration unprotonated while $\Delta G_{\mathrm{ele}}(\mathrm{A_s} \to \mathrm{A_sH})$ is the difference of the free energy between protonated and unprotonated residues alone. Here p$K_{\mathrm{a},i}^{0}$ is the model pK_a at $T = 298$K taken from [4] as in Table 2. For the related constants, R is the gas constant and T is the temperature in Kelvin. R is related to the Boltzmann constant, $k_\mathrm{B} = 1.3806 \cdot 10^{-23}$J/K, and the Avogadro constant, $N_\mathrm{A} = 6.02 \cdot 10^{23}$/mol, as

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

Table 2: model pK_a value for titration sites.

$$R = k_\mathrm{B} \cdot N_\mathrm{A} \approx 8.31\mathrm{J}/(\mathrm{mol}\cdot\mathrm{K}), \tag{1.2}$$

Since the energy calculated from MIBPB using the unit kcal/mol, we finally use the RT values as

$$RT \approx 8.31 \cdot 298\mathrm{J/mol} \approx 2.5\mathrm{kJ/mol} = (2.5/4.182)\mathrm{kcal/mol}, \tag{1.3}$$

thus the energy from MIBPB divided by $(2.5\ln 10/4.182 = 1.3765)$ kcal/mol leads to the unit of pK_a values.

Step 4: Titrating final pK_a with energies including site-site interactions

Recall that pK_a of a titration site is defined as the pH value in which half of the site is protonated. In the context of computing pKa using electrostatic free energy under different titration states, we are looking for the pH which makes the Boltzmann average $<\theta_i,\mathrm{pH}>$ as in Eq. (1.4) to equal 0.5.

$$<\theta_i,\mathrm{pH}> \; = \frac{\sum_\theta \theta_i e^{-\triangle G(\mathrm{A} \to \mathrm{A}(\theta);\ \mathrm{pH})/RT}}{\sum_\theta e^{-\triangle G(\mathrm{A} \to \mathrm{A}(\theta);\ \mathrm{pH})/RT}}, \tag{1.4}$$

where $\theta \in \{0,1\}^{N_t}$ and $\theta_i \in \{0,1\}$ is the ith entry of θ. Note the number of states θ increases exponentially (2^{N_t}) with number of titrating sites N_t, thus is computationally prohibitive. In this paper we use direct formulation for proteins with less than 20 titrating sites and we are working on statistical sampling approaches to handle cases with larger N_t for future work.

In Eq. (1.4), the energy $\triangle G(\mathrm{A}\rightarrow\mathrm{A}(\theta);\mathrm{pH})$ is evaluated as

$$\triangle G(\mathrm{A}\rightarrow\mathrm{A}(\theta);\mathrm{pH}) = -RT\ln 10\sum_i \theta_i(\mathrm{p}K_{\mathrm{a},i}^{intr} - \mathrm{pH}) + \frac{1}{2}\sum_i \theta_i \sum_{j\neq i}\theta_j \triangle G_{ij}. \tag{1.5}$$

where $\mathrm{p}K_{\mathrm{a},i}^{intr}$ is the intrinsic pK_a for the ith titration site as calculated in step 3 by Eq. (1.1). $\triangle G_{ij}$ as calculated in Eq. (1.6) is the site-site interaction energy (the free energy of the protein having the ith site protonated for producing electrostatic potential and the jth titration site protonated for producing energy without the background charge) computed in step 2 as well.

$$\triangle G_{ij} = t_i^T W t_j = \frac{1}{2}(t_i+t_j)^T W(t_i+t_j) - \frac{1}{2}t_i^T W t_i - \frac{1}{2}t_j^T W t_j \tag{1.6}$$

The second equality holds under the assumption that W is symmetric.

1.2 Software dissemination

With the publication of this manuscript, the python wrapper (*wrapper_pka.py*) and the TABI solver on multiple platforms such as MacOS and Linux/Unix) can be found on the author's website sponsored by Southern Methodist University. The user also needs the MSMS software [5] for molecular surface generation and we include its binary versions as well. In addition, the users need to install PDB2PQR [1].

To compute pKa, the user specifies PB model and TABI related parameters (dielectric constants, ion concentration, mesh size, density of molecular surface triangulation, boundary conditions, method of charge regularization, etc.) in *usrdata.in* file. On a computer with Python and Fortran compilers installed and target protein specified with its four-digit PDB ID, the user runs the program by simply typing:

```
python wrapper_pka.py PDBID
```

The wrapper will download PDB file from the protein data bank, identify all titration sites, call TABI solver for electrostatics, and return the computed pK_a values.

References

[1] T. J. Dolinsky, P. Czodrowski, H. Li, J. E. Nielsen, J. H. Jensen, G. Klebe, and N. A. Baker, "PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations," *Nucleic Acids Res.*, vol. 35, 2007.

[2] W. Geng and R. Krasny, "A treecode-accelerated boundary integral Poisson-Boltzmann solver for electrostatics of solvated biomolecules," *J. Comput. Phys.*, vol. 247, pp. 62 – 78, 2013.

[3] K. L. Ho, *Fast Direct Methods for Molecular Electrostatics*. PhD thesis, New York University, 2012.

[4] Y. Nozaki and C. Tanford, "Examination of titration behavior," *Methods Enzymol.*, vol. 11, pp. 715 – 734, 1967.

[5] M. F. Sanner, A. J. Olson, and J. C. Spehner, "Reduced surface: An efficient way to compute molecular surfaces," *Biopolymers*, vol. 38, pp. 305–320, 1996.