Performing pK_a Calculations in Proteins Using Free Energy Perturbation Adiabatic Charging (FEP/AC)

Lynn Kamerlin
Uppsala University

Purpose of the workshop: The aim of this workshop will be to provide a simple introduction to performing pK_a calculations in proteins and in solution, using the free energy perturbation adiabatic charging (FEP/AC) approach. For simplicity, we will be calculating the pK_a of an aspartate in the bovine pancreatic trypsin inhibitor during the hands-on exercise. As a project, you will repeat the process with a more challenging system, namely calculating the pK_a of K102 in the M102K mutant form of T4-lysozome.

Recommended further reading: There is extensive literature available on calculating pK_as. As a starting point, I recommend the following review article: Kamerlin *et al*., "Progress in *ab initio* QM/MM free-energy simulations of electrostatic energies in proteins: Accelerated QM/MM studies of pKa, redox reactions and solvation free energies". J*. Phys. Chem. B.*, **113** (2009), 1253, which this workshop will be partially based. You will also find several useful references cited for further reading cited there.

Why should we care about the pK_as of ionizable groups in proteins?

By definition, a pK_a is a quantitative measure of the strength of an acid in solution. Fig. 1 shows an overview of the structures of the 20 naturally occurring amino acids. Of these, seven of have ionizable sidechains, namely Asp, Glu, His, Cys, Tyr, Lys and Arg. The solution pK_as for these amino acids are shown in Table 1. However, the pK_a of an amino acid in a protein can be very different from that in solution, as, once the protein folds, the amino acid will find itself transferred from solution (solvent-exposed) to an environment that can be very different (e.g. it could suddenly find itself buried deep in a hydrophobic pocket). Additionally, not only can the amino acid find itself suddenly no longer exposed to solvent, but also, it is now in the vicinity of other ionizable groups, as well as interacting with other permanent charges and protein dipoles. As a result, the

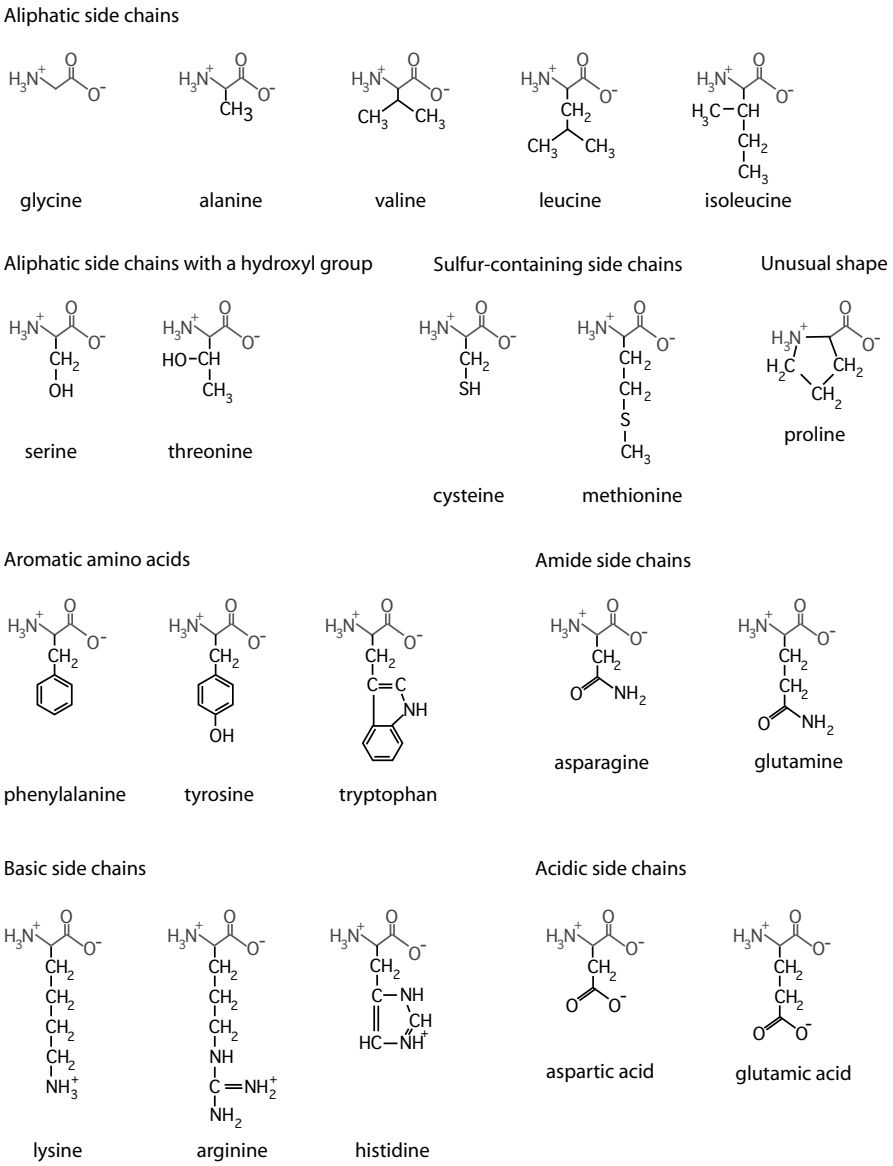


Figure 1: Overview of the structures of the 20 naturally occurring amino acids. Image courtesy of Jeanette Mowery from Madison Area Technical College.

Ionizable Amino Acid	Side Chain pK_a	Ionized Group
Asp(D)	3.9	$-COO^-$
Glu(E)	4.3	$-COO^-$
His(H)	6.1	$-NH^+$
Cys(C)	8.3	$-S^-$
Tyr(Y)	10.1	$-O^-$
Lys(K)	10.5	$-NH_3^+$
Arg(R)	12.0	$-NH_2^+$

Table 1: Overview of solution pK_as and charges of the ionized for amino acids with ionizable sidechains.

electrostatic environment can cause quite spectacular shifts in the pK_a values of amino acids in a protein compared to the corresponding values for model compounds in solution, according to the new environment the amino acid finds itself in. The resulting stability of the protein is then dependent on the ionization states of its ionizable residues, as demonstrated by the fact that a change in ionization state of e.g. a residue buried deep in a hydrophobic pocket has the capability to trigger a large conformational response in the respective protein. Additionally, the pK_a of an ionizable group can be an important guide to determining its correct protonation state, information that is essential when attempting to model chemical catalysis by enzymes. Following on from this, the pK_a values of the relevant ionizable groups in a protein determine both the pH dependence of the activity and the stability of an enzyme. Finally, considering the important role of electrostatics in the determination of pK_as of ionizable groups in a protein, and the availability of reliable experimentally measured values for these pK_as, being able to correctly reproduce the pK_as of ionizable groups in a protein is perhaps one of the best benchmarks for verifying the quality of how a given computational approach is treating electrostatic effects.

How does one calculate pK_as of ionizable groups in proteins?

Clearly, there are several experimental ways to do this reliably. These include differential titration, NMR studies, and more recently, time-of-flight neutron diffraction (see e.g. Katz *et al*., *Proc. Natl. Acad. Sci. USA*, **103** (2006), 8342). Here, we will focus on computational approaches, and, specifically, calculating pK_as of ionizable groups in proteins using the free energy perturbation adiabatic charging approach (FEP/AC). This approach involves evaluating the free energy of charging an ionizable group in a protein from its non-polar neutral form, to it's fully charged form (using the relevant charges for this group in solution). This approach uses a mapping potential of the form:

$$E_k = E_{tot}(1-\lambda_k) + E'\lambda_k \qquad (1)$$

where E_{tot} denotes the total energy of the system, E' denotes the energy of the system without the electrostatic solute-solvent interactions, and λ_k is progressively modified from 0 to 1 in n+1 mapping increments. From here, the solvation free energy can simply be evaluated using the standard FEP equation:

$$\Delta\Delta G_{solv}(\lambda_k \rightarrow \lambda_{k+1}) = -\beta^{-1} \ln < \exp\{-(E_{k+1} - E_k)\beta\} >_{E_k}$$
$$\Delta G_{solv} = \sum_{k=1}^{n+1} \Delta\Delta G_{solv}(\lambda_k \rightarrow \lambda_{k+1}) \quad (2)$$

where $\beta=1/(k_BT)$ (k_B is Boltzmann's constant and T denotes the absolute temperature), and $<>_{E_k}$ represents the average obtained during the propagation of configurations that use E_k. The pK_a shift upon moving the ionizable group from the solution to the protein interior can then be evaluated using the thermodynamic cycle shown in Fig. 2.

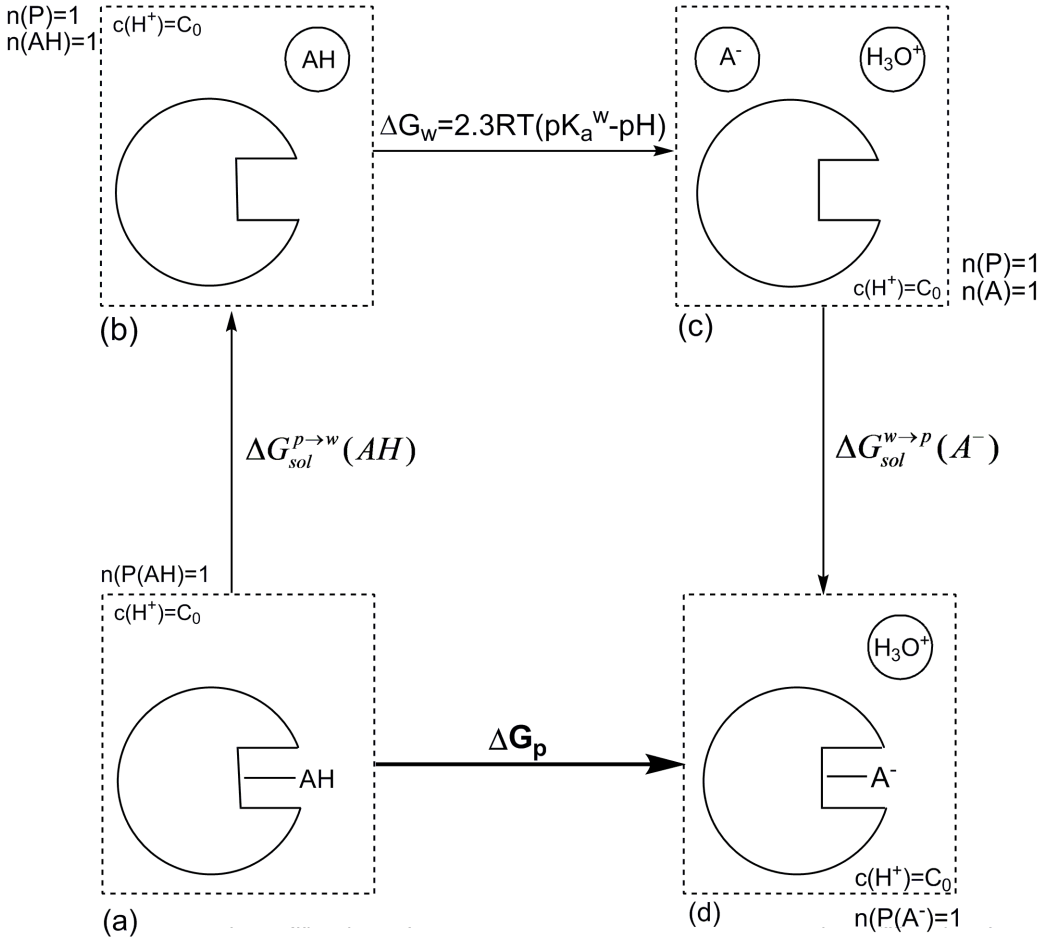


Figure 2: Thermodynamic cycle for evaluating the change in solvation free energy upon transferring an ionizable group from solution to a protein interior.

In Fig. 2, the energy of the ionization process in the protein is represented in terms of the energetics of the corresponding reaction in water (b→c), as well as the solvation free energies of the neutral (AH) and ionized (A^-) forms of the species in the protein. In the first step (a→b), 1 mol of neutral acid (AH) bound to the protein is transferred to a solution at a constant pH corresponding to a hydrogen ion concentration of C_0. The

energetics of this process can be simply obtained from the difference between the solvation free energies of AH in water and in protein (i.e. $\Delta G_{solv}^{p\to w}(AH)$). In the second step, 1 mol of AH is ionized to A^- in solution (again at constant pH), and the energetics of this step can simply be obtained from the difference between the pK_a of the ionizable group in solution (pK_a^w) and the pH, i.e. $\Delta G_w = 2.3RT(pK_a^w - pH)$, where R is the ideal gas constant and T the temperature. Finally, in the last step, the solvated ionized species, A^-, is moved from water back to the protein, where, as in step 1, the energetics of this process is related to the difference between the solvation energies of the ionized species in water and in the protein ($\Delta G_{solv}^{w\to p}(A^-)$). Therefore, the free energy of ionizing an acid in a protein at any given pH (ΔG_p) can be evaluated by:

$$\Delta G_p(AH_p \to A_p^- + H_w^+) = \Delta G_{sol}^{w\to p}(A^-) - \Delta G_{sol}^{w\to p}(AH) + \Delta G_w(AH_w \to A_w^- + H^+) \quad (3)$$

where p and w designate protein and water respectively, and $\Delta G_{solv}^{w\to p}$ designates the free energy difference of moving the ionizable group in it's relevant protonation state from water to the protein. Eq. 3 can be further simplified for the i^{th} ionizable residue to give:

$$pK_{a,i}^p = pK_{a,i}^w - \frac{\bar{q}_i}{2.3RT}\Delta\Delta G_{sol}^{w\to p}(AH_i \to A_i^-) \quad (4)$$

where $\bar{q}_i$ represents the charge of the ionized form of the group (-1 for acids, +1 for bases), and $\Delta\Delta G_{sol}^{w\to p}(AH_i \to A_i^-)$ simply consists of the first two terms on the right hand side of Eq. 3.

It should be noted that what we are evaluating in this workshop using the approach outlined above is the **intrinsic pK_a** of the ionizable group ($pK_{a,int}$), when all other ionizable groups are uncharged. In order to obtain the actual *apparent* pKa ($pK_{a,app}$), we would have to also evaluate the effect of charging all other ionizable groups in the protein to their given ionization state, and add this to the intrinsic pK_a as a correction. In principle, it is possible to do this macroscopically, using a distance dependent dielectric constant. However, this is out of the scope of this workshop, and here, the focus will be on FEP/AC calculations of the charging of Asp3 in BPTI and K102 in the M102K T4-Lysozyme mutant.

MOLARIS Overview

For the purposes of this workshop, we will be using the MOLARIS software package, developed by Warshel and coworkers at the University of Southern California (see http://www.futura.usc.edu). MOLARIS is a multipurpose simulation package that comprises two programs, namely POLARIS and ENZYMIX (which were originally two separate simulation packages). Our focus here will be on using ENZYMIX (with the corresponding ENZYMIX forcefield), which is a macromolecular simulation program specifically designed in order to study functional properties of proteins, ranging from ligand binding free energies to free energy profiles of enzymatic reactions using the empirical valence bond (EVB) and free energy perturbation (FEP) approaches. Here, we will be using ENZYMIX for the aforementioned example of FEP/AC calculations. The full MOLARIS manual is available for download at:
http://futura.usc.edu/programs/doc/molaris_manual.pdf
and the corresponding theoretical background can be found at:
http://futura.usc.edu/programs/doc/molaris_theory_2009.pdf
Below, an overview of the key points relevant to this workshop will be provided.

Running MOLARIS

MOLARIS has been installed on the server muon.nbi.dk. In order to run MOLARIS, you will need 3 things. The first of these is access to the binary, which can be found at /scr/ess/essmed00/molaris/bin/molaris_9.09. The second of these is access to the MOLARIS library files, which are found in the directory /scr/ess/essmed00/molaris/lib. Finally, the third of these is a pdb structure containing coordinates of the relevant system (either the protein or ligand). Before running MOLARIS, you will need to tell your system where your binary and libraries are. This can be achieved by setting appropriate environment variable and sourcing the file .molaris_rc, which contains paths to each of the relevant library files. Enter the cshell by typing csh, and, on the command line, type the following sequence of commands:

setenv MOLARIS_PATH /scr/ess/essmed00/molaris/
setenv PATH "$MOLARIS_PATH/bin:$PATH"
source /scr/ess/essmed00/molaris/bin/.molaris_rc

This tells your system where to find the MOLARIS binary, and gives MOLARIS the following information: (1) the path to the directory containing the library files, (2) locations for the relevant library files and (3) the path to the directory where MOLARIS will put the generated output files. There are four library files of importance here, specifically **amino98.lib** which contains topologies for all the relevant available residues, **parm.lib** which is the ENZYMIX parameter library, **evb.lib** which is the parameter library for EVB calculations and **solvent.opt** which includes information such as densities, bond lengths, charges etc for the solvent being used. Note that even though we will not be performing EVB calculations in this workshop, nevertheless, MOLARIS still needs to know the path to the evb library in order to be able to run the program. Once you have told MOLARIS where to find the relevant library files, you can now run MOLARIS by typing:

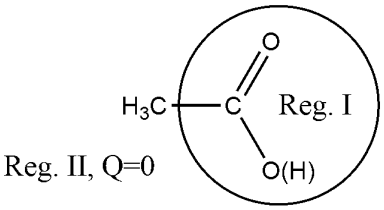
molaris <filename.inp >filename.out &

Note that running MOLARIS without an output filename will redirect output to STDOUT (in this case your screen), and if you do not give MOLARIS an input file name this allows you to run MOLARIS interactively.

A sample MOLARIS FEP/AC input file for BPTI is shown below, with highlights to illustrate what each point on the input file is. In order to evaluate a pK_a using the thermodynamic cycle shown in Fig. 2, you will need to run 8 discrete calculations. Specifically, you will need to evaluate the solvation free energy of charging the neutral and ionized forms of the relevant ionizable group in both water and protein (in both forward and reverse directions to check for convergence), and each case is an individual calculation. Prior to the AC run, you will run a short relaxation run to relax the protein structure, after which you will run FEP/AC in 26 mapping frames. Note that the number of frames the FEP/AC is run in is flexible, but the smaller the number of frames, the longer each individual frame will need to be to obtain reliable convergence, and vice versa. For a small system, the minimum total simulation time to get reliable convergence would be ~250ps. Therefore, the short runs we will run today will only provide an initial number with a very large error so that you can understand the basics of the approach, and much longer runs would be needed to be able to obtain reliable results (at least 10,000 steps (10ps)/frame). The charges in this case have been obtained from *ab initio* using the COSMO solvation model and a small basis set (6-31+G* 5D). However, note that in principle, charges could also have been taken from

amino98.lib. Finally, in order to obtain more rapid convergence, we will be performing the FEP/AC calculations with a fixed solute. However, ideally, in order to obtain proper sampling, one wants to also allow the solute to fluctuate and not just the solvent/environment.

```
bpti.pdb                    ! Name of pdb file.
Enzymix                     ! Initialize ENZYMIX
  pre_enz                   ! Prepare ENZYMIX (set charges)
setcrg  43     0.000
setcrg  44     0.000
setcrg  45     0.000
setcrg  46     0.809
setcrg  47    -0.650
setcrg  48    -0.639
setcrg  49     0.480
end
  relax                     ! Initial relaxation run
  rest_out bpti.res         ! Restart file name
  md_parm                   ! Define MD parameters
      region2a_r 20.0       ! Radii for region 2 atoms, water sphere
      water_r 20.0                    and langevin grid
      langevin_r 20.0
      temperature 300.      ! Simulation temperature
      ss 0.001              ! Stepsize in ps
      nsteps 1000           ! Number of steps
      fix_atom 43 to 49     ! Atoms to fix (solute)
  end
  end
end
enzymix                     ! Actual AC run, reinitialize ENZYMIX
  ac                        ! AC module
  rest_in bpti.res          ! Read in restart from relaxation
  reg1_atm 46 to 49         ! Define AC atoms
  ab_crg 43  0.000   0.000  ! Define AC atom charges
  ab_crg 44  0.000   0.000
  ab_crg 45  0.000   0.000
  ab_crg 46  0.000   0.809
  ab_crg 47  0.000  -0.650
  ab_crg 48  0.000  -0.639
  ab_crg 49  0.000   0.480
map_lambda 1.0              ! Mapping starts at 100% state A
   map_pf 26 1 2            ! Perform mapping in 26 frames
   md_parm
      region2a_r 20.0
      water_r 20.0
      langevin_r 20.0
      temperature 300.
      ss 0.001
      nsteps 1000
      fix_atom 43 to 49
      end
   end
   end
end
end
end
```



Once the FEP/AC runs have completed, you then need to run the mapping program (mapping_9.09, found in the same directory as the MOLARIS binary) in order to be able to obtain the free energy. During the AC runs, energetic information during the run has been written to 26 gap files, titled map_ac.gap0xx (where xx=01 to 26). The mapping program then extracts this information and patches together the information from each frame to obtain the total solvation free energy for the run. A very simple sample mapping input file looks like:

```
mapping_type AC
fileroot map_output/ac.gap 26
points_throw 10

end
```

And this will tell the mapping program (1) what kind of run has been performed (e.g., AC, EVB, …), (2) what the root of the name for the gap files is, where to find them, and how many of them there are, (3) how many points to throw from the beginning of each frame (equilibration period at that frame) and (4) to exit the program. The mapping program can be run by typing:

mapping <ac_map.inp >ac_map.out &

And, as with MOLARIS, this can be run either interactively or through the input file. The relevant information will then be extracted from the output file as outlined in the workshop.

Test System: Bovine Pancreatic Trypsin Inhibitor

For our test system, we will be examining the bovine pancreatic trypsin inhibitor, BPTI. BPTI is a classical test system for pK_a calculations, but also, it is a very small protein (only 58 residues) allowing for rapid calculations in the limited time of this workshop. Fig. 3 shows the position of the solvent exposed residue Asp3 on BPTI, as well as some relevant nearby sidechains. The pK_a of this residue has been determined by NMR by Wütrich and Wagner in 1979 to be 4.0 (compared to a pK_a of 3.9 in solution), and thus the pKa shift for this residue is very small. In addition, the experimental solvation energy of the acetate ion in solution is ~-80.7 kcal/mol, and that of the

unionized ion is -10 kcal/mol. Since both pK_a^p and pK_a^w are so similar, it is expected for $\Delta\Delta G_{solv}(AH \rightarrow A^-)$ to be ~-70 kcal/mol in both protein and solution, and our aim is to reproduce this with as high accuracy as possible.

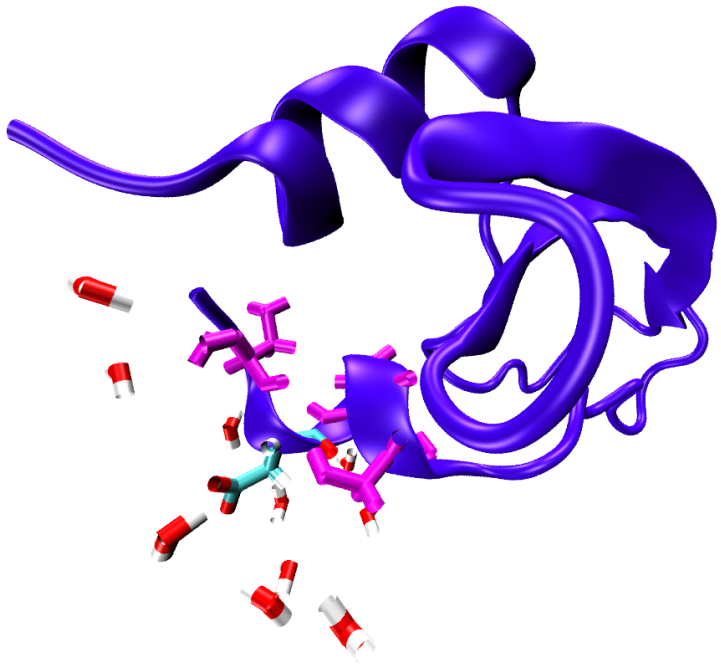


Figure 3: Position of Asp3 in the Bovine Pancreatic Trypsin Inhibitor (BPTI).

Independent Project

For your independent project, you will be moving to a more challenging system, namely K102 in the M102K mutant form of T4 Lysozyme (which is another benchmark system for pK_a calculations). What makes this simulation so challenging is the fact that K102 is buried deep in a hydrophobic pocket (see Fig. 4), and changing the protonation state of a buried residue has the potential to trigger a large conformational response in the system. Therefore, this system is a perfect example of the need for proper sampling. The pK_a of this system has been measured by differential titration and NMR to have a significant downward shift of four pKa units, yielding a pKa of 6.5 for K102 compared to 10.5 in solution. This corresponds to a significant destabilization of 2-9 kcal/mol over a pH range of 10-3 respectively, and attempts to study this system using QM/MM have overestimated this pK_a shift by up to 11.6 pK_a units (i.e. an error of 16 kcal/mol) (see Riccardi *et al.*, J. Phys. Chem. B. **109** (2005), 17715). A PDB structure for the mutant T4

lysozyme as well as *ab initio* charges for the K102 sidechain in solution have been provided. Your aim will be to set up the system as for BPTI and to repeat the same process. *However*, you will need to run longer runs in order to be able to obtain reliable results, so these should ideally be left overnight. Your challenge is to obtain better reliability than an overestimate of 16 kcal/mol in this pK_a shift. Good luck!

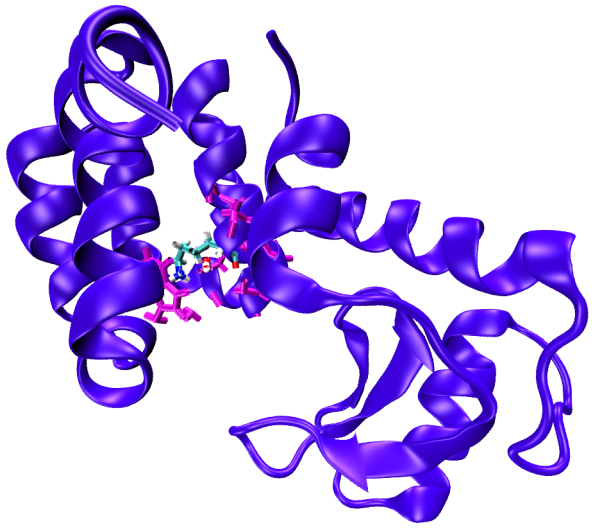


Figure 4: Position of K102 in the M102K T4 lysozyme mutant. Note that this residue is buried deep in a hydrophobic pocket, and thus any change in its protonation state can potentially trigger a large conformational response in the protein, emphasizing the importance of proper sampling.