Bioinformatics, 2026, **42**(2), btag063
https://doi.org/10.1093/bioinformatics/btag063
Published: 12 February 2026
Applications Note | Structural bioinformatics

OXFORD

TrIPP: a trajectory iterative pK_a predictor

Christos Matsingos[1,2,‡], Ka Fu Man[1,‡], Arianna Fornili[1,*]

[1]Department of Chemistry, School of Physical and Chemical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom
[2]Saints-Pères Paris Institute for the Neurosciences, Université Paris Cité, Paris 75270, France
*Corresponding author. Department of Chemistry, School of Physical and Chemical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, UK. E-mail: a.fornili@qmul.ac.uk
[‡]These authors contributed equally to this work.
Associate Editor: Daisuke Kihara

Abstract

Summary: **The protonation propensity of ionizable residues in proteins can change in response to changes in the local residue environment. The link between protein dynamics and pK_a is particularly important in pH regulation of protein structure and function. Here, we introduce TrIPP (Trajectory Iterative pK_a Predictor), a Python tool to track and analyze changes in the pK_a of ionizable residues along Molecular Dynamics trajectories of proteins. We show how TrIPP can be used to identify residues with physiologically relevant variations in their predicted pK_a values during the simulations and link them to changes in the local and global environment.**

Availability and implementation: **TrIPP is available at https://github.com/fornililab/TrIPP.**

1 Introduction

The conformation of proteins can depend on different factors, including the pH. Indeed, several examples exist of proteins that undergo structural shifts in response to physiological pH changes (Di Russo *et al.* 2012, Schönichen *et al.* 2013, Warwicker 2022, Matsingos *et al.* 2024). The link between pH and protein structure is mediated by the dependence between pK_a values, which regulate the protonation propensity of ionizable residues, and the local residue environment, which can significantly change with the protein conformation.

Due to the difficulty in experimentally measuring pK_a values, different computational methods (Rabenstein and Knapp 2001, Gordon *et al.* 2005, Alexov *et al.* 2011, Olsson *et al.* 2011, Anandakrishnan *et al.* 2012, Reis *et al.* 2020, Gokcan and Isayev 2022, Wei *et al.* 2023) have been developed to predict pK_a deviations from their reference values (isolated amino acid in solution). These approaches are normally applied to single structures, e.g. to define the protonation state of ionizable residues before starting a Molecular Dynamics (MD) simulation. Few studies have also considered pK_a calculations over structural ensembles to take into account that structural variations observed during MD simulations can influence predicted pK_a values (Lee *et al.* 2014, Matsingos *et al.* 2024). Building on this multiple-structure perspective, we introduce TrIPP (Trajectory Iterative pK_a Predictor), a tool to track, analyze, and visualize pK_a changes when postprocessing MD trajectories.

TrIPP is based on the iterative use of PROPKA 3 (Olsson *et al.* 2011), one of the most popular empirical approaches for single-structure pK_a prediction. In addition, TrIPP provides optional support to run the deep learning predictor pKAI (Reis *et al.* 2022), trained on Poisson Boltzmann-based predictions from PypKa (Reis *et al.* 2020). We show that TrIPP can be used to visualize pK_a distributions and time evolutions over MD trajectories and identify residues with conformation-dependent pK_a values. Pseudo-mutations can be carried out to assess the influence of specific residues on pK_a regulation. In addition, TrIPP can cluster trajectory frames according to the local environment of selected ionizable residues, to identify structures representative of the environments leading to different pK_a values. Insights from TrIPP can be used to guide further studies and formulate mechanistic hypotheses on pH-dependent regulation of protein function.

2 Implementation

TrIPP has been implemented using Python 3.9 along with several Python packages, including PROPKA 3.5.0 (Olsson *et al.* 2011), pKAI 1.2.0 (Reis *et al.* 2022), MDAnalysis 2.5.0 (Michaud-Agrawal *et al.* 2011, Gowers *et al.* 2016), Pandas 2.0.3 (McKinney 2010), NumPy 1.25.0 (Harris *et al.* 2020), scikit-learn 1.5.0 (Pedregosa *et al.* 2011), scikit-learn-extra 0.3.0, scipy 1.13.0 (Virtanen *et al.* 2020), and seaborn 0.13.2 (Waskom 2021). The software has been packaged with Poetry 1.7.1 and is available for installation through PyPI.

2.1 TrIPP workflow

The workflow comprises four stages: data input, data preprocessing, pK_a prediction, and data analysis (Fig. 1A).

Received: 9 September 2025. **Revised:** 7 January 2026. **Accepted:** 29 January 2026

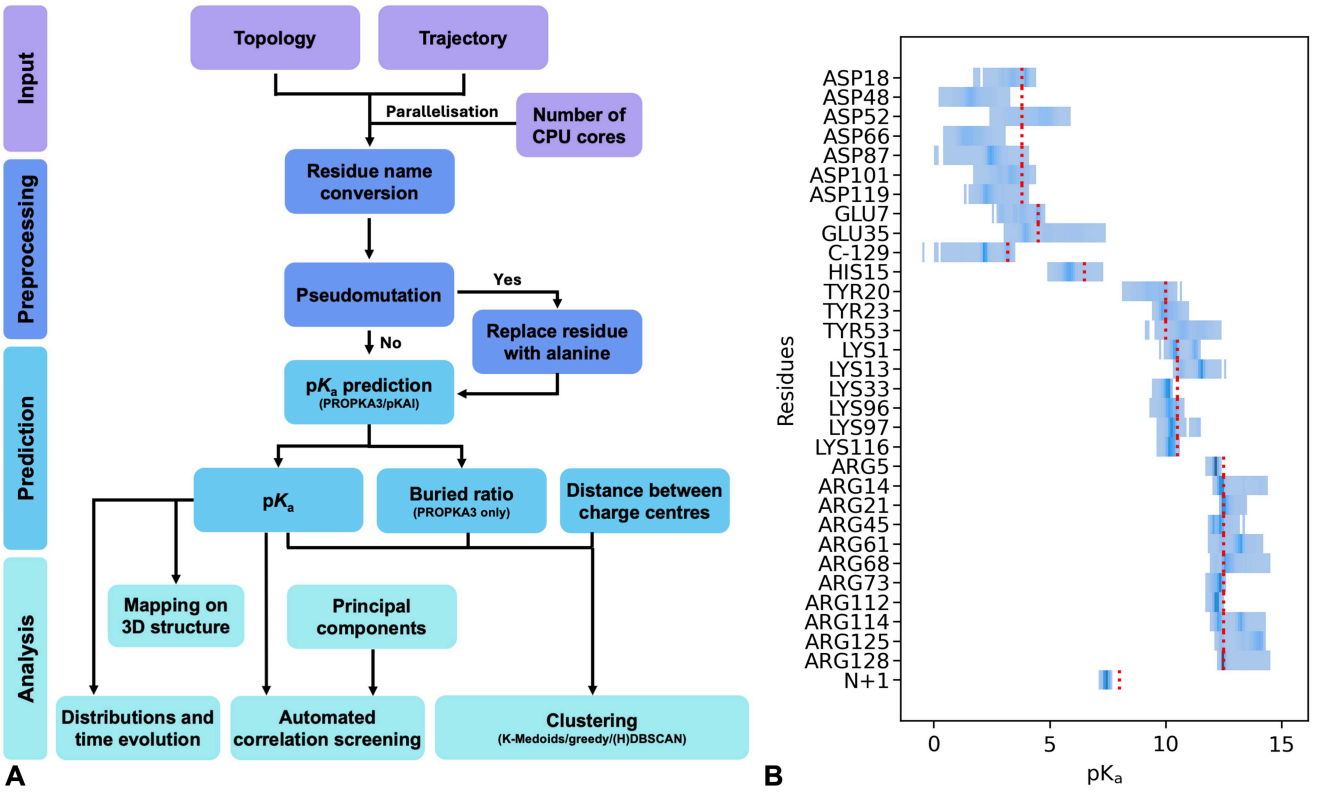

Figure 1 TrIPP overview and application to the hen egg white lysozyme. (A) Different stages in the TrIPP workflow. (B) Distributions of predicted PROPKA pK_a values for ionizable residues in lysozyme during one of the trajectories (MD1). For each residue, pK_a values are binned along the *x*-axis and more frequent values are indicated with a darker shading. A dotted vertical line indicates the model value for the corresponding residue type.

Data input. TrIPP uses MDAnalysis to load input trajectory and topology files. Parallelization is carried out by splitting each trajectory into as many subsets of frames as the number of cores used in the calculation. Each subset is processed by a different core in the subsequent steps.

Data preprocessing. Preprocessing includes an optional pseudo-mutation feature, which can be activated by the user to replace selected residues with alanine. Where required, residue names are also converted into versions recognized by PROPKA. Each snapshot is then temporarily converted to a PDB file, which is the format required by PROPKA.

Prediction. The chosen pK_a predictor (PROPKA by default) is iteratively run on each snapshot (via Python API) and its output is parsed to extract the pK_a values and (optionally only for PROPKA predictions) the buried ratio (given as %) of all the ionizable residues. Temporary files (e.g. PDB snapshots and PROPKA output) are deleted after processing. The extracted values are stored in CSV files and sorted by snapshot timeframe.

Data analysis. Plots of pK_a distributions and time evolutions can be generated from the pK_a values predicted for each snapshot. TrIPP can write PyMOL sessions (PSE format) to colour-map different properties on the 3D structure, including the pK_a values of all the ionizable residues for a selected frame or averaged over a set of trajectories. The difference between the average and reference model pK_a values used by the specific predictor can be also mapped. If the projection of the trajectories on selected principal components is provided by the user, an automatic scanning of the correlation coefficients between projections and pK_a values can be run to identify possible pK_a-collective motions coupling.

The generated data can be used to extract representative structures from the trajectory using one of the available clustering algorithms: K-Medoids (Kaufman 2005), greedy (Micheletti *et al.* 2000), DBSCAN (Ester *et al.* 1996), and HDBSCAN (Campello *et al.* 2013). The feature matrix used for the clustering is built from pK_a values of user-selected residues and (optionally) their buried ratio (PROPKA only) and inter-residue distances. Features are Z-score normalized before clustering and (optionally) subjected to Principal Component Analysis (PCA). For each method, clustering hyperparameters can be optionally optimized via grid search, using silhouette widths (Rousseeuw 1987) to assess the quality of the clustering.

At last, TrIPP provides a function to automatically scan all ionizable residues for possible correlations between their pK_a values and a time-dependent property provided by the user. For example, projections from a principal component analysis of the trajectories can be used to represent collective motions in the protein.

2.2 TrIPP classes

TrIPP provides its functionalities through 3 main classes: *Trajectory* (input, preprocessing and prediction), *Visualization* (3D mapping of pK_a- related values) and *Clustering*. A comprehensive tutorial demonstrating the use of these classes is included in the TrIPP github distribution as Jupyter notebook.

3 Application

TrIPP was applied to MD simulations of the hen egg-white lysozyme to illustrate its features. This enzyme was chosen since the pK_a values of its residues have been extensively studied both experimentally and computationally (Williams *et al.* 2010, Webb *et al.* 2011). In particular, the pK_a of acidic residues in its active site has been shown to be modulated by interactions with nearby residues (Williams *et al.* 2010). MD simulations were run for 300 ns (production) in five replicates (MD1–MD5), system preparation and MD protocol are described in the Supplementary Methods and Table 1, available as supplementary data at *Bioinformatics* online. TrIPP was run on each replica using frames sampled every 100 ps and PROPKA as predictor.

Average pK_a values show small variations across replicas, indicating good reproducibility, and a good agreement with the available experimental values (Table 2, available as supplementary data at *Bioinformatics* online). Inspecting the distribution of pK_a values observed for all ionizable residues (Fig. 1B and Fig. 1, available as supplementary data at *Bioinformatics* online) highlights the residues that have pK_a values (blue shades) significantly shifted from their model pK_a (dotted line). Among the acidic residues, Asp52 and Glu35 stand out because their pK_a values are upshifted towards the region of physiological pH. This can be also easily spotted in the PyMOL visualizations of the average pK_a values for acidic residues and their deviation from model values (Fig. 2, available as supplementary data at *Bioinformatics* online), where Asp52 and Glu35 are the only acidic residues with an increase in the average pK_a from the model values. While the average increase is small, inspection of the pK_a time evolution for these residues indicates that values can be as high as ∼7 for Glu35 and almost 6 for Asp52 (Fig. 2A). Interestingly, Glu35 and Asp52 are located in the lysozyme active site and are directly involved in the catalytic mechanism (Ramos *et al.* 2025).

It is worth noting that these simulations were performed using conventional MD, in which the protonation states of ionizable residues are fixed. Comparing the pK_a values sampled during the simulations with the range over which both protonation states are expected to have significant populations (grey shaded region in Fig. 1, available as supplementary data at *Bioinformatics* online) can flag residues for which alternative fixed protonation states or constant-pH MD (Oliveira *et al.* 2022) may be warranted, to ensure that conformational sampling remains consistent with protonation propensities over the course of the simulation. Notably, a subset of frames falls within this range for Glu35 (Fig. 1, available as supplementary data at *Bioinformatics* online), indicating appreciable population would be expected for the protonated form, consistent with protonation of this residue during catalysis (Ramos *et al.* 2025).

To detect possible couplings between pK_a changes and protein dynamics, the correlation coefficient between the pK_a time evolution and the main collective motions observed during the simulations (as described by projections of each trajectory on its

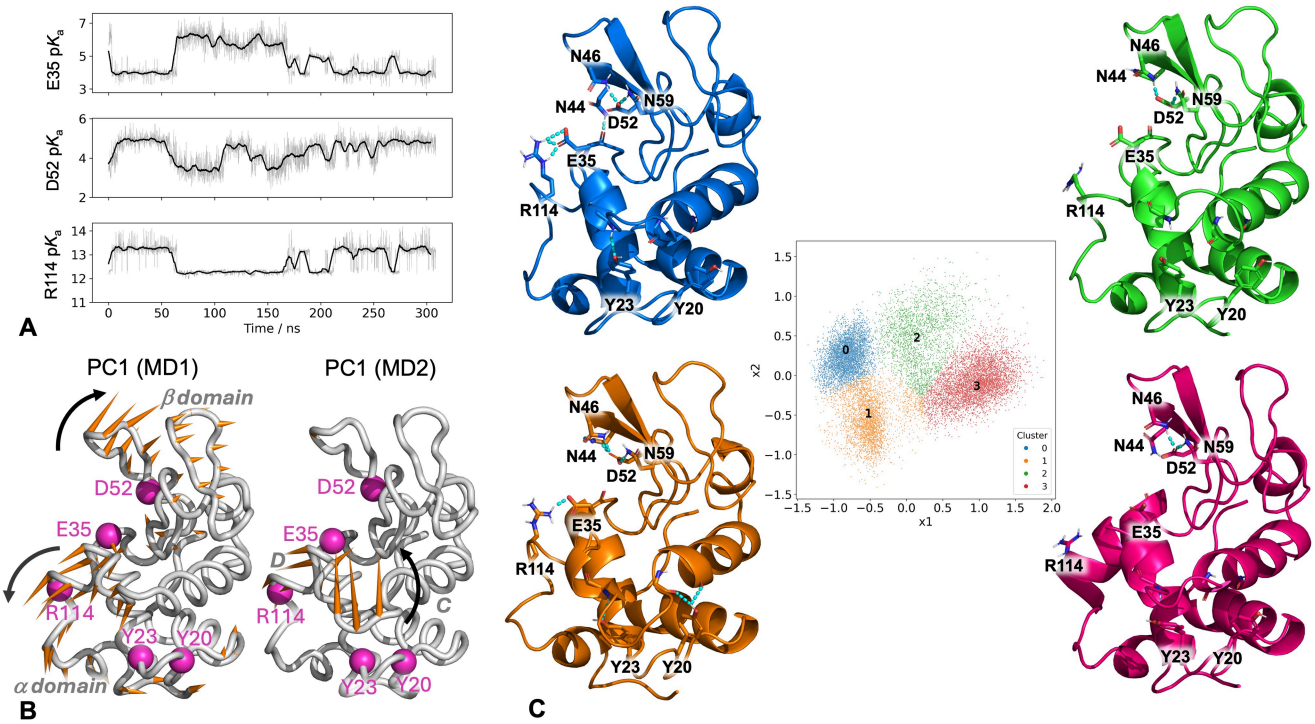

Figure 2 TrIPP-based analysis of MD simulations of lysozyme. (A) Time evolution of pK_a values predicted for Glu35, Asp52, and Arg114 during one of the lysozyme trajectories (MD1). The running average (5-ns window, dark shade) is shown together with the instantaneous values (light shade). (B) Porcupine representation of the first Principal Component (PC1) from the MD1 and MD2 lysozyme trajectories. Orange spikes show the direction and relative amplitude of motion of each residue along the PC (spikes shorter than 20% of the maximum length are omitted). Key residues are labelled and their position indicated with magenta spheres. Additional labels indicate the α and β domains in the left panel, and the C and D helices in the right panel. (C) K-Medoid cluster representatives (cartoon) illustrating different structural environments leading to different pK_a values for Glu35, Asp52, Arg114, Tyr20, and Tyr23. Key residues are highlighted as sticks and labelled. Polar contacts from PyMOL are represented with dashed lines. The feature matrix used for clustering was first subjected to dimensionality reduction via a Principal Component Analysis. The middle panel shows a projection of all trajectories on the first two principal components x_1 and x_2, with each frame coloured according to its cluster ID.

top-ranking principal components) was calculated for each ionizable residue (Fig. 3, available as supplementary data at *Bioinformatics* online). The largest correlations (0.5 or higher) with the first two principal components (PC1 and PC2) were observed for Tyr20, Glu35, Asp52, Arg114, and to a less extent Tyr23, suggesting that the dominant protein motions can modulate the pK_a value of these residues.

Two different types of recurring motions were identified. The first one, exemplified by PC1 from MD1 (Fig. 2B, left), is the previously observed hinge-bending motion of the α- and β-domains(Brooks and Karplus 1985), where Asp52, Glu35, Arg114 are part of or close to the moving regions. The projections of the different trajectories (Fig. 4A, available as supplementary data at *Bioinformatics* online) show patterns that are clearly correlated (Glu35) or anti-correlated (Asp52, Arg114) with pK_a time evolutions (Fig. 5, available as supplementary data at *Bioinformatics* online). PC1 from MD2 instead describes a more localized restructuring of the loop between helices C and D, which is positioned above Tyr20 and Tyr23 (Fig. 2B, right).

Trajectories were then clustered to identify structures representative of the different environments around residues with motion-sensitive pK_a values (Fig. 2C). Clusters 0 and 1 feature a salt-bridge between Glu35 and Arg114, which is instead absent in clusters 2 and 3. This explains their distinct pK_a distributions across the clusters (Fig. 6, available as supplementary data at *Bioinformatics* online), since acidic and basic residues forming salt bridges are expected to have down- and up-shifted pK_a values, respectively, compared to their non-interacting states. Clusters 1 differs from cluster 0 because of the presence of additional hydrogen bonding between Asp52 and Asn44, which explains the lower Asp52 pK_a values for this cluster, and between Tyr20 and the CD loop backbone, which leads to increased Tyr20 pK_a values.

The environment of Glu35 and Asp52 was further analyzed using the pseudo-mutation tool implemented in TrIPP, where pK_a values are re-calculated after replacing selected residues with alanine, while keeping all the other coordinates in the protein unchanged. It is important to note that the pseudo-mutation feature is meant to help tracking the contribution of nearby residues to pK_a shifts from model values in the original (wild type) protein, rather than providing estimates of pK_a in mutants. Such quantitative assessment would require directly sampling the mutant ensemble with new MD simulations.

Replacing residues Asn44, Asn46 and Asn59 with alanine shows an increase of Asp52 pK_a across all the replicas (Fig. 7A and C, available as supplementary data at *Bioinformatics* online). This is consistent with the formation of hydrogen bonds between all these residues and Asp52. Interestingly, instead of just shifting the Asp52 pK_a time evolution profile, Asn44 also modulates its shape. Indeed, the Asn44Ala pseudo-mutant shows smaller pK_a fluctuations compared to the wild type (Fig. 7C, available as supplementary data at *Bioinformatics* online), consistently with the formation and breaking of Asn44-Asp52 hydrogen bonds during the trajectory. The role of Asn44 in stabilizing Asp52 negative charge and modulating its availability for substrate binding has been highlighted in recent structural studies of lysozyme (Ramos *et al.* 2025). Similarly, replacing Arg114 with alanine induces an increase of the Asp52 pK_a but only in the parts of the trajectory where the two residues are interacting with each other (Fig. 7B and D, available as supplementary data at *Bioinformatics* online). These examples illustrate how analyzing the effect of pseudo-mutations on the shape of the pK_a time evolution profile of a given residue can help identifying the residues that mediate the dynamic response of its pK_a value.

4 Concluding remarks

We developed a Python tool to track, analyze and visualize changes in pK_a values of ionizable residues in the post-processing of MD trajectories.

Information from TrIPP can be used to guide protein modeling, especially when a pH sensing behaviour is known or suspected (Matsingos *et al.* 2024). It is important to highlight that TrIPP is not meant to replace in any way constant-pH simulations (Williams *et al.* 2010, Oliveira *et al.* 2022), where the protonation state of ionizable residues is allowed to change. Here, we have shown how TrIPP can be used in conjunction with conventional MD simulations. While the sampled frames will reflect fixed protonation states, detection of pK_a changes during the trajectory can highlight residues that might require further simulations with alternative protonation states. Observing large correlations between residues pK_a and global motions might suggest that pH modulation of protein conformation is involved, warranting further investigation. Moreover, clustering and pseudo-mutation analyses can be used to build testable hypotheses about the molecular mechanisms underlying pH-dependent structural and functional changes.

Note on related software. An open-source Python implementation for running PROPKA across MD trajectory frames is also available (Irfan Dotson *et al.* 2020). Complementary features of TrIPP are support for parallel execution and for an alternative pK_a prediction engine (pKAI), residue-name preprocessing that allows for ligand containing systems and an extensive downstream analysis framework including pseudo-mutations and clustering of pK_a-related features as described in this contribution.

Acknowledgements

We would like to thank Yu-Yuan Yang and Dr Alessandro Pandini for testing and providing feedback on earlier versions of TrIPP.

Author contributions

Christos Matsingos (Conceptualization [lead], Data curation [supporting], Formal analysis [equal], Investigation [equal], Methodology [lead], Project administration [supporting], Software [equal], Validation [equal], Visualization [equal], Writing—original draft [equal], Writing—review & editing [equal]), Ka Fu Man (Conceptualization [equal], Data curation [lead], Formal analysis [equal], Investigation [equal], Methodology [equal], Project administration [equal], Software [lead], Validation [equal], Visualization [equal], Writing—original draft [equal], Writing—review & editing [equal]), and Arianna Fornili (Conceptualization [equal], Funding acquisition [lead], Investigation [equal], Methodology [equal], Project administration [equal], Supervision [lead], Writing—original draft [lead], Writing—review & editing [equal])

Supplementary material

Supplementary material is available at *Bioinformatics* online.

Conflicts of interest

None declared.

Funding

This work was supported by the Biotechnology and Biological Sciences Research Council [BB/M009513/1] and the Engineering and Physical Sciences Research Council [EP/T518086/1], and made use of time on HPC granted via the UK High-End Computing Consortium for Biomolecular Simulation, HECBioSim (http://hecbiosim.ac.uk), supported by the Engineering and Physical Sciences Research Council [EP/X035603/1].

Data availability

TrIPP source code and tutorial files are available at https://github.com/fornililab/TrIPP.

References

Alexov E, Mehler EL, Baker N *et al.* Progress in the prediction of p K_a values in proteins. *Proteins* 2011;**79**:3260–75.

Dotson D, Alibay I, Sexton R *et al.* (2020). Becksteinlab/propkatraj: 1.1.x. Zenodo. https://doi.org/10.5281/zenodo.3228425

Anandakrishnan R, Aguilar B, Onufriev AV. H++ 3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulations. *Nucleic Acids Res* 2012;**40**:W537–41.

Brooks B, Karplus M. Normal modes for specific motions of macromolecules: application to the hinge-bending mode of lysozyme. *Proc Natl Acad Sci USA* 1985;**82**:4995–9.

Campello RJGB, Moulavi D, Sander J *et al.* Density-based clustering based on hierarchical density estimates. In: Pei J, Tseng VS, Cao L (eds.), *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer, 2013, 160–172.

Di Russo NV, Estrin DA, Martí MA *et al.* pH-dependent conformational changes in proteins and their effect on experimental pKas: the case of nitrophorin 4. *PLoS Comput Biol* 2012;**8**:e1002761.

Ester M, Kriegel H-P, Sander J *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*. Portland, Oregon: AAAI Press, 1996, 226–31.

Gokcan H, Isayev O. Prediction of protein p K_a with representation learning. *Chem Sci* 2022;**13**:2462–74.

Gordon JC, Myers JB, Folta T *et al.* H++: a server for estimating p Ka s and adding missing hydrogens to macromolecules. *Nucleic Acids Res* 2005;**33**:W368–71.

Gowers RJ, Linke M, Barnoud J *et al.* MDAnalysis: A Python package for the rapid analysis of molecular dynamics simulations. In S. Benthall and S. Rostrup (eds), *Proceedings of the 15th Python in Science Conference*, Austin, TX: SciPy, 2016, 98–105. https://doi.org/10.25080/majora-629e541a-00e

Harris CR, Millman KJ, Van Der Walt SJ *et al.* Array programming with NumPy. *Nature* 2020;**585**:357–62.

Kaufman L. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: Wiley, 2005.

Lee C, Yashiro S, Dotson DL *et al.* Crystal structure of the sodium–proton antiporter NhaA dimer and new mechanistic insights. *J Gen Physiol* 2014;**144**:529–44.

Matsingos C, Howell LA, McCormick PJ *et al.* Elucidating the activation mechanism of the proton-sensing GPR68 receptor. *J Mol Biol* 2024;**436**:168688.

McKinney W. *Data Structures for Statistical Computing in Python*. Austin, Texas: SciPy, 2010, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Michaud-Agrawal N, Denning EJ, Woolf TB *et al.* MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. *J Comput Chem* 2011;**32**:2319–27.

Micheletti C, Seno F, Maritan A. Recurrent oligomers in proteins: an optimal scheme reconciling accurate and concise backbone representations in automated folding and design studies. *Proteins* 2000;**40**:662–74.

Oliveira VMd, Liu R, Shen J. Constant pH molecular dynamics simulations: current status and recent applications. *Curr Opin Struct Biol* 2022;**77**:102498.

Olsson MHM, Søndergaard CR, Rostkowski M *et al.* PROPKA3: consistent treatment of internal and surface residues in empirical pK_a predictions. *J Chem Theory Comput* 2011;**7**:525–37.

Pedregosa F, Varoquaux G, Gramfort A *et al.* Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**:2825–30.

Rabenstein B, Knapp E-W. Calculated pH-dependent population and protonation of carbon-monoxy-myoglobin conformers. *Biophys J* 2001;**80**:1141–50.

Ramos J, Laux V, Mason SA *et al.* Structure and dynamics of the active site of hen egg-white lysozyme from atomic resolution neutron crystallography. *Structure* 2025;**33**:136–48.e3.

Reis PBPS, Bertolini M, Montanari F *et al.* A fast and interpretable deep learning approach for accurate electrostatics-driven p K_a predictions in proteins. *J Chem Theory Comput* 2022;**18**:5068–78.

Reis PBPS, Vila-Viçosa D, Rocchia W *et al.* PypKa: a flexible Python module for Poisson–Boltzmann-based pKa calculations. *J Chem Inf Model* 2020;**60**:4442–8.

Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.

Schönichen A, Webb BA, Jacobson MP *et al.* Considering protonation as a posttranslational modification regulating protein structure and function. *Annu Rev Biophys* 2013;**42**:289–314.

Virtanen P, Gommers R, Oliphant TE *et al.*; SciPy 1.0 Contributors. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* 2020;**17**:261–72.

Warwicker J. The physical basis for pH sensitivity in biomolecular structure and function, with application to the spike protein of SARS-CoV-2. *Front Mol Biosci* 2022;**9**:834011.

Waskom M. seaborn: statistical data visualization. *JOSS* 2021;**6**:3021.

Webb H, Tynan-Connolly BM, Lee GM *et al.* Remeasuring HEWL pK_a values by NMR spectroscopy: methods, analysis, accuracy, and implications for theoretical pK_a calculations. *Proteins* 2011;**79**:685–702.

Wei W, Hogues H, Sulea T. Comparative performance of high-throughput methods for protein p K_a predictions. *J Chem Inf Model* 2023;**63**:5169–81.

Williams SL, De Oliveira CAF, McCammon JA. Coupling constant pH molecular dynamics with accelerated molecular dynamics. *J Chem Theory Comput* 2010;**6**:560–8.

Bioinformatics, 2026, 42, 1–5
https://doi.org/10.1093/bioinformatics/btag063
Applications Note