UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS

A new generation of user-friendly and machine learning-accelerated methods for protein pK_a calculations

"Documento Definitivo"

Doutoramento em Bioquímica
Biofísica Molecular

Pedro de Brito Pires Santos Reis

Tese orientada por:
Doutor Miguel Ângelo dos Santos Machuqueiro
Doutor Diogo Ruivo dos Santos Vila Viçosa
Doutor Walter Rocchia

Documento especialmente elaborado para a obtenção do grau de doutor

2022

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS

A new generation of user-friendly and machine learning-accelerated methods for protein pK_a calculations

Doutoramento em Bioquímica
Biofísica Molecular

Pedro de Brito Pires Santos Reis

Tese orientada por:
Doutor Miguel Ângelo dos Santos Machuqueiro
Doutor Diogo Ruivo dos Santos Vila Viçosa
Doutor Walter Rocchia

Júri:
Presidente:

- Doutor Manuel Eduardo Ribeiro Minas da Piedade, Professor Catedrático, Presidente do Departamento de Química e Bioquímica da Faculdade de Ciências da Universidade de Lisboa

Vogais:

- Doutor Adrian E. Roitberg, Full Professor do Department of Chemistry College of Liberal Arts and Sciences da University of Florida
- Doutor Paulo José Garcia de Lemos Trigueiros de Martel, Professor Auxiliar Faculdade de Ciências e Tecnologia da Universidade do Algarve
- Doutor Francisco Jorge Dias Oliveira Fernandes, Investigador Júnior, INESC – ID (Instituto de Engenharia de Sistemas e Computadores: Investigação e Desenvolvimento em Lisboa)
- Doutor António Eduardo Nascimento Ferreira, Professor Auxiliar com Agregação da Faculdade de Ciências da Universidade de Lisboa
- Doutor Miguel Ângelo dos Santos Machuqueiro, Investigador Auxiliar (Emprego Científico Individual 2017) Faculdade de Ciências da Universidade de Lisboa (Orientador)

Documento especialmente elaborado para a obtenção do grau de doutor

Esta tese foi apoiada financeiramente pela FCT através da bolsa SFRH/BD/136226/2018

2022

Acknowledgments

What an incredible journey these last four years have been! On the scientific level, I explored numerous fascinating topics that kept fueling my curiosity. On a personal level, I crossed paths with so many impressive and unique people who shaped my mind and warmed my heart. This period of my life was a notoriously enriching experience of which I am both grateful and proud. The work developed during this time is epitomized in this thesis. Naturally, none of the projects would have been possible without a great deal of support from a lot of people.

First and foremost, I must thank Miguel Machuqueiro. More than a supervisor, Miguel is a mentor and a friend that I admire as a scientist and as a person. I will forever be thankful to him for all he has taught me since joining his group in 2016. Unlike so many students I've met, I couldn't have asked for a better supervisor. Miguel is always eager to discuss science, ready to work with you on tight deadlines while shooting up dad jokes and having a laugh. Miguel imposes strict scientific quality control on everyone, yet, he allows his students to find their space and delve into their own research interests. Working with Miguel is an absolute pleasure; thus, it is no surprise that he has created a wonderful group filled with bright and motivated people. Diogo Vila-Viçosa, like myself, was a Master's and Ph.D. student at the Molecular Modeling and Simulation group. In fact, when I met my now supervisor Diogo, he was doing his Ph.D., and he has been someone I look up to ever since. Learning from Diogo is a delight as he explains himself very clearly with simple yet effective drawings and relevant examples. Besides the support and the enthusiastic sharing of his knowledge, I thank him for being my accomplice in getting into machine learning. Other than Miguel and Diogo, I received help from many others at the University of Lisbon. For the help and for making lunch, coffee breaks, and overall time fun, I also have to thank, among others, Paulo Costa, Bruno Victor, Pedro Magalhães, and Rafael Nunes. On a special note, I'd like to thank Tomás Silva for being a pal and sharing a significant portion of my student life and respective joys and pains since we started studying biochemistry at the bachelor's level.

Before starting the Ph.D., I had the opportunity to spend a few months at the CONCEPT lab of Istituto Italiano di Tecnologia. During the Erasmus period, it became clear that it would be a shame not to continue learning at Walter Rocchia's group. Discussing with Walter is always stimulating, and he often challenges me with problems outside my comfort zone. Walter genuinely cares about his students and makes a great effort to constantly advance their knowledge. I am very grateful to him for pushing me to look deeper, not to be satisfied with superficial understanding, and to become a more well-rounded scientist. Walter's qualities as a supervisor are reflected in the people of his group, a collection of highly skilled and friendly individuals. I'd like to highlight Miguel Soler and Nicola Scafuri, not only for the exciting conversations but also for the fraternity and well-spent time-off together with Artemi Bendandi, Patricio Barletta, and Sara Fortuna.

A substantial part of this Ph.D. thesis was not planned from the beginning. Moreover, if not for a fortuitous encounter with Floriane Montanari the projects related to machine learning would likely not have happened. Luckily, I did attend her lovely talk about using graph neural networks for toxicity prediction of small molecules at Vienna's summer school on drug design. At the Machine Learning Research group, I found a collaborative environment filled with talented researchers with diverse backgrounds. I was incredibly well received at Djork-Arné "Okko" Clevert's group. Okko has an inspiring vision of open science and fosters interdisciplinary cooperation as a catalyst for groundbreaking research. Despite the Covid outbreak severely limiting our ability to interact in person, it was still a life-changing learning experience that forged plenty of meaningful relationships over scientific discussions and the occasional beer with Okko, Floriane, Santiago Villalba, Marco Bertolini, Tuan Le, Vicenzo Palmacci, and many others.

I would also like to thank the members of my doctoral supervisory committee – Paulo Martel and António Ferreira – for their guidance and insightful feedback. I extend the acknowledgment to António Baptista for countless book recommendations and didactic clarifications.

On a more personal note, I thank my family, and in particular my mom, my dad, Minhaia and Zé, for all the unconditional love and support throughout my life and for always encouraging me to pursue my passions. Another special acknowledgment must be made to Joana Fernandes for being my home, believing in me even more than I do, and putting up with the distance and the stress of the mobility periods. In Manesseno, I was fortunate to be gifted a one-of-a-kind Italian family – i Pastorini. *Elena e Paolo, vi sarò eternamente grato per avermi accolto nella vostra bellissima famiglia e per avermi fatto sempre sentire a casa.*

Last but not least, a word of appreciation to Fundação para a Ciência e a Tecnologia for funding this Ph.D. plan through grant SFRH/BD/136226/2018, and to BioISI – Biosystems & Integrative Sciences Institute at the Faculty of Sciences, University of Lisbon, as well as Istituto Italiano di Tecnologia for hosting this work.

Preface

During this Ph.D. thesis, I had the privilege to work with and learn from several inspiring researchers and research groups. Consequently, I was given the opportunity to gain hands-on experience in a diverse set of computational methodologies and biological systems. The four years-long learning journey has been encapsulated in the present document. This highly interdisciplinary Ph.D. thesis is a selection of projects devised with a clear goal in mind: creating the next generation of fast and user-friendly pK_a calculations. It was only possible to pursue this endeavor due to a notable ensemble of research groups with complementary expertise and backgrounds: the *Molecular Modeling and Simulation* lab at the University of Lisbon led by Miguel Machuqueiro, with vast know-how in developing rigid body pK_a calculations and constant-pH molecular dynamics (CpHMD); Walter Rocchia's *CONCEPT* lab at Istituto Italiano di Tecnologia is a knowledge hub in biophysical systems and electrostatics; the *Machine Learning Research* group at Bayer headed by Djork-Arné Clevert has extensive experience in applying and developing deep learning models and methods. Evidently, the resulting work lies at the interface of biochemistry, biophysics, and data science.

The initial step towards our goal was given by the development of PypKa (Chapter 2.1), a flexible tool to predict physics-based pK_a values of titratable sites in proteins:

[1] P. B. P. S. Reis, D. Vila-Viçosa, W. Rocchia*, M. Machuqueiro*. PypKa: A Flexible Python Module for Poisson–Boltzmann-Based pK_a Calculations. Journal of Chemical Information and Modeling, 60 (10), 4442-4448, 2020

Even though it was the first tool to be released, we continued adding features to PypKa over the years. For example, it now estimates the isoelectric point and supports membrane and nucleic acids in addition to solvated proteins, and it is possible to use CHARMM-derived radii and charges. Given the ease of running Poisson–Boltzmann-based pK_a calculations provided by its command-line interface (CLI) and application programming interface (API), PypKa became the foundational work most of the remaining projects shown in this thesis have been built upon. Furthermore, the advantage of having developed the different projects in a modular way is that the improvements done in one package cascade into the others. The original motivation to develop PypKa was to leverage it to abstract Poisson–Boltzmann and Monte Carlo simulations in a novel implementation of the stochastic titration method, PypKa-MD (Chapter 2.2). Although PypKa-MD has not yet been published, we have performed a preliminary validation.

During the course of this Ph.D., the artificial intelligence revolution started to spread to the biological sciences, and we noticed an opportunity to drastically accelerate our methods. Since we wanted to train deep learning models to predict pK_a values and the amount of experimental data available was limited, we first created a large database of pK_a values, pKPDB (Chapter 4.1), by running PypKa on thousands on proteins from the Protein Data Bank:

[2] P. B. P. S. Reis*, D. A. Clevert, M. Machuqueiro*. pKPDB: a protein data bank extension database of pK_a and pI theoretical values. Bioinformatics, 1 (1), 297–298, 2022

To make pK_a predictions readily available, we have built a web app that allows users to query pKPDB, as well as launch PypKa runs (Chapter 4.2). We have also added support to a common use case of pK_a calculations: the preparation of biomolecular structures for molecular dynamics simulations.

The pKPDB database was then used as training data for pKAI, a deep learning model that predicts pK_a values up to 1000× faster than PypKa and with comparable accuracy (Chapter 3.1):

[3] P. B. P. S. Reis*, M. Bertolini, F. Montanari, W. Rocchia, M. Machuqueiro*, and D. A. Clevert*. A Fast and Interpretable Deep Learning Approach for Accurate Electrostatics-Driven pK_a Predictions in Proteins. Journal of Chemical Theory and Computation, 2022

An akin acceleration could benefit CpHMD making it less computationally expensive, one of its main drawbacks. However, before pKAI could be used in a CpHMD setting, a few limitations needed to be fixed. For example, sampling protonation states from pK_a values is an extreme approximation that leads to incorrect protonation and, consequently, conformational ensembles. Thus, we have trained a new model that improves on pKAI's shortcomings and can be used to replace PypKa within the CpHMD framework. The AI-accelerated CpHMD, pKAI-MD, is still currently being developed. Nevertheless, encouraging preliminary results are shared in Chapter 3.2.

Complementary to the work done within my Ph.D. plan, I have contributed to several other projects during the last four years. Some of these projects had strong synergy with the work presented in this thesis. One of which aimed at increasing the computational efficiency of CpHMD with enhanced sampling:

[4] D. Vila-Viçosa*, P. B. P. S. Reis, A. M. Baptista, C. Oostenbrink, M. Machuqueiro. A pH replica exchange scheme in the stochastic titration constant-pH MD method.

Journal of Chemical Theory and Computation, 15 (5), 3108-3116, 2019

Unfortunately, GROMACS does not support the charge groups needed for the reaction field method. Hence, the validation of GROMOS with particle mesh Ewald, instead of reaction field with which it was parameterized was also important for accelerating CpHMD as it allowed the MD simulations to be ran in the GPU:

[5] T. F. D. Silva, D. Vila-Viçosa, P. B. P. S. Reis, B. L. Victor, M. Diem, C. Oostenbrink, M. Machuqueiro*. The impact of using single atomistic long range cutoff schemes with the GROMOS 54A7 force field. Journal of Chemical Theory and Computation, 14(11): 5823-5833, 2018

The extension of CpHMD to CHARMM is another example of a project quite aligned with the objectives of my Ph.D. as supporting a popular force field contributes to the adoption of CpHMD:

[6] J. G. N. Sequeira, F. E. P. Rodrigues, T. G. D. Silva, P. B. P. S. Reis*, M. Machuqueiro*. Extending the stochastic titration CpHMD to CHARMM36m. Journal of Physical Chemistry B. *In review*

I also contributed to two book chapters about computational methods for studying membrane systems. In the first one, we describe the linear response approximation, an alternative to CpHMD, to estimate pK_a values of membrane proteins from MD simulations:

[7] N. F. B. Oliveira, T. F. D. Silva, P. B. P. S. Reis, M. Machuqueiro*. pK_a Calculations in Membrane Proteins from Molecular Dynamics Simulations. Computational Design of Membrane Proteins. Methods in Molecular Biology, 2315, 185–195, 2021

In the second book chapter, we characterize a protocol to identify membrane disruptive small molecules, PAINS, using umbrella sampling:

[8] P. R. Magalhães, P. B. P. S. Reis, D. Vila-Viçosa, M. Machuqueiro, B. L. Victor*. Identification of Pan-Assay INterference compoundS (PAINS) Using an MD-Based Protocol. Computational Design of Membrane Proteins. Methods in Molecular Biology, 2315, 263-271, 2021

Later, we adapted the PAINS identification protocol to be more efficient:

[9] P. R. Magalhães, P. B. P. S. Reis, D. Vila-Viçosa, M. Machuqueiro*, B. L. Victor*. Optimization of an in Silico Protocol Using Probe Permeabilities to Identify Membrane

Pan-Assay Interference Compounds. Journal of Chemical Information and Modeling, 62 (12), 3034-3042, 2022

In collaboration with experimentalists, we performed molecular docking simulations of a biologically relevant protein-ligand system to explain the observed bioactivity:

[10] L. Guedes, P. B. P. S. Reis, M. Machuqueiro, A. Ressaissi, R. Pacheco, M. L. Serralheiro*. Bioactivities of Centaurium erythraea (Gentianaceae) Decoctions: Antioxidant Activity, Enzyme Inhibition and Docking Studies. Molecules, 24 (20), 3795, 2019

Finally, I got to work on the characterization of a particularly significant type of protein-protein interaction, the antibody-antigen interface:

[11] P. B. P. S. Reis, G. P. Barletta, L. Gagliardi, S. Fortuna, M. A. Soler*, W. Rocchia*. Antibody-Antigen Binding Interface Analysis in the Big Data Era. Frontiers in Molecular Biosciences, 9, 2022

Abstract

The ability to sense and react to external and internal pH changes is a survival requirement for any cell. pH homeostasis is tightly regulated, and even minor disruptions can severely impact cell metabolism, function, and survival. The pH dependence of proteins can be attributed to only 7 out of the 20 canonical amino acids, the titratable amino acids that can exchange protons with water in the usual 0-14 pH range. These amino acids make up for approximately 31% of all amino acids in the human proteome, meaning that, on average, roughly one-third of each protein is sensitive not only to the medium pH but also to alterations in the electrostatics of its surroundings. Unsurprisingly, protonation switches have been associated with a wide array of protein behaviors, including modulating the binding affinity in protein-protein, protein-ligand, or protein-lipid systems, modifying enzymatic activity and function, and even altering their stability and subcellular location. Despite its importance, our molecular understanding of pH-dependent effects in proteins and other biomolecules is still very limited, particularly in big macromolecular complexes such as protein-protein or membrane protein systems.

Over the years, several classes of methods have been developed to provide molecular insights into the protonation preference and dependence of biomolecules. Empirical methods offer cheap and competitive predictions for time- or resource-constrained situations. Albeit more computationally expensive, continuum electrostatics-based are a cost-effective solution for estimating microscopic equilibrium constants, pK_{half} and macroscopic pK_{a}. To study pH-dependent conformational transitions, constant-pH molecular dynamics (CpHMD) is the appropriate methodology. Unfortunately, given the computational cost and, in many cases, the difficulty associated with using CE-based and CpHMD, most researchers overuse empirical methods or neglect the effect of pH in their studies. Here, we address these issues by proposing multiple pK_{a} predictor methods and tools with different levels of theory designed to be faster and accessible to more users.

First, we introduced PypKa, a flexible tool to predict Poisson–Boltzmann/Monte Carlo-based (PB/MC) pK_{a} values of titratable sites in proteins. It was validated with a large set of experimental values exhibiting a competitive performance. PypKa supports CPU parallel computing and can be used directly on proteins obtained from the Protein Data Bank (PDB) repository or molecular dynamics (MD) simulations. A simple, reusable, and extensible Python API is provided, allowing pK_{a} calculations to be easily added to existing protocols with a few extra lines of code. This capability was exploited in the development of PypKa-MD, an easy-to-use implementation of the stochastic titration CpHMD method. PypKa-MD supports GROMOS and CHARMM force fields, as well as modern versions of GROMACS. Using PypKa's API and consequent abstraction of PB/MC contributed to its greatly simplified modular architecture that will serve as the foundation for future developments. The new implementation was validated on alanine-based tetrapeptides with closely interacting titratable residues and four commonly used benchmark proteins, displaying highly similar and correlated pK_a predictions compared to a previously validated implementation.

Like most structural-based computational studies, the majority of pK_a calculations are performed on experimental structures deposited in the PDB. Furthermore, there is an ever-growing imbalance between scarce experimental pK_a values and the increasingly higher number of resolved structures. To save countless hours and resources that would be spent on repeated calculations, we have released pKPDB, a database of over 12M theoretical pK_a values obtained by running PypKa over 120k protein structures from the PDB. The precomputed pK_a estimations can be retrieved instantaneously via our web application, the PypKa Server. In case the protein of interest is not in the pKPDB, the user may easily run PypKa in the cloud either by uploading a custom structure or submitting an identifier code from the PBD or UniProtKB. It is also possible to use the server to get structures with representative pH-dependent protonation states to be used in other computational methods such as molecular dynamics.

The advent of artificial intelligence in biological sciences presented an opportunity to drastically accelerate pK_a predictors using our previously generated database of pK_a values. With pKAI, we introduced the first deep learning-based predictor of pK_a shifts in proteins trained on continuum electrostatics data. By combining a reasonable understanding of the underlying physics, an accuracy comparable to that of physics-based methods, and inference time speedups of more than 1000 ×, pKAI provided a game-changing solution for fast estimations of macroscopic pK_a from ensembles of microscopic values. However, several limitations needed to be addressed before its integration within the CpHMD framework as a replacement for PypKa. Hence, we proposed a new graph neural network for protein pK_a predictions suitable for CpHMD, pKAI-MD. This model estimates pH-independent energies to be used in a Monte Carlo routine to sample representative microscopic protonation states. While developing the new model, we explored different graph representations of proteins using multiple electrostatics-driven properties.

While there are certainly many new features to be introduced and a multitude of development to be expanded, the selection of methods and tools presented in this work poses a significant improvement over the alternatives and effectively constitutes a new generation of user-friendly and machine learning-accelerated methods for pK_a calculations.

Keywords: pK_a, protonation, constant-pH, machine learning

Resumo

A capacidade de deteção e reação a alterações externas e internas do pH é um requisito de sobrevivência para qualquer célula. A homeostase do pH é rigorosamente regulada, e mesmo pequenas perturbações podem afetar gravemente o metabolismo, função e sobrevivência das células. Pode atribuir-se a dependência das proteínas em relação ao pH a apenas a 7 dos 20 aminoácidos canónicos. A estes resíduos chamamos de aminoácidos tituláveis e possuem a habilidade única de poder trocar protões com a água a valores de pH entre 0 e 14. Estes aminoácidos constituem cerca de 31% de todos os aminoácidos do proteoma humano, o que significa que, em média, cerca de um terço de cada proteína é sensível não só ao pH do meio mas também a alterações na eletrostática do seu ambiente. Devido às suas propriedades, os resíduos tituláveis estão intimamente relacionados a inúmeros fenómenos vitais de proteínas, incluindo a modulação da afinidade de ligação nos sistemas proteína-proteína, proteína-ligando, ou proteína-lípido, modificando a atividade e função enzimática, e até alterando a sua estabilidade e localização subcelular. Apesar da sua importância, a nossa compreensão molecular dos efeitos dependentes do pH em proteínas e outras biomoléculas é ainda muito limitada, particularmente em grandes complexos macromoleculares, tais como sistemas proteína-proteína ou proteínas de membrana.

Ao longo dos anos foram desenvolvidas várias classes de métodos que oferecem detalhes moleculares sobre a preferência e dependência da protonação em biomoléculas. Os métodos empíricos oferecem previsões rápidas para situações com limitações de tempo ou de recursos. Embora computacionalmente mais caros, os métodos baseados em eletrostática de contínuo (CE) são uma solução eficiente para estimar constantes de equilíbrio microscópico, pK_{half} e pK_{a} macroscópicos. Para estudar as transições conformacionais dependentes do pH, a dinâmica molecular a pH constante (CpHMD) é a metodologia apropriada. Infelizmente, dado o custo computacional e, em muitos casos, a dificuldade associada à utilização de CE e CpHMD, a maioria dos investigadores utiliza apenas métodos empíricos ou acaba por negligenciar o efeito do pH nos seus estudos. Neste trabalho, abordamos estas limitações propondo múltiplos métodos e ferramentas de previsão de pK_{a} com diferentes níveis de teoria concebidos para serem mais rápidos e acessíveis.

Inicialmente introduzimos o PypKa, uma ferramenta flexível para prever valores de pK_{a} em proteínas, baseada em cálculos de Poisson–Boltzmann (PB) e Monte Carlo (MC). O PypKa foi validado num grande conjunto de valores experimentais, e demonstrou um desempenho competitivo. Diversas partes deste software são paralelizáveis e pode ser corrido diretamente em proteínas obtidas do repositório do Protein Data Bank (PDB) ou de simulações de dinâmica molecular (MD). É possível interagir com o PypKa através de uma API simples, reutilizável e extensível, que permite adicionar a capacidade de correr cálculos pK_{a} a outros protocolos apenas com a introdução de algumas linhas extra de código. Esta capacidade foi explorada no

desenvolvimento do PypKa-MD, uma implementação do método CpHMD de titulação estocástica focada na usabilidade. O PypKa-MD suporta os campos de força GROMOS e CHARMM, bem como versões modernas do GROMACS. A utilização da API do PypKa e consequente abstração do PB/MC contribuíram para a sua arquitetura modular simplificada que servirá de base para futuros desenvolvimentos. A nova implementação foi validada em tetrapéptidos de alanina com resíduos tituláveis em estreita interação, e em quatro proteínas de referência, apresentando previsões de pK_a altamente semelhantes e correlacionadas com uma outra implementação previamente validada.

Tal como na maioria dos estudos computacionais de base estrutural, também uma parte significativa de cálculos pK_a são efetuados em estruturas experimentais depositadas no PDB. Além disso, existe um desequilíbrio cada vez maior entre os escassos valores experimentais de pK_a e o número cada vez mais elevado de estruturas resolvidas. Para poupar inúmeras horas e recursos que seriam gastos em cálculos repetidos, criámos o pKPDB, uma base de dados com mais de 12 milhões de valores teóricos obtidos através da execução do PypKa em mais de 120 mil estruturas do PDB. As estimativas pré-calculadas de pK_a podem ser obtidas instantaneamente através da nossa aplicação web, o PypKa Server. Caso a proteína de interesse não esteja no pKPDB, o utilizador pode facilmente executar o PypKa na nuvem, quer carregando uma estrutura personalizada ou submetendo um código identificador do PBD, ou UniProtKB. É também possível utilizar o servidor para obter estruturas com estados de protonação representativos a um certo pH para serem posteriormente utilizadas noutros métodos computacionais, tais como a dinâmica molecular.

O advento da inteligência artificial nas ciências biológicas apresentou uma oportunidade de acelerar drasticamente os preditores de pK_a utilizando a nossa base de dados anteriormente gerada. Com o pKAI, introduzimos o primeiro de *deep learning* para prever pK_a em proteínas treinado em exemplos de valores estimados por CE. Ao combinar uma precisão comparável à dos métodos baseados em CE, e uma acerelação de tempo de inferência superior a 1000 vezes, o pKAI apresenta-se como uma solução competitiva para estimativas rápidas de p$K_\mathrm{a}$macroscópicos a partir de conjuntos de cálculos microscópicos. No entanto, existem várias limitações que necessitam ser abordadas antes da sua integração no CpHMD como substituto do PypKa. Assim, propusemos uma nova rede neural baseada em grafo para prever pK_a em proteínas que é adequada para ser usada no contexto do CpHMD, o pKAI-MD. Este modelo estima as contribuições necessárias para ser utilizadas numa rotina de Monte Carlo por forma a amostrar estados representativos de protonação. Ao desenvolver o novo modelo, explorámos diferentes formas de representar proteínas como grafos. Para tal fim, otimizámos o modelo para estimar múltiplas propriedades eletrostáticas.

Embora existam certamente muitas características novas a serem introduzidas e um universo de desenvolvimentos para serem expandidos, a seleção de métodos e ferramentas apresentadas

neste trabalho representa uma melhoria significativa em relação às alternativas e constitui efetivamente uma nova geração de métodos para estimar valores de pK_a mais rápidos e mais fáceis de utilizar.

Palavras-chave: pK_a, protonação, pH constante, machine learning

Contents

List of Figures

List of Tables

List of Abbreviations

Å Angstrom

AD Alzheimer’s disease

AI Artificial intelligence

API Application Programming Interface

CE Continuum Electrostatics

CG Coarse-Grained

CLI Command-line Interface

CpHMD Constant-pH Molecular Dynamics

CPU Central Process Unit

FEP Free Energy Perturbation

FNN Feedforward Neural Network

GB Generalized Born

GNN Graph Neural Network

GPU Graphics Processing Unit

HH Henderson–Hasselbalch

ITC Isothermal titration microcalorimetry

LJ Lennard–Jones

LRA Linear Response Approximation

MAE Mean Absolute Error

MC Monte Carlo

MD Molecular Dynamics

ML Machine Learning

MM Molecular Mechanics

MSE Mean-Square Error

NMR Nuclear magnetic resonance

PB Poisson–Boltzmann

PDB Protein Data Bank

QM Quantum Mechanics

RMSE Root-Mean-Square Error

Chapter 1

Introduction & State of the Art

1.1 The importance of pH in biological systems

Nowadays, pH is a mainstream concept, present in many aspects of our modern life. pH tests are standard for food and water quality control, while blood or urine acidity levels are standard health checks. It is an essential property of a solution, and many chemical and biochemical processes, such as fermentation, are pH-dependent. The roughly hundred-year-old term was coined by Sørensen, working on brewing processes for the Danish beer company Carlsberg [12]. The pH scale was invented as a convenient way of expressing hydrogen ion (H^+) concentration spanning several orders of magnitude:

$$10^{-\mathrm{p}H} = [H^+] \tag{1.1}$$

In this base–10 logarithmic scale, each unit decrease corresponds to a ten-fold increase in acidity. A slight rearrangement gives the more familiar equation:

$$\mathrm{p}H = -\log[H^+] \tag{1.2}$$

By defining pH in this way, one can easily refer to H^+ concentrations ranging between 10^0 mol dm^{-3} and 10^{-14} mol dm^{-3} as pH values from 0 to 14, respectively. This range is not random and stems from the equilibrium constant for the autoionization of water (K_w) at 298K:

$$K_w = [H^+][OH^-] = 10^{-14} \tag{1.3}$$

At this temperature in pure water, the identical concentration of H^+ and OH^- is of 10^{-7} M. By taking the negative logarithm of each term, we obtain:

$$\mathrm{p}K_w = \mathrm{p}H + \mathrm{p}OH = 14 \tag{1.4}$$

The pH scale is most useful for diluted solutions, although negative and greater than 14 values are allowed for strong acids and bases, respectively. Furthermore, like all equilibrium constants, K_w is temperature-dependent, and pure neutral water can have a pH value different from 7. At 310K, a typical temperature of the human body, pK_w equals 13.6, and neutrality occurs at a pH of 6.8 [13].

The addition of weak acids and/or bases to a solution can change its pH value. Biological solutions are complex mixtures, and their acidity is dictated by a balance of countless deprotonation and protonation reactions. Each of these exchanges of H^+ with the solvent can be written as an acid-base reaction:

$$\mathrm{HA} + \mathrm{H_2O} \rightleftharpoons \mathrm{A}^- + \mathrm{H_3O}^+ \tag{1.5}$$

In biological systems, it is vital to keep the pH within a narrow range. For instance, the slightly alkaline human blood may only fluctuate between pH 7.35 and 7.45, as many biological processes rely on it. Buffers can be used to prevent severe pH variations when H^+ or OH^- are added or removed. Buffers consist of a mixture of a weak acid and its conjugate base, or vice-versa. The primary buffer system for maintaining the pH homeostasis of blood is the carbonic acid/bicarbonate buffer:

$$2\,H_2O + CO_2 \rightleftharpoons H_2CO_3 + H_2O \rightleftharpoons H_3O^+ + HCO_3^- \tag{1.6}$$

Catalyzed by carbonic anhydrase, the gaseous metabolic byproduct carbon dioxide reacts with water to form carbonic acid, dissociating into bicarbonate and a hydrogen ion. In this system, only the second equilibrium is an acid-base reaction, which can be simplified as an acid ionization equilibrium between the protonated (HA) and deprotonated species (A^-):

$$HA \rightleftharpoons H^+ + A^- \tag{1.7}$$

In the bicarbonate buffer, HA is the weak carbonic acid, and A^- is its conjugated base bicarbonate. If a strong acid or H^+ is added to such a solution, A^- will be consumed to form HA, and the $[H^+]$ will increase less than the amount introduced. The opposite will happen in case a strong base or OH^- is added. The OH^- reaction with H^+ to form water causes the equilibrium to dissociate HA to satisfy the expression:

$$K_a = \frac{[H^+][A^-]}{[HA]} \tag{1.8}$$

For the buffer to work, HA and A^- need to be available in solution to react with the H^+ being added. Thus, the greater the concentration of HA and A^-, the better the solution's buffering capacity. Optimum efficiency is obtained when [HA] equals $[AH^-]$, which corresponds to the case when the pH of the buffer matches the $\mathrm{p}K_\mathrm{a}$ of the weak acid, as given by the Henderson–Hasselbalch (HH) equation:

$$\mathrm{p}K_a = \mathrm{p}H - \log\frac{[A^-]}{[HA]} \tag{1.9}$$

At normal body temperature, carbonic acid has a $\mathrm{p}K_\mathrm{a}$ of 6.1. This means that the normal blood pH is outside the range in which the buffer is efficient, typically $\mathrm{p}K_\mathrm{a}$ ±1 pH unit. In order to be suitable for maintaining the pH of the blood at 7.4, the bicarbonate to carbonic acid ratio must be shifted to approximately 20:1. In our blood, the $[HCO_3{}^-]$ to $[CO_2]$ ratio is kept, being ~25 mM and ~1.23 mM, respectively [14], since carbonic acid is not stable in aqueous solutions and its conversion into carbon dioxide by carbonic anhydrase is very fast (at a rate 25000-fold faster than the uncatalyzed reaction[15]). The significantly larger concentrations of bicarbonate and carbon dioxide compared to that of H^+ ($10^{-7.4}$ M) allow for this ratio to remain relatively constant. However, in certain conditions, such as vigorous physical exercise, the buffer effect alone might not be enough to maintain blood pH in the desired range. Fortunately, two organs are orchestrating this buffer system: the lungs adjusting the $[CO_2]$, and the kidneys regulating the $[HCO_3{}^-]$. By accelerating breathing activity, the lungs are fast to counteract the pH-lowering effects by removing carbon dioxide. On the other hand, the kidneys slow down the excretion of bicarbonate into urine and its reabsorption. Some proteins are also essential for maintaining this buffering ability since they play a central role in pH homeostasis. In fact, around 20% of the carbon dioxide and 40% of the hydrogen ions formed in the tissues are readily transported to the lungs and kidneys by hemoglobin (Figure 1.1)) [16]. This protein function complements the bicarbonate buffer system, which transports the remainder of H^+ and CO_2. Also, as it is present in red blood cells, it can react with the H^+ product of the carbon dioxide conversion into bicarbonate. Hemoglobin (Hb) is perhaps most known for being an oxygen carrier, and rightfully so as 98% of oxygen present in the blood is bound to these proteins [17]. The oxygen binds to Hb via the iron (Fe) present in the porphyrin ring of the heme group contained in each of the four subunits. Despite the distinct interaction sites of O_2 and H^+, the ability of this tetrameric protein ($\alpha_2\beta_2$) to binding oxygen is intertwined with its buffering function. One could describe the equilibrium between the protonated form of hemoglobin (HbH^+) and the oxygen-binding state oxyhemoglobin (HbO_2) as:

$$HbH^+ + O_2 \rightleftharpoons HbO_2 + H^+ \tag{1.10}$$

As follows, when in high O_2concentration environments like the lungs, the oxygen-bound state is preferred. The opposite happens in the tissues where oxygen concentration is low and deoxyhemoglobin, the more effective buffer, is favored. The conversion between the oxy and deoxy states entails the formation of inter- and intra-subnit salt bridges, which force a more

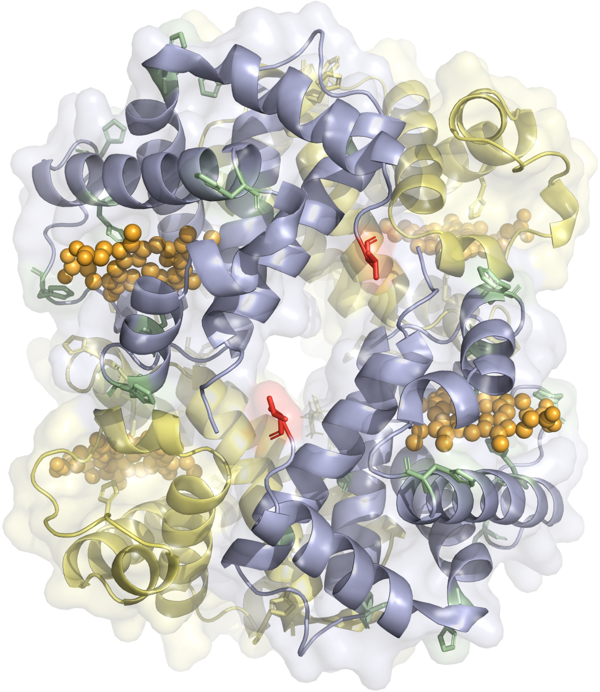


Figure 1.1: Representation of the O_2 (heme groups in orange), CO_2 (N-terminal sites in red), and H^+ (histidine residues in green) binding sites of human deoxyhemoglobin (PDB 1A3N; α chains in blue and β chains in yellow).

constrained configuration (see Figure 1.2). One of these deoxy state stabilizing salt bridges is between β chain's histidine HC3 and aspartate FG1. This interaction favors the protonated histidine, thus increasing its pK_a by 1.5 pK units to 8.0 [18] and beyond the pH of its medium. In the relaxed state, this hydrogen bond does not occur, which keeps the pK_a of this histidine close to 6.5 [19], and thus the deprotonated state becomes predominant. Due to the pK_a value of the imidazole group being close to physiological pH, histidine residues are responsible for the buffering capacity of hemoglobin. Since histidine is present in most proteins, one could say that most proteins, when abundant, can exhibit this buffering effect at physiological pH. Hemoglobin is extremely important for the regulation of the blood's pH since they are very abundant (150 g/L) compared to other plasma proteins (70 g/L) [20]. Furthermore, hemoglobin contains a notably higher relative incidence of histidine residues than most plasma proteins. As an example, 6.6% (38 in 574) of human Hb (65 kDa) residues are His, while this residue type only accounts for 2.5% (30 in 1170) of human serum albumin (133 kDa) or 2.8% (38 in 1352) of human serum transferrin (152 kDa). This enrichment is certainly not fortuitous, and given the function of Hb, it is pretty helpful to sense pH. In peripheral tissues, Hb must release oxygen and bind to carbon dioxide. Conveniently, the deoxy state is more prone to CO_2 binding even though the binding site is different from those of O_2 and H^+ (allosteric effect). The N-terminal of each chain can react with carbon dioxide forming a carbamate group:

$$\text{R-NH}_2 + \text{CO}_2 \rightleftharpoons \text{R-NHCOO}^- + \text{H}^+ \tag{1.11}$$

In this reaction, a hydrogen ion is produced, further contributing to the stability of the deoxy

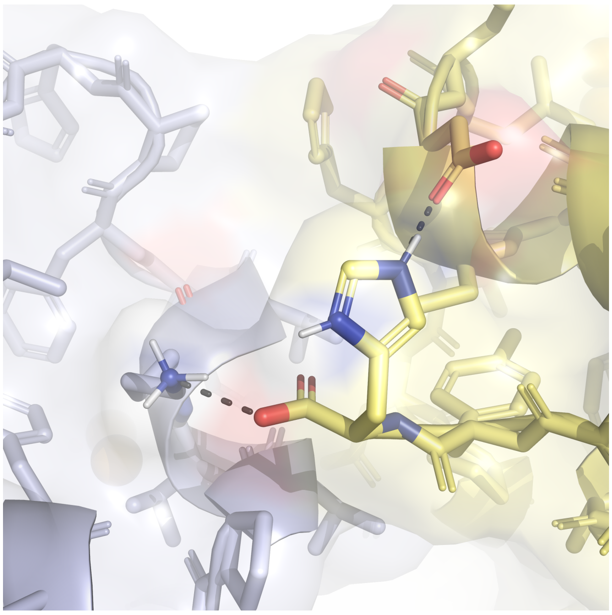


Figure 1.2: Representation of two important salt bridges stabilizing the deoxy state of human hemoglobin (PDB 1A3N; α chains in blue and β chains in yellow). The central residue is the protonated His-46 of a β chain whose side chain interacts with the intrachain Asp-94, and its C-terminal forms an interchain salt bridge with the protonated Lys-40.

state. The inverse relation between Hb oxygen binding affinity and both the concentration of carbon dioxide and the acidity is known as the Bohr effect [21]. Interestingly, in spite of the hundred-year-old known effect, Hb oxygen affinity and the role of blood acidity are still actively studied and discussed in the context of the COVID-19 disease [22–24].

There is yet another vital buffer system in our body based on inorganic phosphate. The secondary pK_a value of the phosphoric acid buffer ($H_3PO_4^-$) is 7.2 at 310 K, allowing this buffer system to be very efficient at physiological pH [30].

$$H_3PO_4 \overset{2.2}{\rightleftharpoons} H^+ + H_2PO_4^- \overset{7.2}{\rightleftharpoons} 2H^+ + HPO_4^{2-} \overset{12.4}{\rightleftharpoons} 3H^+ + PO_4^{3-} \tag{1.12}$$

However, the concentration of HPO_4^{2-} in the blood is much lower ($\sim$1 mM [31]) than that of $HCO_3{}^-$. Nonetheless, the phosphate buffer plays an important part in regulating intracellular pH where inorganic phosphate is more concentrated [32–34]. Intracellular pH (pH$_i$) is slightly lower than the one of the blood, and it is conserved at $\sim$7.2 for most cells, even in different tissues and cellular locations (see Table 1.1).

Cellular metabolism is one of the main culprits for the acidification of the pH$_i$, with vital pro-

Table 1.1: Typical extracellular (pH$_e$) and intracellular (pH$_i$) pH values of cells in different tissues [25–29].

Cell Type/Tissue	pH$_e$	pH$_i$
Erythrocyte	7.4	7.2
Lymphocyte	7.4	7.1
Uterus	7.6	7.1
Muscle	7.3	7.1
Liver	7.2	7.2
Lung	7.3	7.1
Brain	7.1	7.2
Skeletal Muscle	7.3	7.1
Lung Tumour	6.5	7.2
Brain Tumour	6.9	7.3
Skeletal Muscle Tumour	6.7	7.2

cesses being net acid.

Glycolysis
$$\text{Glucose} + 2\ \text{NAD}^+ + 2\ \text{ADP}^{3-} + 2\ \text{HPO}_4^{2-}$$
$$\rightarrow 2\ \text{Pyruvate}^- + 2\ \text{NADH} + 2\ \text{H}^+ + 2\ \text{ATP}^{4-} + 2\ \text{H}_2\text{O}$$

Citric acid cycle
$$\text{Acetyl-CoA} + 3\ \text{NAD}^+ + \text{FAD} + \text{GDP}^{3-} + \text{HPO}_4^{2-} + 2\ \text{H}_2\text{O}$$
$$\rightarrow \text{CoA-SH} + 3\ \text{NADH} + \text{FADH}_2 + 3\ \text{H}^+ + \text{GTP}^{4-} + 2\ \text{CO}_2$$

Pentose phosphate pathway
$$\text{Glucose 6-phosphate} + 2\text{NADP}^+ + \text{H}_2\text{O}$$
$$\rightarrow \text{ribose 5-phosphate} + \text{CO}_2 + 2\text{NADPH} + 2\text{H}^+$$

ATP hydrolysis
$$\text{ATP}^{4-} + \text{H}_2\text{O} \rightleftharpoons \text{ADP}^{3-} + \text{HPO}_4^{2-} + \text{H}^+$$

Notice the importance of phosphate ions and phosphorylated compounds in these reactions. Adenosine triphosphate (ATP) is the main carrier of energy in cells, and its hydrolysis generates energy for many cellular processes, releasing H^+. In the opposite direction, ATP formation naturally helps alkalinizing pH_i. Our cells make so much use of ATP that in a day, the human body turns over its weight in ATP [35]. In normal conditions, the cell is at a steady state, and all these reactions and consequently the levels of metabolites, such as ATP, will also be balanced. Interestingly, ATP has a pK_a of 6.5, suitable to act as a cellular buffer. During physical activity, cellular respiration (including the net acid processes of glycolysis and the citric acid cycle) intensifies, ATP is used to power muscle contraction, CO_2 is formed, and even lipolysis can be triggered. All these biochemical processes contribute to the acidification of muscle cells during physical exercise, which despite contrarian reactions that consume H^+ such phosphocreatine hydrolysis amounts to a pH decrease of about one pH unit [36, 37].

Another factor greatly contributing to the acidification of cytosolic pH relative to the heavily buffered, and more alkaline medium it is immersed in, is the existence of an electrical potential across the membrane. A transmembrane electrical gradient or membrane potential is formed when there is a difference in the concentration of ions of opposite charge on two sides of a permeable membrane. In cells, the negative membrane potential can be mostly attributed to the balance between the asymmetric concentration of sodium ($[Na^+]_i$ = 15 mM; $[Na^+]_e$ = 142 mM), potassium ($[K^+]_i$ = 150 mM; $[K^+]_e$ = 4 mM), calcium ($[Ca^{2+}]_i = 10^{-4}$ mM; $[Ca^{2+}]_e$ = 1 mM) and chloride ($[Cl^-]_i$ = 5 mM; $[Cl^-]_e$ = 120 mM) inside and outside the cell [38, 39]. Due to the negative intracellular potential, cations will tend to flow inwards and anions outwards of the cell. Additionally, the high permeability of H^+ – four orders of magnitude higher compared to K+ [40, 41] – should originate a significant flux of H^+ to enter the cell. However, due to its low concentration, H^+ has low conductance across biological membranes. Taking the membrane potential into consideration and assuming an extracellular pH (pH_e) of 7.4, one would expect pH_i to be around 6.4 if protons were able to be at equilibrium across the cell membrane [42]. pH_i is considerably more alkaline than expected if only passive diffusion is considered and even more so when accounting for the aforementioned acidification pressure imposed by the metabolism of the cell. Furthermore, against the continuous passive acidifying influx of H^+, buffer systems are not sufficient to maintain pH homeostasis. The only way to regulate pH_i is through the activity of membrane-bound transporters.

There are a plethora of membrane proteins specialized in the transport of protons with great significance to cellular pH regulation. Throughout evolution, several strategies both to intake and outtake H^+ have appeared: some are powered by ATP hydrolysis (direct active transport) while others harness the energy of electrochemical ion gradients (secondary active transport); some do not involve net charge movement (electroneutral) while others do perform a net charge translocation across the membrane (electrogenic); some proteins co-transport different chemical species in the same direction (symport) and others exchange chemical species between the extracellular and intracellular media (antiport).

One of the simplest strategies for protecting cells against excessive acidification is to exchange intracellular H^+ for extracellular monovalent cations such as Na^+. The Na^+H^+ exchanger (NHE) is a secondary active transporter that exports H^+ in an electroneutral manner. As it works, it dissipates the sodium gradient generated by Na^+K^+ ATPase, and it is responsible for most sodium influx into cells [43]. NHE is ubiquitously distributed in most tissues, and one of the most important transporters responsible for pH regulators [44]. Its activity can be allosterically modulated by pH_i due to an additional intracellular H^+ binding site. Because of this H^+ sensor, this transporter is mostly inactive at cytosolic pH values ≥ 7.2 to prevent further alkalinization, and contrarily the decrease of pH_i sharply increases its activity [45]. Furthermore, the internal pH sensitivity can also be altered by a variety of signals (hormones, neurotransmitters, growth factors, and even physical stimuli) that are believed to modulate its phosphorylation state [42].

Monocarboxylate transporters (MCTs) co-transport H^+ and monocarboxylic acids, such as lactate, pyruvate, and ketone bodies. MTCs are widely expressed, although their tissue abundance depends on their metabolic requirements and specific function. It is important to note that in most tissues, MCTs do not function as a pH regulatory mechanism. However, they are essential for preventing the acidification of certain types of muscle (namely red skeletal muscle fibers, white skeletal muscle fibers, and cardiomyocytes), accounting for up to 40% of the H^+ efflux in an oxygen-restricted environments [46, 47].

Up until now, we have seen examples of transporters that effectively extrude H^+ in order to prevent pH_i acidification. An alternative strategy to achieve the same goal involves the intake of a weak extracellular base such as HCO_3^-. As we have seen, an influx of carbonate would decrease the [H^+] by forming H_2CO_3. The sodium bicarbonate cotransporter (NBC) are membrane proteins that translocate Na^+ and HCO_3^- in the same direction. Interestingly, depending on the isoform NBCs can be either electroneutral and electrogenic due to different coupling stoichiometries [48]. NBCn1 and NBCn2 mediate a 1 Na^+/1 HCO_3^- cotransport which is solely controlled by the combined chemical gradients of the ions. Contrarily, NBCe1 and NBCe2 are electrogenic and thus also dependent on the electrical potential. Interestingly, these isoforms can change their flux direction. With a 1 Na^+/2 HCO_3^- stoichiometry the transporter mediates a net influx whereas adopting a 1 Na^+/3 HCO_3^- stoichiometry – provoked by an increase in intracellular [Ca^{2+}] which possibly controls its phosphorylation state [49] – dictates a net HCO_3^- efflux across the membrane.

For a fine regulation of pH_i, both acid extruders and acid-loading transporters are necessary. In order to prevent over-alkalinization, there are membrane transporters such as anion exchangers (AE) that leverage the Cl^- gradient to expel HCO_3^- from the cell. Cells that secrete acid, such as epithelial cells in the stomach, require the action of AEs to prevent threateningly alkaline levels [50]. AEs are able to prevent an excess alkaline efflux with their pH-sensitive N-terminal cytoplasmic domain, which can decrease its activity by around 11-fold as pH_i drops from 7.8 to 7.05 [51]. The pairing of pH-regulated alkalinizing NHEs and acidifying AEs allow for rigorous control of pH_i near 7.2 [52, 53].

Arguably, the most well-known proton transporters are the proton pumps, particularly those driven by ATP hydrolysis – the H^+ ATPases – to move protons against their electrochemical gradient. There are several types of these H^+ ATPases, one of which is the P-type ATPase. The naming of P-type derives from the fact that these plasma membrane transporters are regulated by phosphorylation. In the gastric glands of mammals, H^+K^+ ATPases secrete H^+ to acidify the stomach and ensure an optimal environment for digestive enzymes such as pepsin [54, 55]. There are other P-type ATPases that, instead of H^+, transport other cations like the previously mentioned Na^+K^+ ATPase responsible for creating asymmetric cytosolic and extracellular ionic concentrations.

V-type ATPases are another class of proton transporters present in several eukaryotic cells and organelles. These proton pumps are responsible for acidifying the vacuole (hence the V) in fungi and higher plants and several sub-cellular organelles (e.g., endosomes, lysosomes, Golgi complex, secretory vesicles) in animal cells. F-type ATPases, an additional category of H^+ ATPases, are structurally related to V-type ATPases. Interestingly, F-type ATPases can both catalyze an ATP consuming membrane translocation of H^+ against the gradient and also the reverse reaction, ATP synthesis. The latter function of these membrane proteins (also named ATP synthases when performing this function) found in the inner membrane of mitochondria is of pivotal importance in cellular respiration, converting the proton gradient created by the electron transport chain into ATP production. As we have seen, all these ATPases, albeit extruding protons, are not particularly relevant in the context of cytosolic pH homeostasis. Yet, they play a central role in organellar pH regulation.

The internal pH of certain organelles is independently controlled and differs from the cytosolic one (Table 1.2). Maintaining these distinct pH values is vital for the cell to function correctly and, ultimately, to its survival. For example, without the proton gradient across its inner membrane, the mitochondria would not be able to use the ATP synthases, thus costing the cell its primary source of energy. Likewise, lysosomes are required to have a pH of around 5.0 so that the hydrolytic enzymes can digest endocytosed nutrients, recycle organelles trapped by autophagosomes or even defend against pathogens such as viruses and bacteria [58]. Luminal acidification is critical, not only for lysosomes but for all endosomes. For example, in receptor-mediated endocytosis, acidification of the endosomes is important to facilitate the dissociation of the internalized ligand-receptor complexes. Furthermore, normal membrane sorting and trafficking depend on the gradual acidification throughout the endocytic and secretory pathway [59].

For the cell to perform its normal functions, its pH homeostasis must be maintained. Naturally, metabolic or genetic perturbations that disturb cytosolic or organellar pH affect the organism's health, lead to the development of diseases, and in some cases, even to its death. For example, mutations of Cl^{-} H^{+} exchangers disrupt the acidification of early endosomes, late endosomes, and lysosomes, causing renal failure, osteopetrosis, and lysosomal storage diseases, respectively [60–64]. In addition, mutations in v-ATPase subunits are associated with bone disorders, deafness, and intellectual disability [65, 66]. Deregulation of lysosomal pH is implicated in cell aging and longevity as well as in neurodegenerative disorders, i.e., Parkinson's disease and

Table 1.2: Typical pH values of several subcellular compartments [56, 57].

Subcellular compartment	pH
Cytosol	7.2
Early endosome	6.5
Late endosome	6.0
Lysosome	5.0
Endoplasmatic Reticulum	7.0
cis-Golgi	6.7
trans-Golgi	6.3
Secretory Vesicles	5.5
Mitocondria (Intermembrane)	6.9
Mitochondria (Matrix)	7.8

Alzheimer's disease [66–68].

Alzheimer's (AD) is the most prevalent neurodegenerative disease, with around 50 million AD patients worldwide. Although there are available treatments to alleviate its symptoms, at the moment, there is no cure for this disease, which is expected to affect 152 million by 2050 [69]. Lysosomal and intracellular pH misregulation is deeply related to this disease, which has many distinct forms. A decrease in pH$_i$ impairs neural activity [70], and AD neurons have been reported to be acidic [71]. In fact, most of the hallmarks of AD are related to the acidic pH$_i$: apoptosis; Tau protein hyperphosphorylation; abnormal extracellular deposits of beta-amyloid protein (Aβ) fibrils (senile plaques). It has been shown that a pH$_i$ of 7.0 is enough to activate pH-dependent endonucleases, which increase DNA fragmentation and trigger apoptosis [72]. Brain acidosis forces Tau hyperphosphorylation by activating an endopeptidase which in turn catalyzes inhibitors for the phosphatase responsible for cleaving Tau's phosphate groups [73]. A lower pH$_i$ also increases the activity of β-secretase, an enzyme that leads to the generation of Aβ by cleavage of the amyloid precursor protein (APP). Under physiological conditions, APP is mostly cleaved by α-secretase. However, in AD brains, the β-secretase activity is augmented, which drives an overproduction of Aβ [71]. The causes of the pH$_i$ deregulation are not consensual, although thought to be mitochondrial damage and autolysosomal membrane damage and leakage [74].

One can consider the opposite of accelerated apoptotic cell death as accelerated cell replication, and both extremes are related to pH$_i$. As we have discussed, there is a link between a lower pH$_i$, increased apoptosis, and AD. There is also a relation between a higher pH$_i$, increased cell replication, and cancer. The connection between pH$_i$, AD, and cancer is surprisingly strong. So much so that an inverse correlation between the incidence of AD and cancer has been reported [75–77]. Both diseases can be considered metabolic diseases and share multiple common epidemiological patterns. Although inheritable genetic risk factors exist for both diseases, most cases are sporadic and markedly age-related. Furthermore, in both AD and cancer, there is a severe deterioration in the energy yield of the cells as a result of mitochondrial dysfunction [78–82].

The shift from oxidative phosphorylation to aerobic glycolysis, named Warburg effect, is a hallmark of cancer [83]. The increased glycolytic activity produces an excess of acid within the cell, which could lead to apoptosis. However, contrary to initial expectations and popular belief, the pH$_i$ of cancer cells is typically alkaline while pH$_e$ is acidic [84]. This reverse pH gradient, not commonly observed in other tissues other than solid tumors, is vital for cancer progression and is achieved by the overexpression of V-ATPases, which has been observed in different types of tumors [85, 86]. These proton transporters play an essential role in regulating the pH$_i$ of cancer cells that are more sensitive to V-ATPase inhibition than normal cells. It has been reported that archazolid, a V-ATPase inhibitor, induces almost three times more apoptosis in tumor cells, suggesting that the transporter is a promising potential target in the development of anticancer drugs [87, 88]. However, preventing apoptosis is not the only capacity in which V-ATPases support cancer cells. These proton transporters are involved in cancer cell migration and invasion. Even though the mechanism involved is not fully understood, the loss of V-ATPase activity has been shown to reduce both *in vitro* invasion, and migration and *in vivo* metastasis of a various cancer cell lines [85, 89–91]. It has been hypothesized that extracellular microenvironment acidification may promote the activity of pH-dependent proteases that participate in the digestion of the extracellular matrix.

The ability to sense and react to external and internal pH changes is a survival requirement for pathological and healthy cells alike. Maintaining homeostasis is of the utmost importance as pH_i regulates a myriad of cell processes. Acidic pH_i values favor cell differentiation and proliferation, and, contrarily, cell death is accelerated by alkalosis. Cytosolic pH homeostasis is tightly regulated, and it has long been established that even minor disruptions can have a severe impact on cell metabolism, function, and survival. However, our molecular understanding of pH-dependent effects in proteins and other biomolecules is still very limited, particularly in big macromolecular complexes such as protein-protein or membrane protein systems.

1.2 pK_a values in proteins

Out of the 20 canonical amino acids that are genetically encoded, only 7 possess the ability to exchange protons with water in the usual 0-14 pH range (Table 1.3). Their pK_a values in water have long been known, with existing reports dating back to the early 20th century [96–98]. Although the absolute values and the accuracy of the measurements have fluctuated with the evolution of the methods employed, these quantities are considered elementary biochemistry knowledge and are taught in introductory classes worldwide. By definition, these pK_a values dictate the relative abundance between the proton-bound and unbound states. Consequently, the pK_a values also determine the charge state preference of the titratable groups. For example, glutamate, an anionic amino acid, will be mostly deprotonated and thus negatively charged at pH 7. Conversely, lysine, a cationic amino acid, will be mainly protonated and positively charged at the same pH value.

Table 1.3: pK_a values of ionizable groups in water [92, 93] and protein environments [19, 93, 94]. The reported experimental pK_a minimum, mean, maximum and amplitude of the different amino acids has been calculated from the PKAD collection [95].

Residue	Chemical group	Type	water pK_a	protein pK_a	Min	Mean	Max	Range
C-terminal	α-Carboxyl	Anionic	2.33	3.67	2.4	3.2	4.0	1.6
Aspartate	γ-Carboxyl	Anionic	3.71	3.94	0.5	3.5	9.9	9.4
Glutamate	β-Carboxyl	Anionic	4.15	4.25	2.1	4.1	7.2	5.1
Histidine	Imidazole	Cationic	6.04	6.54	2.5	6.6	9.2	6.7
N-terminal	Primary amine	Cationic	9.71	8.00	6.9	7.6	9.1	2.2
Cysteine	Sulfhydryl	Anionic	8.14	8.55	2.9	6.1	11.1	8.2
Tyrosine	Alcohol	Anionic	10.10	9.84	9.7	10.6	12.1	2.4
Lysine	Primary amine	Cationic	10.67	10.40	6.5	10.7	12.1	5.6
Arginine	Guanidine	Cationic	13.90*	13.80*	—	—	—	—

* Values for Arg have been taken from Fitch *et al.* [93]. To estimate the contribution of the protein backbone, a capped glycine-based tripeptide with an Arg residue in the middle was utilized instead of the alanine-based pentapeptides used for the remaining amino acids.

In proteins, the protonation preference of a residue has a far-reaching impact on the electrostatics of the environment and on its availability to participate in hydrogen bonds, salt bridges, and other electrostatic interactions. Likewise, the environment itself greatly influences the proton or the charge avidity of a titratable residue. To include the effect of the protein backbone, Nick Pace's group has measured the pK_a values of alanine-based pentapeptides with capped termini (Acetyl-A-A-X-A-A-Amide, where X $\in$ Asp, Glu, His, Cys, Tyr, Lys) [19, 94]. These pentapeptides are a good model for an unfolded and uncharged protein, as they minimize secondary structure formation and hydrogen bonding interactions that occur in proteins and are comprised only of neutral groups. Therefore, these values are often used as a reference to determine the pK_a shifts induced by the remaining environment. Desolvation is one of the most significant effects contributing to the perturbation of a residue's proton affinity. By inserting the Ala-based pentapeptides into a lipid membrane, it is possible to observe a gradual medium exchange accompanied by increased desolvation and consequent pK_a shifts that can go as high as 3.5 pH units [99]. As the residue stabilizes its neutral state in a water-deprived environment, the remaining molecule is required to accommodate the new charge.

The inseverable link between protonation and conformation has been ingeniously taken advantage of by nature. The 7 titratable amino acids make up for $\approx$31% of all amino acids in the human proteome (Arg: 5.8% ± 2.3; Asp: 4.7% ± 1.7; Cys: 2.4% ± 2.2; Glu: 7.0% ± 2.9; His: 2.6% ± 1.3; Lys: 5.8% ± 2.8; Tyr: 2.8% ± 1.4), meaning that on average roughly one third of a protein is sensitive not only to the medium pH but also to alterations in its surroundings. Unsurprisingly, protonation switches have been associated with a wide array of protein behaviors, including modulating the binding affinity in protein-protein, protein-ligand, or protein-lipid systems, modifying enzymatic activity and function, and even altering their stability, and subcellular location [100]. The (de)protonation events are so important to regulate protein structure and function that it has been proposed to consider protonation as a post-translational modification, albeit being an enzyme-free reaction, contrarily to most post-translational modifications [100]. The relevance of chemical groups with labile protons in enzymatic function is well illustrated by the fact that the top seven most frequent amino acids in catalytic sites are all pH titratable (Table 1.4). In total, they make up approximately 80% of all catalytic residues. Furthermore, only these residues are more represented in catalytic sites than in the rest of the protein.

Histidines and cysteines are exceptionally interesting as the most under-represented titratable residues in the human proteome, yet the most enriched catalytic residues. Also, the two residues have the closest pK_a values to the physiological pH range, making these residues the most sensitive to subtle pH changes in the acidic or alkaline direction. For these reasons, His and Cys residues are generally the most important to predict accurately. Histidine is the most frequent catalytic residue and has the largest enrichment in this region of enzymes. Moreover, His residues are present in key regions of proteins with very different functions [100]: actin regulatory protein cofilin [103]; phosphoinositides binding Grp1 [104], the islet amyloid polypeptide, a hormone produced by pancreatic cells [105]; "mad cow" disease prion protein [106]; influenza A viruses proton channel M2 [107]; HIV-1 Gag polypeptide [108]. Also, it has been reported that mutations in different cancer cell lines cause a net gain in His, as well as in Cys residues [109]. Cysteine is arguably one of the most unique and reactive amino acids. It possesses the ability to form a disulfide bond (oxidize) by reacting with another Cys, and it can replace another amino acid [110], the Selenocysteine, with a much lower pK_a of 5.24 [111]. Under certain

Table 1.4: Frequency of the canonical amino acids as catalytic residues. The respective enrichment is calculated as the ratio between the frequency in the catalytic site and the frequency in the protein sequence. The data set used consists of 684 enzymes from the Mechanism, and Catalytic Site Atlas [101]. Adapted from reference [102].

Amino Acid	Frequency	Enrichment
His	19.5%	8.2
Asp	17.4%	3.0
Glu	13.0%	2.0
Lys	9.3%	1.7
Arg	9.2%	1.8
Cys	6.3%	4.6
Tyr	6.2%	1.8
Ser	5.4%	0.9
Asn	3.9%	0.9
Thr	3.1%	0.6
Gln	1.8%	0.5
Phe	1.6%	0.4
Trp	1.5%	1.0
Met	0.6%	0.2
Leu	0.4%	< 0.1
Ile	0.2%	< 0.1
Val	0.2%	< 0.1
Gly	0.2%	< 0.1
Ala	0.1%	< 0.1
Pro	0.1%	< 0.1

conditions like Se deficiency, Sec – known as the 21^{st} amino acid to be discovered as genetically encoded (by a stop codon) – Cys can occur in the Sec position and partially preserve its function [112, 113]. Cys is the least exposed protein residue, and unexpectedly, more solvent-exposed Cys are more pH-sensitive than buried ones [114].

The placement of titratable amino acids in the protein has also been subject to evolutionary pressure. An evidence of that is the bimodal and trimodal isoelectric point (pI) distribution of bacterial, archaeal and eukaryotic proteomes [115–119] and its adaptation to environmental conditions [120–122]. The pI of a protein is the pH at which its net charge is zero, and it is commonly used to distinguish between proteins in methods for protein separation, such as electrophoresis. In water and at the pH that corresponds to their pI, proteins are less soluble, which may result in protein aggregation [123]. Appropriately, the multimodal proteome-wide distributions show a low abundance of proteins with a pI close to the physiological pH. The pI distribution of the human blood proteome is bimodal and exhibits a local minimum at pH 7.4, where the number of proteins is approximately 20 times lower than that at pH 5.3 [124]. It was shown that AD patients with fast memory decline had a significantly higher concentration of proteins with a pI of 7.4±0.1 compared to patients with a slow memory decline. Interestingly, the major and minor peaks of the whole-proteome pI distribution are associated with subcellular compartments [118, 125]. Furthermore, organelle-specific protein pIs are largely defined by local pH and membrane charge: basic pI values are correlated with nuclear and mitochondrial localization; plasma membrane and extracellular proteins exhibit no preference for either the acidic or alkaline regions; pI in the acidic regions are more characteristic of cytoplasmic, cytoskeletal, and lysosomal proteins.

The overall charge of a protein is governed by the individual charge states of its titratable amino acids, which can be regulated by the respective local environments. The diversity of existing environments can be noticed by extraordinary dispersion of reported experimental pK_a values in wild-type proteins (Table 1.3). The range of pK_a values observed for Asp is particularly impressive (from 0.5 to 9.9). Nonetheless, at pH 7, many other residues (Asp, Cys, His, Lys, Glu) can also exist primarily in neutral or ionized form. It should be noted that even though Tyr residues with pK_a values inferior to 9.7 have not been reported, environments in which much lower pK_a values can exist are, at least theoretically, possible. Moreover, only 23 experimental values for Tyr residues have been reported, a very limited amount compared to other residues (Glu: 411, Asp: 381, His: 224, Lys: 135). The reported amplitudes can nevertheless be interpreted as representative of the typical environments experienced by the various residue types. Termini residues are generally exposed to water and thus are less shifted. The same logic can explain residues with lengthier side chains, such as Glu and Lys, which tend to be more water-exposed. Contrarily, the shorter Asp, Cys, and His show the broadest range of pK_a values. There are no reported experimental pK_a values in the PKAD database for Arginine residues, nor has a neutral Arg ever been observed near-neutral pH [93]. In part, this is due to its extremely high pK_a of $\approx$14, roughly two log units higher than commonly cited values for the arginine amino acid that stem from century-old work [126, 127]. Also, the length of its side chain helps Arg to discover polar groups or water even in internal positions where other residues are forced to shift to their neutral states [128].

Experimental pK_a can be obtained employing a variety of methods [129]. The first one to be developed was potentiometric titration, with the first setup to determine equilibrium constants being described in the early 1900s [130]. This method involves a step-wise addition of a known volume of acid or base to a solution of our molecule of interest, typically a drug or a protein. By measuring the change in pH solution with a pH electrode, and plotting it as a function of the volume of the added titrant, the pK_a value can be identified as the inflection point in the curve. Potentiometry is still nowadays a common method for determining pK_a values of small molecules, however, its usage is not as widespread in proteins [93]. Multiple residues titrating in the same pH region are common in proteins. Without complementary information, it is impossible to extract and assign pK_a values to specific ionizable groups from a titration curve. If a chromophore is found near the titration site, one can distinguish his pK_a by measuring the UV absorbance at different pH values [131]. Unfortunately, nearby chromophores are usually affected by other sites as well, and thus UV spectrophotometry derived titration curves are also more common for small molecules than for proteins. Isothermal titration microcalorimetry (ITC) is another method with which it is possible to get titration curves selectively. ITC is a popular choice for measuring equilibrium constants and related thermodynamic parameters of binding reactions. By monitoring the pH dependency of the time derivative of heat change of the reaction between a reagent and an ionizable group, it is possible to obtain titration curves for a specific group [132]. For example, by following the reaction of iodoacetamide with the ionized cysteine's thiol, one can assess the pK_a of those residues. Still, if multiple residues have similar pK_a values, the same disentangling difficulties apply (see below).

NMR pH titration is the method of choice for site-specific pK_a values in proteins. In fact, 96.2% of the experimental pK_a values collected in PKAD have been determined with this methodology. NMR spectroscopy is highly sensitive to protonation events as the chemical shift of a nucleus close to a titratable site depends on its protonation state. Much like in calculations for structure determination, a single protein sample uniformly enriched with isotopes is used to measure chemical shifts of individual amino acid types at once. Alternatively, specific isotopes can be introduced to a small subset of residues. The titratable protons can be directly monitored if significantly protected from the fast exchange with water by being buried or performing hydrogen bonds [133, 134]. In most cases, the pH-dependent chemical shifts need to be inferred from nearby nonlabile nuclei. ^{1}3C and ^{1}5N are usually well suited for these studies since the ionization of a nearby site triggers a more significant response than changes in the electric field. Compared to the heavier isotopes, ^{1}H sensor protons are typically further away and experience irregular chemical shifts [135].

The observed chemical shifts δ_{obs} relate to changes in the weighted average of the deprotonated and protonated populations, p_0 and p_1, respectively. It can be defined by:

$$\delta_{obs} = p_0\delta_0 + p_1\delta_1 \tag{1.13}$$

where δ_0 and δ_1 are the chemical shift of the deprotonated and protonated species. After the careful assignment of the NMR peaks at different pH values to the correct nuclei, it is possible to track the titration of a site and fit it to a Henderson-Hasselbalch equivalent equation [135]:

$$\delta_{obs} = \frac{\Delta\delta}{1 + 10^{\,\mathrm{pH}-\mathrm{p}K_\mathrm{a}}} + \delta_0 \tag{1.14}$$

Here $\Delta\delta$ refers to the chemical shift change upon proton binding. However, this equation can only be applied to a single non-interacting titrating site. NMR chemical shifts are extremely

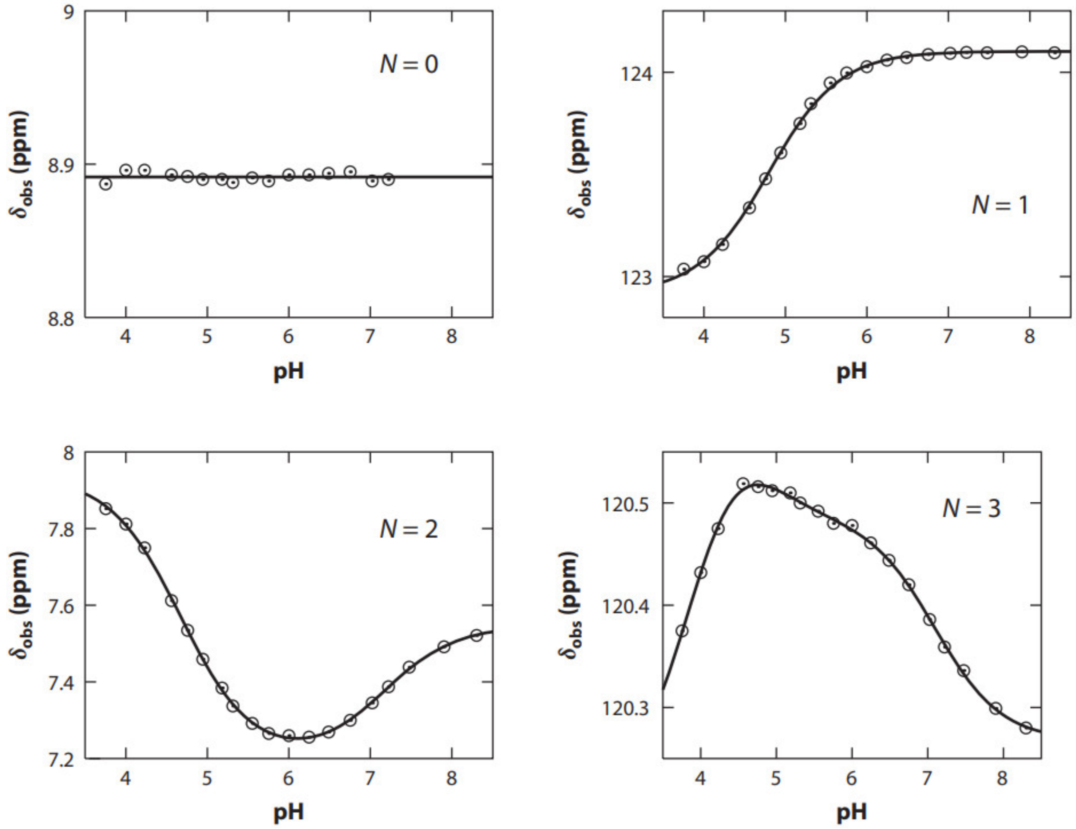


Figure 1.3: Examples of fits with Equation 1.15 to experimental chemical shift titration curves influenced by N noninteracting titration sites. Adapted from Hass *et al.* [135, 136].

sensitive to changes in the environment. As such, several nuclei can report the same titration, and one nucleus may sense more than one site. Furthermore, many titratable residues are often close to each other in proteins, resulting in complex titration behaviors and complicating the tracking and assignment of the shifts. If the monitored nucleus is affected by more than one non-interacting protonations sites N, their contributions to the chemical shift are additive and can be generally described as:

$$\delta_{obs} = \sum_{i=1}^{N} \frac{\Delta\delta_i}{1 + 10^{\,\mathrm{pH}-\mathrm{p}K_{\mathrm{a}i}}} + \delta_0 \tag{1.15}$$

The number of titratable residues can be determined, and this equation can fit very convoluted titration curves like nonmonotonous ones (Figure 1.3). In certain conditions, it is possible to determine all the N pK_a values from a single titration curve, however, in some cases it is necessary to know one of the pK_a values to resolve the others. Furthermore, this equation can not describe titration curves of interacting sites in which there are cooperativity effects. Cooperative protonation can be approximated with the Hill model:

$$\delta_{obs} = \frac{\Delta\delta}{1 + 10^{n(\mathrm{pH}-\mathrm{p}K_\mathrm{a})}} + \delta_0 \tag{1.16}$$

If the Hill coefficient n is 1, no cooperativity is observed, and the equation reverts to the Eq 1.14. Despite its inability to reproduce multiphasic titration curves, it is still quite a helpful approximation as it limits the amount of unknown independent parameters to be determined.

Determination of pK_a values using NMR pH-titration is not as straightforward as one might think. A fitting equation is required, and its selection is subjective and accompanied by the inherent assumptions and limitations of the model chosen. Sometimes it is possible to use statistical treatments to choose the fitting model [137]. Nevertheless, it is necessary to visually inspect titration curves and resulting fits, as well as use structural information and knowledge about titration behavior to check the appropriateness of the fitting model. The error of a pK_a determination with NMR, therefore, has two components, one from the NMR experiment and another from the fitting procedure, and both are hard to access. Consequently, true accuracy is hard to determine from a single work, and usually, the experimental errors reported only reflect the precision of the measurements. In an effort to obtain a realistic accuracy of experimentally determined pK_a values, Webb *et al.* have remeasured pK_a values of HEWL – a common benchmark for protein pK_a calculations – with NMR spectroscopy and compared the results with previous works [138]. In this comparison, differences as high as one pH unit can be observed between the new and old measurements. pK_a values have also been reported for distinct nuclei (^{1}3C, ^{1}5N, and ^{1}H), and the RMSD values between the more accurate ^{1}3C experiments and the others range from 0.4 to 1.3. Interestingly, the two theoretical methods tested have exhibited a lower RMSD (0.8 and 0.9) than some of the nuclei used. In conclusion, it is important to acknowledge that not all experimental values have the same validity and that one can not interpret experimental pK_a values as absolute truths without any associated error. Nonetheless, NMR titration is still one of the most, if not the most accurate, experimental methods available.

Unfortunately, NMR is a quite expensive equipment to acquire, maintain and operate [139, 140]. This explains in part why the number of protein pK_a values has not expanded significantly in recent years. In the same way, the rate of NMR structure releases in the PDB has dropped substantially since 2007 and is now only a third compared to their maximum. Last year, new NMR structures were one-tenth of the electron microscopy (EM) ones and $\approx$ 25 times less than X-ray releases.

Characterizing a titration curve of a residue by its pK_a value leads to an enormous loss of information regarding the dependency of other sites on its behavior. Furthermore, the interaction between sites can produce irregular non-sigmoidal titration curves, which can not be described by a set of pK_a values [135, 141–144]. Even if only two sites are interacting, this irregular curve can be observed (Figure1.5C). To fully characterize the titration of coupled sites, it is necessary to consider the microscopic model and its relation with the macroscopic observables. In a system with two interacting proton binding sites, there are three macroscopic states Figure 1.4A) and four microscopic states can be adopted (Figure 1.4B). The microscopic equilibrium con-

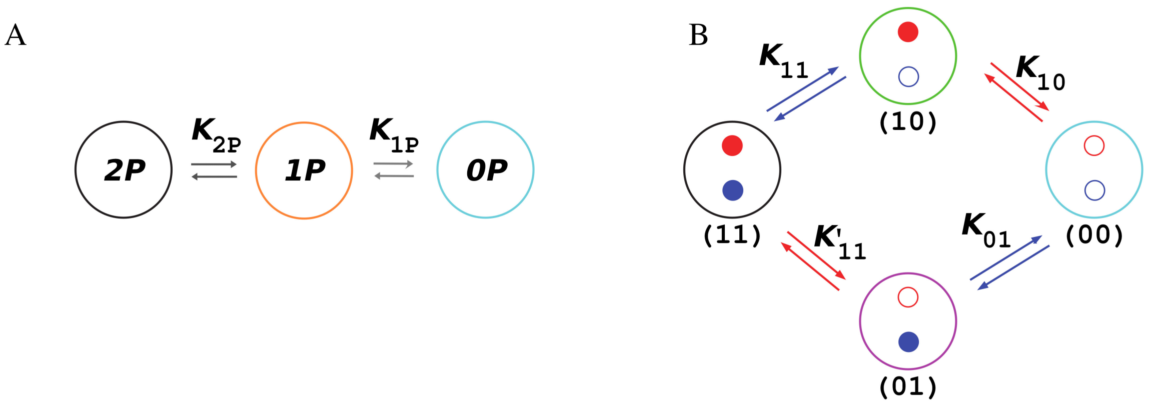


Figure 1.4: Macroscopic (A) and microscopic (B) interpretation of a system with two coupled sites s_1 in red and s_2 in blue. Adapted from Ullmann *et al.* [142]

stants can be defined as

$$\begin{aligned} K_{11} &= \frac{[(10)][H]}{[(11)]} \\ K'_{11} &= \frac{[(01)][H]}{[(11)]} \\ K_{10} &= \frac{[(00)][H]}{[(10)]} \\ K_{01} &= \frac{[(00)][H]}{[(01)]} \end{aligned} \tag{1.17}$$

These microscopic dissociation constants are related to the macroscopic ones by

$$\begin{aligned} K_{2\mathrm{P}} &= K_{11} + K'_{11} \\ K_{1\mathrm{P}} &= \frac{K_{10}K_{01}}{K_{10}+K_{01}} \end{aligned} \tag{1.18}$$

or in their pK equivalent form as

$$\begin{aligned} \mathrm{p}K_{2\mathrm{P}} &= -log_{10}(10^{-\mathrm{p}K_{11}} + 10^{-\mathrm{p}K'_{11}}) \\ \mathrm{p}K_{1\mathrm{P}} = \mathrm{p}K_{10} + \mathrm{p}K_{01} &+ log_{10}(10^{-\mathrm{p}K_{10}} + 10^{-\mathrm{p}K'_{01}}) \end{aligned} \tag{1.19}$$

A titration curve of a site is no more than its probability of existing in a protonated species. For site s_1 in our system consisting on two interacting sites, its total average protonation (the observable quantity in a titration) $<x_1>$ is equal to the sum of <(11)> and <(10)> and in the case of s2 is $<x_2> = <(11)> + <(01)>$. Hence, from the two titration curves alone, it is impossible to infer the microstates' probabilities. To add to the complexity of experimental determination of pK_a values, as we have seen, an observed experimental titration curve may be reporting multiple sites, which makes the information unraveling even more complicated.

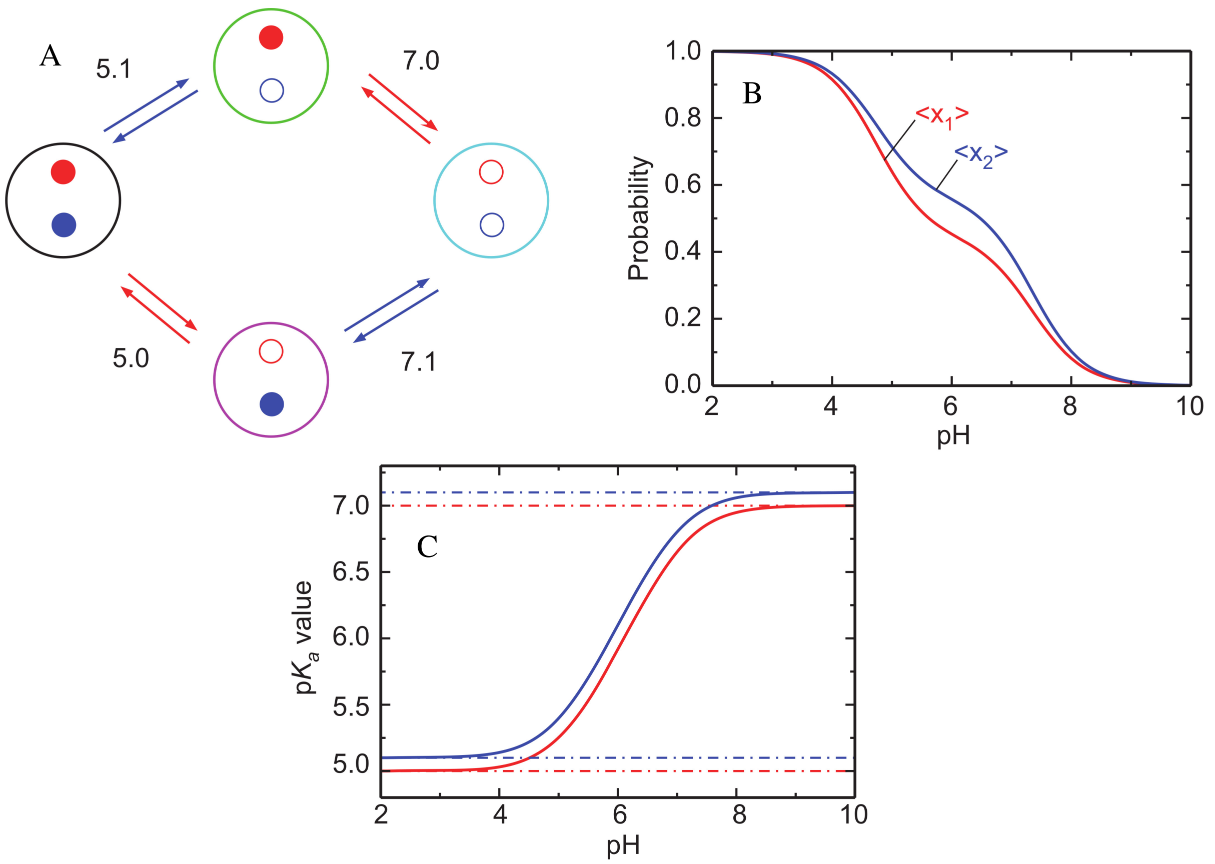


Figure 1.5: (A) Microscopic pK_a values of a system with two interacting titratable sites (s_1 in red and s_2 in blue) with a interaction strength of 2 pH units that titrate in the same pH range. (B) Titration curves of the two sites. (C) pH-dependent pK_a values of the two sites. Adapted from Ullmann *et al.* [142]

The strength of the interaction between sites is reflected in the difference between the microscopic proton dissociation constants of the same site (pK_{11} and pK_{01}), and the stronger the interaction, the larger the difference in these constants. In the limit case of no interaction between the sites, the two microscopic pK_a values would be equal. Figure 1.5A illustrates a system where the interaction energy between the two sites is equivalent to 2 pH units in which both sites titrate in the same pH region. In this strongly interacting system, macroscopic pK_a values are not enough to describe the non-sigmoidal titration curves (Figure 1.5B), as in reality these sites possess pH-dependent pK_a values (Figure 1.5C). Moreover, the macroscopic pK_a values (p$K_{2\mathrm{P}}$ = 4.74; p$K_{1\mathrm{P}}$ = 7.35) are not coincident with the so-called pK_{half} or p$K_{1/2}$, pH value of the titration curve point at which the protonation probability is 0.5. The pK_{half} is often used to describe the titration behavior of a site, and if the titration curve can be fitted by an HH equation, it is not a bad descriptor. However, for irregular titration curves, pK_{half} values do not correspond to any quantity with physical meaning, being good first guesses nonetheless.

If the two sites do not titrate in the same region, the titration curves recover their sigmoidal shape (Figure 1.6B), even if they are interacting strongly (Figure 1.6A). In this case, the pK_half values match the macroscopic pK_a values (p$K_{2\mathrm{P}}$ = 2.0; p$K_{1\mathrm{P}}$ = 7.0), however, thanks to the interaction

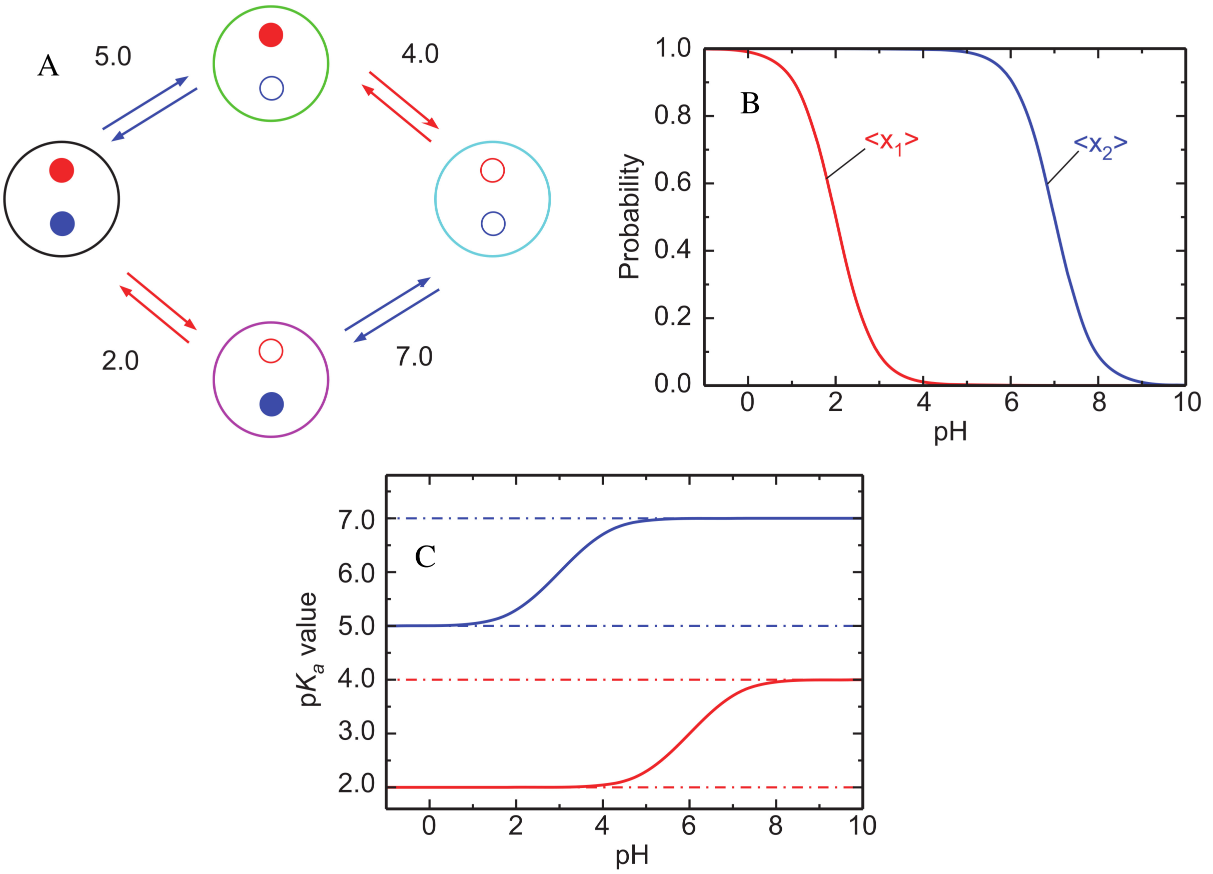


Figure 1.6: (A) Microscopic pK_a values of a system with two interacting titratable sites (s_1 in red and s_2 in blue) with a interaction strength of 2 pH units that titrate in distinct pH ranges. (B) Titration curves of the two sites. (C) pH-dependent pK_a values of the two sites. Adapted from Ullmann *et al.* [142]

between the sites, the pK_a values of each site are also pH-dependent (Figure 1.6C). At pH 8, both sites will be predominately deprotonated, thus, the microscopic pK_a values related to this microscopic state will be in effect ((pK_{10} = 4.0; pK_{01} = 7.0)). For site s_1 this means that at this pH value its pK_a value differs significantly from the macroscopic one, however, its protonated population is very low. Nevertheless, in non-equilibrium situations that may occur during a catalytic reaction these microstates could play an important functional role.

As we have seen, complex titration behavior can be interpreted by the microscopic model. With this framework, a protein with N titratable groups is defined by $\mathrm{N2^{N-1}}$ microscopic equilibrium constants. Even though only the 2^N - 1 independent constants need to be determined, for systems with more than three sites, it is impossible to derive them all from their titration curves since only $\mathrm{N^2}$ - N + 1 parameters can be resolved [141].

Computational methods for pK_a estimation can be used to complement experimental methods and aid in the interpretation of intricate experimental titration curves by providing suitable micro-constants. They can also be employed as a replacement for the expensive experimental measurements of pK_a values. Additionally, computational predictors may refine the macroscopic pK_a values and or microscopic constants for the desired conditions, as often the experimental conditions are not appropriate for the study of a particular system or process. In the PKAD database, some pK_a values have been experimentally determined with an ionic strength as high as 5M, and the acid dissociation constant is, in fact, dependent on, among others, temperature, ionic strength, and the solvent.

Nowadays, if one wants to predict pK_a values in proteins, there is a vast array of options available with quite distinct characteristics. There are several possible ways to group the existing methods. It is possible to distinguish them based on the type of model used to describe the system, the degree of conformational flexibility allowed, or whether the microscopic model is used. Table 1.5 contains conventional classes of pK_a estimators: empirical, Quantum Mechanics (QM)-based, Continuum-Electrostatics (CE)-based, Molecular Dynamics (MD)-based.

Ab initio QM methods solve the Schrödinger equation for a subatomic characterization of a system. Due to the high computational cost, it is impossible to solve it for a complete protein. Thus, in most methods, only part of the system is described by QM, and several types of models can describe the remaining. In the QM/LPBE method, the titratable group is modeled with QM, and the bulk solvent with the linear Poisson–Boltzmann equation [145]. A variant of this method is the QM/MM/LPBE, in which part of the protein is described by Molecular Mechanics (MM) [146]. SCC-DFTB/MM-GSBP is another QM/MM method developed which has been used to calculate pK_a values using a free energy perturbation (FEP) approach [147]. QM-based methods are instrumental in dealing with titratable residues near metal atoms [148], which are poorly described by electrostatics alone.

Empirical models are the most modern and the fastest as they rely on simple system descrip-

Table 1.5: Examples of pK_a predictor methods organized into classes based on the used model to describe the system and degree of conformational flexibility. The methodologies developed and presented in this thesis have been underlined.

Class	Examples	Model	Conformational Flexibility	Speed
QM-based	QM/LPBE, QM/MM-GSBP	QM/MM	−−	−
Empirical	PROPKA, PKA17	Empiral Function	−−	++++
	pKAI, pKa-ANI	Machine Learning	−−	++++
CE-based	GBneck, GBSW, GBMV	GB	+	+++
	PypKa, DelPhiPka, H++	PB	−	++
	MCCE	PB	+	+
MD-based	MM-GBSA	GB	−	−
	LRA	PB	++	−−
	PypKa-MD, GBNeck2-CpHMD	CE	+++	−−−
	pKAI-MD	Machine Learning	+++	−

tions. A significant part of empirical methods uses simplified energy functions to describe the protein and the interactions between sites. PROPKA[149] is the most widely used, although several others have been developed such as MM-SCP[150], DEPTH [151], PKAcal [152], PKA17[153], and even a consensus approach with a combination of different methods [154]. New empirical predictors based on Machine Learning (ML) have been developed in recent years. Like all empirical methods, the ML-based are trained on experimental pK_a values. However, ML models have many more parameters and require more training data. Chen *et. al* [155] and Gokcan *et. al* [156] have used experimental pK_a values as training data. Unfortunately, experimental data is limited in terms of quantity and variety. To overcome these restrictions and train on much larger data sets, DeepKa [157], and pKAI [3] have taken advantage of synthetic data generated by constant-pH MD (MD) and a CE-based method, respectively.

The CE-based class is one of the most popular featuring the vastest amount of methods. They are not as fast as the empirical, however, they tend to be much faster than QM/MM and MD-based methodologies. The speed of the method is related to the model used to describe the interactions and also depends on the inclusion of conformational sampling. The most popular CE models are generalized Born (GB) and Poisson–Boltzmann (PB). In PB, the solvent is described by a continuum dielectric, and the protein is modeled as a rigid body with a lower dielectric constant. GB is an analytical approximation of the linearized PB equation, which allows for a much faster alternative to the numerically solved PB. There are several GB variations available implemented in different software packages. For example, in AMBER it is possible to use GB/HCT[158], GB/OBC[159], GBneck[160] or GBNSR6[161], and in CHARMM the options include GBMV[162], GBSW[163]. pK_a predictors based on PB usually couple a previously developed PB solvers, like DelPhi [164], APBS [165] or MEAD [166], with a Monte Carlo routine to sample ionization states [167]. DelPhiPKa [168] can be used as a CLI, others like H++ [169] provide a web server, and PypKa [1] also offers a python API. Most model the conformational flexibility of the protein by increasing or modulating its dielectric constant. Some can sample proton tautomers [170] and others side-chain rotamers [171].

A more thorough conformational sampling can be done by using MD. Conformational averaging can help to improve predictions in residues whose environment is poorly described by a single average structure, such as an experimental one. However, one should be careful in this endeavor as in standard MD, only one protonation microstate is being simulated, and for example, HEWL can transition between 24 protonation microstates at pH 7 [172]. In MM-GBSA [173], this is not taken into consideration which means that the ensemble of conformations sampled is biased. The linear response approximation (LRA) approach partially addresses this issue by using conformations of both the protonated and deprotonated forms of the residue for which the pK_a is to be calculated [174, 175]. A linear dependency between the two extreme cases can be assumed if the two environments do not change dramatically. However, this method is not suitable for dealing with multiple or interacting residues since the number of simulations quickly escalates. To obtain the correct populations of microscopic states and deal with the inter-dependency between conformations and protonation states, it is necessary to couple their sampling. CpHMD methodologies periodically reassess the protonation state of the conformational sampling simulation. For the evaluation of likely ionization states some methods use GB [176, 177] and others PB [1, 178]. At the expense of higher computational costs, these methods can be used to study pH-dependent conformational phenomena. In the near future, machine learning accelerated methods like pKAI-MD will likely reduce the computational penalty of CpHMD dramatically.

No accuracy considerations have been included in Table 1.5 intentionally because it is extraordinarily hard to evaluate the accuracy of pK_a methods. The difficulty of quantitatively characterizing the performance of the different predictors has been highlighted in the ambitious pK_a Cooperative project, the first blind prediction challenge for pK_a values in proteins [179, 180]. Regretfully, no additional similar initiatives have occurred. There is a need for more high-quality experimental data and more blind predictions. Furthermore, comparing rigid body methodologies is particularly complicated due to their dependence on the representativity of the input structure, which is almost always unknown.

In essence, the error of each class of methods is system-dependent, and the choice of one over the other should be examined on a case-by-case basis as each one has its strengths and weaknesses. QM-based models should be preferred when dealing with metal atoms or modeling chemical reaction mechanisms. Empirical methods provide cheap and competitive predictions for time- or resource-constrained situations. Albeit more computationally expensive, CE-based are a cost-effective solution for estimating microscopic equilibrium constants, pK_{half} and macroscopic pK_{a}. To study pH-dependent conformational transitions, CpHMD is the go-to methodology.

1.3 Protonation equilibria from continuum electrostatics

One of the most popular classes of pK_a predictors describes the system with CE. In these models, a solvated protein may be described as a set of point charges in a low dielectric surrounded by a high dielectric region (Figure 1.7).

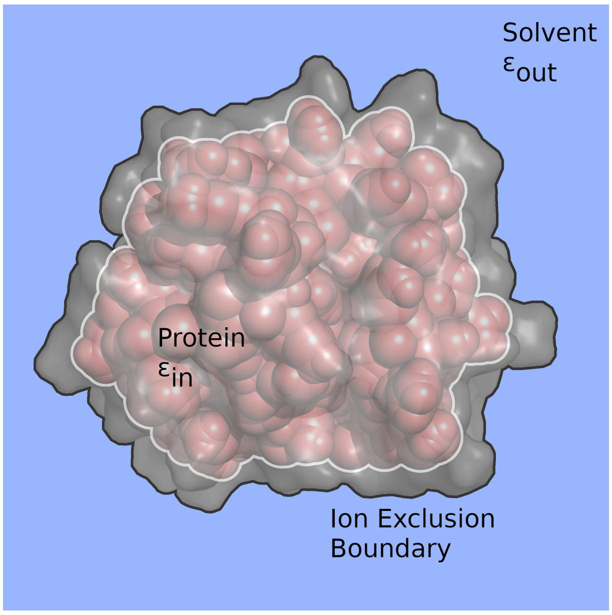


Figure 1.7: Continuum electrostatic model of a protein.

The dielectric constant of the water ε_{out} is usually set to 80, and that of the protein ε_{in} to a value between 2 and 20. In a more realistic setting, the dielectric constant of water should reduce in proximity of the solute; however, the approximated bulk value is commonly used. In contrast, it has been observed that using for the solute the dielectric value corresponding to electronic polarizability, i.e., around 2, leads to worse agreement with experiments since larger ε_{in} values can actually compensate for factors not explicitly considered in a given model [181]. The dielectric constant is a measure of the ability of a medium to react to an applied electric field. Electronic polarization alone can generate a dielectric constant of ≈ 2 and reduce the electrostatic potential by the same factor relative to vacuum [182]. Water also has orientational polarizability and thus produces a much stronger screening. Its ability to quickly adapt to changes in the local environment makes water an efficient screener of electrostatic interactions. Parts of a protein may also reorient in response to an electric field, although not as much as water. To explain the experimental polarizability of a dry protein, an average dielectric constant of around 3 is needed [183]. However, this screening ability depends on the conformation. Furthermore, the dielectric constant is not uniform across all regions of a protein, as the interior of a protein is less flexible to reorganization and has a much lower dielectric. To account for this, different protein regions may be assigned specific dielectrics [184] as ε_{in} can be modulated based on local properties such as solvent exposure or atomic density [185, 186]. In these models, the values of ε_{in} may range around 6–7 in buried regions and 20-30 at the surface of the protein. Some models explicitly consider the conformational rearrangement of the side chain [171, 187], and thus require lower dielectric constants (4–8). In methods such as CpHMD, in which full conformational flexibility is included, an ε_{in} of 2 should be used to account for the electronic polarization effect. When all contributions are accurately and explicitly treated, an ε_{in} of 1 should be used [181].

In a way, all CE methods for proteins stem from the Poisson equation for inhomogeneous media,

$$\nabla \cdot [\varepsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r})] = -4\pi\rho(\boldsymbol{r}) \tag{1.20}$$

where ε, ϕ and ρ are the dielectric constant, electrostatic potential and charge density at point $\boldsymbol{r}$. This equation explains the observable relation between polarization and electric field. However, it does not account for the distribution of ions in a solution with electric potential. To include this effect, we need to consider that $\rho(\boldsymbol{r})$ depends on the charge density of the molecule $\rho(\boldsymbol{r})^{mol}$ and mobile ions $\rho(\boldsymbol{r})^{ion}$,

$$\nabla \cdot [\varepsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r})] = -4\pi \left[\rho(\boldsymbol{r})^{mol} + \rho(\boldsymbol{r})^{ion}\right] \tag{1.21}$$

$$\rho(\boldsymbol{r})^{ion} = \sum_{i}^{K} ez_i c_i(\boldsymbol{r}) \tag{1.22}$$

where K is the number of ion types in solution, z_i is the charge number of the i ionic species, and e is the elementary charge (i.e., the charge of a proton). By assuming that the local ionic concentration $c_i(\boldsymbol{r})$ follows a Boltzmann distribution depending only on the electrostatic potential,

$$c_i(\boldsymbol{r}) = c_i^{\infty}\lambda_i(\boldsymbol{r})\exp\left(\frac{-z_i e\phi(\boldsymbol{r})}{k_B T}\right) \tag{1.23}$$

The bulk concentration of the i ion c_i^{∞} represents its concentration at an infinite distance from the protein, where its electrostatic potential is no longer felt. The Boltzmann constant and the system's temperature are represented by k_B and T, respectively. The ion-accessibility parameter $\lambda_i(\boldsymbol{r})$ is equal to 0 in the region excluded for mobile ions and 1 elsewhere. The ion exclusion region, or Stern layer, in which no mobile ions are present, has a thickness corresponding to the hydrated radius of the ion [188]. Substituting Eqs. 1.22 and 1.23 in 1.21, we get the nonlinear Poisson-Boltzmann equation,

$$\nabla \cdot [\varepsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r})] = -4\pi \left[\rho(\boldsymbol{r})^{mol} + \sum_{i} ez_i c_i^{\infty}\lambda_i(\boldsymbol{r})\exp\left(\frac{-z_i e\phi(\boldsymbol{r})}{k_B T}\right)\right] \tag{1.24}$$

When the electrostatic potential is weak, the exponential may be approximated for small arguments. By applying this approximation and assuming a neutral bulk solution, the linearized Poisson-Boltzmann equation, in the case of a monovalent salt can be obtained,

$$\nabla \cdot [\varepsilon(\boldsymbol{r})\nabla\phi(\boldsymbol{r})] = -4\pi\rho(\boldsymbol{r})^{mol} + \frac{8\pi e^2 I}{k_B T}\lambda(\boldsymbol{r})\phi(\boldsymbol{r}) \tag{1.25}$$

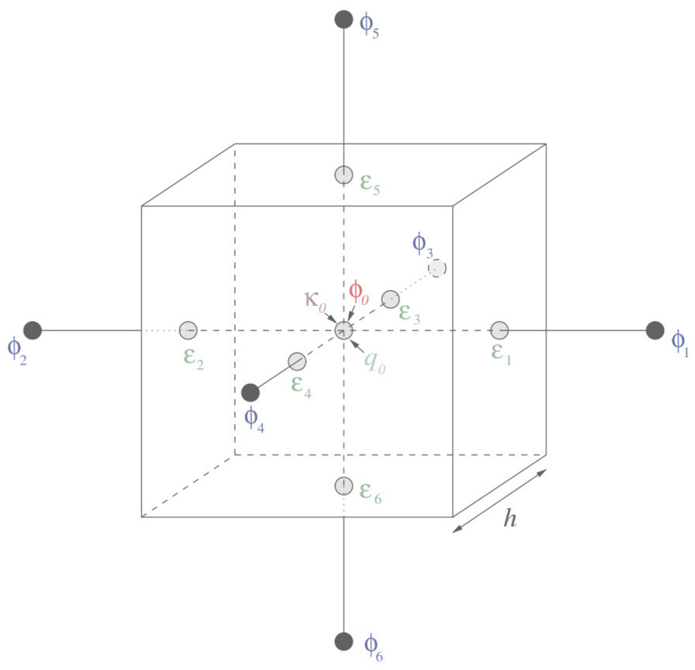


Figure 1.8: Scheme of a grid cell and necessary quantities to solve the PB equation with the finite-difference method (Equation 1.27). Adapted from reference 191.

$$I = \frac{1}{2}\sum_i c_i^{\infty} z_i^2 \tag{1.26}$$

Note that when the ionic strength I is zero, the Poisson–Boltzmann equation reverts to the original Poisson equation. The linearized PB equation can be used for systems with low local charge density. While some proteins fit the criterion, particular attention must be paid not to apply this approximation to more charged systems such as charged membranes and nucleic acids.

There are several strategies to solve the linear PB equation, including analytical solutions for simple geometries [189]. However, solving for a protein in 3 dimensions is more complex, and a numerical approach is required. The finite-difference method is arguably one of the most popular techniques, and the one implemented by DelPhi [164], MEAD [166] and APBS [165]. Other strategies are possible, including the boundary element, finite element, or the adaptive-grid method. Here we will focus solely on the finite-difference method, and a more in-depth review of the different numerical methods for biophysical applications can be found in reference 190.

To numerically solve the PB equation, all properties of the system (charge q, dielectric constant ε and ion accessibility λ, electrostatic potential ϕ) need to be mapped onto a grid (Figure 1.8). The finer the grid (smaller grid spacing h), the better the accuracy of the finite-difference approximated solution. In order to assign ε and λ values, the surface of the molecule and the ion exclusion layer need to be calculated. These are usually defined by rolling a ball over the atoms of the protein. A probe of 1.4 Å representing the water molecule is usually used for the molecule surface, and the ion exclusion layer is calculated assuming a width of typically 2.0 Å representing the Cl$^-$ ion [192]. The protein atoms are represented as spheres with given radii depending on their type. While the charges can be obtained directly from a force field such as CHARMM, AMBER, or GROMOS, the radii need to be derived. A common procedure is to set the atomic radius equal to half the distance between the minimum Lennard–Jones (LJ) interaction energy R_{min} between two identical atoms [193–195]. Alternative approaches include using half of σ, the distance corresponding to zero LJ energy R_{0RT} [196], or half the thermal radius R_{1RT}, the distance corresponding to an RT increase in the LJ energy relative to the minimum [197], or even, double the thermal radius R_{2RT} of the interaction energy of an atom with water [198]. Ultimately, the choice of charges and of the corresponding radii is empirical, and optimizing parameters to reproduce solvation energies or pK_a values with PB is a regular practice [198–200].

The molecular surface is then used to map the ε_{in} and ε_{out} values, and the Stern layer for the λ function. The point charges of the protein atoms are smeared over the surrounding grid points by linear interpolation. After calculating an initial guess for the potential for each grid cell (for example, from an analytical solution), the electrostatic potential at each cell center can be calculated by imposing that the following equation is satisfied within some accuracy [201],

$$\phi_0 = \frac{\sum_i^6 \varepsilon_i \phi_i + 4\pi \frac{q_0}{h}}{\sum_i^6 \varepsilon_i + \kappa_D^2 \varepsilon_0 \lambda_0 h^2} \tag{1.27}$$

$$\kappa_D^2 = \frac{8\pi e^2 I}{\varepsilon_0 k_B T} \tag{1.28}$$

This equation states that the electrostatic potential ϕ in each grid cell depends on the potential of the 6 neighboring grid cells and the corresponding dielectric constant on the face of the two cells, as well as its own charge q_0, Debye screening constant κ_D, and dielectric constant ε_0. The finite-difference solution can be seen as a set of equations that can be solved in an iterative fashion, where each grid node can be defined by Equation 1.27. The exception to the rule is the treatment of the boundary cells of the lattice whose initial potential is kept fixed. There are several possibilities for obtaining the boundary potentials. One of the simplest solutions is to consider the protein as a sphere and use an analytical solution to the linearized PB equation for a sphere [202]. A more accurate approach involves running a first calculation with a larger and coarser grid and then using the resulting potential to set up the values for the boundary cells in a successive calculation. This procedure is called focusing and can be repeated multiple times until the desired boundary potential for the finer grid is reached. In this way, it is possible to calculate the potentials for a part of the system, such as one titratable residue, with high accuracy using a small h (usually 0.25 Å) grid and still consider the remaining protein atoms in the calculation without the need to compute a large grid with the same level of detail which in some cases would be impossible due to memory limitations.

The electrostatic potential $\phi(r)$ given by the PB equation can be divided into two parts: the Coulombic potential and the reaction field potential $\phi(r)_{rf}$,

$$\phi(r) = \sum_{i}^{M} \frac{q_i}{4\pi\varepsilon_{in}\varepsilon_0|r - r_i|} + \phi_{rf}(r) \tag{1.29}$$

The reaction field potential always contrasts the field generated by the fixed partial charges of the protein. The reaction field energy is the interaction energy of the set of free charges q with the polarization charge at the boundary between the different media, and with the mobile ions in solution, screened by the dielectric of water. This energy is an essential component of the solvation free energy and is given by,

$$\Delta G_{rf} = \frac{1}{2}\sum_{i} q_i \phi_{rf}(r_i) \tag{1.30}$$

For a single ion of radius a and charge q, it is possible to find the potentials analytically. Born described the free energy of transferring a single charged sphere to a water environment described as a continuum dielectric ε_{out} [203],

$$\Delta G_{solv} = -\frac{q^2}{2a}\left(1 - \frac{1}{\varepsilon_{out}}\right) \tag{1.31}$$

The Born formula is an exact solution of the Poisson equation for a spherical solute. One can generalize this equation to a collection of atoms and define the solvation free energy as the sum of individual Born terms, and pairwise Coulombic terms [204]. The generalized Born (GB) equations try to replicate the physics of the Poisson equation for complex molecular geometries, like those of proteins, as an analytical formula [159],

$$\Delta G_{solv} \simeq -\sum_{i} \frac{q^2}{2R_i}\left(1 - \frac{1}{\varepsilon_{out}}\right) - \frac{1}{2}\sum_{ij,i\neq j} \frac{q_i q_j}{f_{ij}^{GB}(r_{ij}, R_i, R_j)}\left(1 - \frac{1}{\varepsilon_{out}}\right) \tag{1.32}$$

where R_i is the effective Born radii of atoms i, r_{ij} is the distance between atoms i and j, and the GB kernel f_{ij}^{GB} is a function that defines, what can be considered as the effective interaction distance between atoms i and j,

$$f_{ij}^{GB}(r_{ij}, R_i, R_j) = \sqrt{r_{ij}^2 + R_i R_j \exp\left(\frac{-r_{ij}^2}{4R_i R_j}\right)} \tag{1.33}$$

This particularly popular GB kernel [205] has been referred to (in conjunction with Eq.1.32) as the canonical GB model [206]. The effective Born radii can be estimated or parameterized in a number of ways [158, 159, 205, 207, 208].

By considering the approximation that is made by GB models that $\Delta G_{rf} \approx \Delta G_{GB}$, and comparing Equations 1.30 and 1.32, this means that the Born radii should have values that would give the same electrostatic solvation free energy without calculating costly electrostatic potentials.

$$\begin{tikzcd}
AH^{sol} \arrow[r, "\Delta G^{sol}_{AH\rightarrow A}"] \arrow[d, "\Delta G^{sol\rightarrow P}_{AH}"'] & A^{sol}+H^{+} \arrow[d, "\Delta G^{sol\rightarrow P}_{A}"] \\
AH^{P} \arrow[r, "\Delta G^{P}_{AH\rightarrow A}"'] & A^{P}+H^{+}
\end{tikzcd}$$

Figure 1.9: Thermodynamic cycle of titratable site *A* in protein *P* and solvated *sol* environments.

Over the years, both PB and GB have found success as CE descriptors of solvated macromolecules to estimate free energies of solvation, binding energies between protein-protein or ligand-protein systems, and even pK_{a} values. Bashford and Karplus have laid out the foundation for pK_{a} estimations of ionizable groups in proteins using CE models [209], with the thermodynamic framework ground in the intrinsic pK_{a} (pK_{intr}, Eq. 1.42) still being valid and widely used nowadays [1, 171, 187, 210, 211]. According to the thermodynamic cycle seen in Figure 1.9, the protein site A protonation free energy can be defined as,

$$\Delta G^{P}_{AH\rightarrow A} = \Delta G^{sol}_{AH\rightarrow A} + \Delta G^{sol\rightarrow P}_{A} - \Delta G^{sol\rightarrow P}_{AH} \tag{1.34}$$

Since a free energy of protonation can be converted to pK units by,

$$\mathrm{p}K = \frac{\Delta G}{2.3k_BT} \tag{1.35}$$

one can rewrite Equation 1.34 for the pK_{a} of a site in a protein having all other sites in a fixed state as,

$$\begin{align}
\mathrm{p}K^{P}_{\mathrm{a}} &= \mathrm{p}K^{sol}_{\mathrm{a}} + \frac{1}{2.3k_BT}\left[\Delta G^{sol\rightarrow P}_{A} - \Delta G^{sol\rightarrow P}_{AH}\right] \tag{1.36}\\
&= \mathrm{p}K^{sol}_{\mathrm{a}} + \frac{1}{2.3k_BT}\left[\Delta G^{P}_{AH\rightarrow A} - \Delta G^{sol}_{AH\rightarrow A}\right] \tag{1.37}
\end{align}$$

The pK^{sol}_{a} is often called the pK_{a} of the model compound (pK_{mod}), as in reality, A^{sol} does not correspond exactly to the entire amino acid in water. Also, if we consider that free energy of protonation can be defined by,

$$\begin{align}
\Delta G^{P}_{AH\rightarrow A} &= \Delta G_{quantum} + \Delta G^{P}_{rf} + \Delta G^{P}_{back} + \Delta G^{P}_{interact} \tag{1.38}\\
\Delta G^{sol}_{AH\rightarrow A} &= \Delta G_{quantum} + \Delta G^{sol}_{rf} + \Delta G^{sol}_{back} \tag{1.39}
\end{align}$$

where $\Delta G_{quantum}$ is the energy of quantum effects like bond formation and charge transfer that we assume to be the same in both the model compound and the protein site, ΔG_{rf} the contribution of the change in the polarization of the surrounding medium, ΔG_{back} the contribution of the interactions with permanent charges, and $\Delta G^{P}_{interaction}$ the energy of the interactions with other titratable sites. Substituting these equations into Eq. 1.37 we get,

$$\mathrm{p}K_\mathrm{a}^P = \mathrm{p}K_\mathrm{mod} + \frac{1}{2.3k_BT}\left[\Delta\Delta G_{rf} + \Delta\Delta G_{back} + \Delta G^P_{interact}\right] \tag{1.40}$$

which can be rearranged to contemplate $\mathrm{p}K_\mathrm{intr}$, the $\mathrm{p}K_\mathrm{a}$ value of a site when all others are in their reference state,

$$\mathrm{p}K_\mathrm{a}^P = \mathrm{p}K_\mathrm{intr} + \frac{\Delta G^P_{interact}}{2.3k_BT} \tag{1.41}$$

$$\mathrm{p}K_\mathrm{intr} = \mathrm{p}K_\mathrm{mod} + \frac{1}{2.3k_BT}\left[\Delta\Delta G_{rf} + \Delta\Delta G_{back}\right] \tag{1.42}$$

The pH-independent $\mathrm{p}K_\mathrm{intr}$ can be determined by adding a known (or calibrated) $\mathrm{p}K_\mathrm{mod}$ to $\Delta\Delta G_{rf}$ and $\Delta\Delta G_{back}$, both of which attainable from the PB calculations of the reference state r and protonation state s,

$$\begin{aligned}\Delta\Delta G_{rf} &= \Delta G^P_{rf} - \Delta G^{sol}_{rf} \\ &= \left(G^P_{rf}(\mathrm{r}) - G^P_{rf}(\mathrm{s})\right) - \left(G^{sol}_{rf}(\mathrm{r}) - G^{sol}_{rf}(\mathrm{s})\right)\end{aligned} \tag{1.43}$$

$$\begin{aligned}\Delta\Delta G_{back} &= \Delta G^P_{back} - \Delta G^{sol}_{back} \\ &= G_{back}(\mathrm{r})^P - G_{back}(\mathrm{s})^P \\ &= \sum_i^K q_i(\mathrm{r})\phi_i(\mathrm{r}) - \sum_i^K q_i(\mathrm{s})\phi_i(\mathrm{s})\end{aligned} \tag{1.44}$$

where K is the number of atoms that do not belong to titratable sites.

So far, we considered only one titrating site at the time, while the others were fixed. While it might be possible to obtain microscopic $\mathrm{p}K_\mathrm{a}$ values from this approach, probing all possible microscopic states is unfeasible for most proteins. A possible solution is to sample protonation states with Monte Carlo simulations [167]. To do that, one needs to define the pH-dependent energy shift due to a protonation state change in site n of a protein with N titratable sites,

$$\Delta\Delta G_{n1\to n2} = 2.3k_BT\left[a_{n1}\gamma_{n1}\left(pH - \mathrm{p}K_\mathrm{intr}(n2)\right) - a_{n2}\gamma_{n2}\left(pH - \mathrm{p}K_\mathrm{intr}(n1)\right)\right] + \sum_{j\neq n}^{N} \Delta G_{n1\to n2,jx} \tag{1.45}$$

where a_n and γ_n are the ionization state (0 for neutral; 1 for ionized) and the charge (+1 for cationic; -1 for anionic) of site n. The effect of the protonation state change from $n1$ to $n2$ in the interaction energy with site j in a fixed state x is given by,

$$\Delta G_{n1\to n2,jx} = G_{n2,jx} - G_{n1,jx} \tag{1.46}$$

$$G_{n2,jx} = \frac{\sum^I (q_n(r) - q_n(n2))(\phi_j(r) - \phi_j(jx)) + \sum^H (q_j(r) - q_j(jx))(\phi_n(r) - \phi_n(n2))}{2} \tag{1.47}$$

$$G_{n1,jx} = \frac{\sum^I (q_n(r) - q_n(n1))(\phi_j(r) - \phi_j(jx)) + \sum^H (q_j(r) - q_j(jx))(\phi_n(r) - \phi_n(n1))}{2} \tag{1.48}$$

where I and H are the atoms of sites n and j, respectively.

Since all terms can be obtained from the PB calculations, Eq.1.45 can be used to sample different ionization states for the protein according to the Metropolis criterion [212],

$$P_{acceptance}(n2) = \begin{cases} 1 & \text{if } \Delta\Delta G_{n1\rightarrow n2} < 0 \\ e^{-\Delta\Delta G_{n1\rightarrow n2}/k_B T} & \text{otherwise.} \end{cases} \tag{1.49}$$

By randomly evaluating new protonation states for different sites, and after some equilibration steps, physically relevant protonation states are sampled. Furthermore, it is possible to extend this approach to include proton isomerism [170] or rotamers [213]. However, this algorithm is inefficient for strongly interacting sites [167]. Since only one site is allowed to change its protonation state at each MC step, the MC time-independent trajectory may get trapped in local minima. To improve the sampling efficiency in these cases, it is necessary to consider new states for two sites in the same step.

The observed microscopic protonation states observed in equilibrated MC trajectories for a range of pH values allow for macroscopic titration curves for individual sites to be constructed and pK_{half} values to be estimated. In fact, from these simulations, it is possible to estimate all microscopic and macroscopic constants. However, in this section, we have neglected the impact of conformational changes. The MC resulting ionization state populations assume a rigid structure and may be significantly inaccurate.

1.4 Constant-pH Molecular Dynamics

Constant-pH molecular dynamics (CpHMD) methods have emerged as a better way to describe pH-dependent phenomena by coupling protonation and conformational sampling. With these methods, it is possible to provide mechanistic insights about pH-dependent processes such as conformational transitions induced by a pH change. Over the last two decades, different groups have developed competing strategies on how to deal with pH effects in MD simulations [214–217].

The type of protonation state representation is an important characteristic of CpHMD methods, which can be used to distinguish them between those using discrete protonations and the methods that use continuous states. One of the earliest CpHMD methods with fractional protonation states dates back to 1997 when Baptista *et al.* published the implicit titration method [218]. However, most methods using continuous protonation states are based on the work of Brooks III and co-workers [219, 220] and use λ-dynamics [221] to sample conformations and protonation states simultaneously. Each titratable site is assigned a λ particle with an artificial mass (typically between 5 and 10 atomic units) that takes values between 0 (protonated state) and 1 (deprotonated state). These fictitious particles affect the whole system, and as such, the Hamiltonian includes their kinetic energy, and the electrostatic and van der Waals energies between the titratable group and its environment are scaled according to the value of λ. The energy function applied to a λ particle depends on the solvent pH, the reference pK_a value of its site in water, and the interactions with the environment. On top of that, one can add a tunable biasing potential to avoid sampling non-physical states [222]. By simulating a series of λ values, the pK_a value of each site can be obtained by thermodynamic integration. The original implementation of this method was done in the CHARMM software package [223] and used a GB model for both proton titration, and conformational dynamics [219]. Since then, several new variations and upgrades have been made to the λ-dynamics CpHMD. Some of them have been implemented in Amber [176, 224] and GROMACS [225] software suites, and extensions have been made to deal with systems other than proteins like membrane proteins [226] and nucleic acids [227]. A hybrid-solvent approach was developed in which conformational states were influenced by explicit solvent while protonation states sampling was done in GB solvent [228]. Explicit solvent treatment for both protonation and conformational dynamics was developed [229] and even titratable water [230] was introduced.

The strategy of continuous protonation states is strongly affected by convergence problems. To speed up the sampling convergence of both protonation and conformation, enhanced sampling methods, such as temperature [231] or pH replica-exchange [232], have been used. Nevertheless, these CpHMD methods inherit the fundamental limitations of their predecessors and do not offer a solution for dealing with biomolecular systems with a large number of titratable sites. Furthermore, when dealing with a small number of sites, there are alternative ways of calculating pK_a values and modeling the coupling between the conformational and protonation spaces. Performing free energy perturbation (FEP) calculations of different protonation configurations is doable for a small number of sites. Likewise, the linear response approximation (LRA) approach may also be a suitable option. A coarse-grained (CG) MD simulation might be useful for extremely large systems. To this effect, Marrink's group has developed a CpHMD method for the Martini force field with the distinctive feature of using transferable titratable beads [233]. Others have developed a CpHMD for OPEP6 – an OPEP CG force field – based on a fast proton titration scheme [234]. Another attempt was made by Reilley *et. al*, extending CG MD with empirical (PROPKA) pK_a predictions [235]. Unfortunately, by using PROPKA, which was trained to reproduce experimental pK_a values, this method samples from the wrong ensemble.

Discrete protonation states provide a more scalable scheme by coupling protonation sampling, using MC, with MD-generated conformations. The stochastic titration model introduced by Baptista *et al.* in 2002 [178] is a pivotal work in this field. The workflow of this method is a loop that can be divided into three steps: (I) a protonation state is chosen based on PB/MC calculations. It should be noted that the ionization state of the protein that is chosen is not the most likely, but rather one that is taken randomly from the equilibrated sampled distribution; (II) a short MD simulation is performed so that the water molecules may accommodate the new charge state of the protein which remains frozen. This is a pragmatic way of stabilizing the simulations and minimizing energy spikes at the water/protein interface. Of this trajectory, only the last conformation is kept as the starting conformation for the next step; (III) an effective MD trajectory is simulated and concatenated with previous production MD segments. The final conformation proceeds to the next loop, starting with the determination of a new protonation state at step I. This method was originally implemented using the GROMOS force field to be applied to peptides and proteins. In the last 20 years, it has then been adapted by different groups to suit their needs. Baptista's lab has applied CpHMD to study dendrimers [236, 237], protein-membrane processes [238] and has implemented a pH gradient across lipid membranes [239]. Machuqueiro's group fork of the stochastic titration has been extended to titrate lipids [240, 241] and coupled with enhanced sampling methods such as a pH replica exchange (RE) scheme [4] and umbrella sampling [242]. Antosiewicz and co-workers have implemented it in CHARMM and accelerated the MD part by using a continuum solvent model [243, 244]. Mongan *et. al* have used GB in both the conformational and protonation sampling and used the AMBER force field [177], for which Roitberg's group has developed a temperature RE [245] as well as pH RE [246]. It should be noted that the pH RE scheme for the discrete protonation states used in GROMOS and AMBER had been previously developed for the CHARMM force field by Itoh *et. al* [247]. Another interesting approach to discrete CpHMD has been developed at Roux's group [248]. In the hybrid non-equilibrium MD/Monte Carlo, a new configuration is evaluated with the Metropolis criterion after a new protonation state produces a short non-equilibrium MD trajectory. This method has been implemented in the NAMD software package and implemented in CHARMM [249].

1.5 Machine Learning

In recent years, we have witnessed large transformations in a wide array of scientific fields thanks to machine learning (ML). From physics, [250] and biology [251, 252] to sociology [253] and genetics [254], ML has enabled a number of technological leaps. It is nowadays an integral and irreplaceable tool not only for research but also in our everyday lives. Natural language processing (NLP) [255], computer vision [256], speech recognition [257] and recommendation systems [257] are one of the first and more popular technologies that have been drastically improved by ML. Researchers and engineers are applying ML to tackle virtually every existing problem. However, not all ML methods are appropriate or applicable to encode every type of question. Different approaches might be considered depending on the type of data, complexity of the problem, and objective. Most approaches can be divided into supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the main goal is to train a function that can map an input into an output. This function is learned by mimicking examples that are provided, and a successful model is one that can generalize to examples not available. Unfortunately, sometimes labeled input-output pair examples are hard or even impossible to obtain. Furthermore, there are questions for which a good answer tells us about the relationship between the data. Anomaly detection methods like Isolation Forest, clustering algorithms such as K-means, or dimensionality reduction methods such as principal component analysis (PCA) are all popular examples of unsupervised ML. In between supervised and unsupervised learning, there is semi-supervised learning in which only a small part of the data is labeled, and also self-supervised learning, where the goal is to predict a hidden part of the input. Masked language models such as BERT [258], or RoBERTa [259] are quite successful in NLP and involve predicting a hidden part of a text given the remaining words. The same training philosophy can be applied to videos by masking frames [260, 261], or even to protein sequences, predicting hidden amino acids [262]. Finally, in reinforcement learning, an agent is trained to navigate a complex space using a set of allowed actions while being rewarded for achieving certain objectives and penalized for bad behavior [263]. Self-driving cars are a good example of this kind of ML technology. The ML agents are taught to drive by receiving rewards for staying in the intended lane and penalized for going off the road, crashing into other cars, and running over pedestrians or cyclists. With an estimated 10^{30} and 10^{60} potential drug-like molecules [264], it is certainly appealing to apply this method to search for viable and target-specific drugs in this immensely complex and vast chemical space [265].

Here, we will focus on supervised learning as this thesis revolves around predictive methods. The simplest way to describe an artificial neural network (ANN) is by defining our model as a function f where $\hat{y} = f(x)$ that best approximates y given a set of examples x for which there is a known y target value. In the context of protein pK_a predictions, y is usually the pK_a value of a residue described by x. The different existing types of ML models are variations of this f function. Linear models, such as linear regressions, are a simple form of formulating f that is fast to train, though very limited, as they cannot capture the complex nonlinear relations in high-dimensional data. However, it is possible to extend a linear model to describe nonlinear phenomena by applying a nonlinear transformation $g(f(x))$ to it, also known as the activation function. Commonly used activation functions include sigmoid, softmax, and hyperbolic tangent, although the ReLu function has become the recommended default for most ANNs [266]. By stacking n nonlinear transformations of linear models, one obtains a multilayer perceptron (MLP) where

$$\mathbf{y} = g^{(n)} \left(\mathbf{W}^{(n)\top} \mathbf{h}^{(n-1)} + \mathbf{b}^{(n)} \right) \tag{1.50}$$

in which a hidden layer $\mathbf{h}$ of depth i is defined by a weight matrix $\mathbf{W}$ and bias $\mathbf{b}$,

$$\mathbf{h}^{(i)} = g^{(i)} \left(\mathbf{W}^{(i)\top} \mathbf{h}^{(i-1)} + \mathbf{b}^{(i)} \right) \tag{1.51}$$

In MLPs, also named feedforward neural networks (FNN), the size of a layer is known as the width, while the number of layers of the network is referred to as its depth. This nomenclature explains the origin of the classification of networks with many layers as deep learning models. FNNs are one of the simplest types of ANNs, and unlike other types, such as recurrent neural networks (RNN), there are no feedback connections, and the information flows in a single direction during forward propagation to output a prediction value $\hat{y}$. In order to obtain a prediction that resembles the true value y, it is necessary first to evaluate its quality. This is the main purpose of defining a loss function $L(\hat{y}, y)$. The choice of the loss function is quite important as it defines the complexity of the problem we want to minimize. The type of problem one wants to solve with an ANN dictates the available loss functions. The mean square error (MSE), or L2 loss, is frequently used in regression problems. In contrast, the cross-entropy function or the negative log-likelihood (NLL) loss are useful to solve classification problems [267]. The average value of the loss function over all the training examples is often named cost function J. Training a neural network means finding values of $\mathbf{W}$ and $\mathbf{b}$ that minimize J, and ideally, after the last epoch, reach a value close to the global minimum of the high dimensional surface. Computational chemists and biophysicists may gain some intuition about this process by thinking of it as analogous to typical energy minimization or geometry optimization procedures. However, instead of performing a gradient descent optimization to update molecular coordinates in ML, adjustments are made to model parameters. The most widely used optimization algorithms are stochastic gradient descent (SGD) [268], and Adam [269]. Regardless of the algorithm, it is always necessary to obtain the derivative of J with respect to the parameters W, which can be obtained by the back-propagation algorithm, which efficiently computes the gradient by applying the chain rule of calculus.

In many situations, it is possible to successfully train a model to faithfully reproduce the training examples without the model having any predictive ability on new examples. This quite common issue is termed overfitting, and there are several possible regularization strategies to tame it [270]. Arguably, one of the most important and overlooked ways of reducing overfitting is to choose an appropriate network configuration. For example, if two models display equivalent performances, the smaller one is less prone to overfit. Using dropout is also an effective regularization technique used in several domains [271]. Randomly dropping weights during training breaks up co-adaptations and increases the robustness of the network. Batch normalization extends the concept of input feature normalization and applies it to hidden layers [272]. The mean and standard deviation standardization is done per mini-batch and helps speed up learning, makes the training less sensitive to initialization, and reduces overfitting. Early stopping is another powerful regularization algorithm that involves stopping training if the performance on the validation set is no longer improving. An ensemble method employs a model averaging strategy in which different models (with different architectures or of different types) are trained separately and used to make predictions. Leaving dropout turned on during inference is a simple way of creating an ensemble method with FNNs. It is important to note that regularization techniques can degrade the model's accuracy on a particular validation set. One must decide how much performance one is willing to trade for a supposedly more robust model. A multi-task learning setting can improve the performance on a particular task we are interested and do not have that much labeled data accessible. Furthermore, even if it does not, by training a model on distinct tasks, its representation of the input object becomes enriched. This representation can then be helpful to apply to other problems for which data is scarce, an exercise called transfer learning.

Encoding the problem in an ML-compatible way is a crucial step to having an effective model. Traditional ML algorithms require hand-crafted features extracted from the raw input, be it a picture or a protein structure. However, these features are subject to bias in the knowledge of their creator and can leave out valuable information. On the other hand, deep learning models can be applied directly to more detailed representations and automatically learn features during training. In structural biology, proteins have long been depicted as graphs where nodes represent atoms, and their covalent bonds are edges. Hence, graph neural networks (GNN) [273] are a natural choice for molecular tasks. Under the message passing framework notation, it is possible to describe a graph convolutional layer l as a collection of node embeddings $\mathbf{h}_i$ that are the result of aggregating messages $\mathbf{m}_i$ from connected nodes,

$$\mathbf{m}_{ij} = \phi_e\left(\mathbf{h}_i^l, \mathbf{h}_j^l, a_{ij}\right) \tag{1.52}$$

$$\mathbf{m}_i = \sum_{j \neq i} \mathbf{m}_{ij} \tag{1.53}$$

$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l, \mathbf{m}_i\right) \tag{1.54}$$

where ϕ_h and ϕ_e are the node and edge operations, respectively, and a_{ij} is an edge attribute between nodes v_i and v_j. As seen in the equations, the node and edge information flow to nearby nodes, and the more layers in the network, the more convoluted is the information in each node embedding. These embeddings represent the nodes, which can be atoms or protein residues and may then be used as input for traditional ML methods like support vector machines (SVM) or FNNs. However, when working with molecular coordinates and 3D GNNs, it is necessary to consider that an infinite number of graphs can represent the same molecular conformation. Fortunately, GNNs are permutation equivariant networks, which means that if the order of our atoms is different, their output node embedding will be preserved at their new position in the graph. Unfortunately, not all GNNs are equivariant with respect to other transformations like translations and rotations. To obtain predictions that are equivariant for rotations, translations, reflections, and permutations, one can use E(n)-Equivariant Graph Neural Networks (EGNNs) [274].

Data-related steps such as gathering, preparation, wrangling, and analysis are a quintessential part of the ML life-cycle. Naturally, data availability is a vital determinant of the success of an ML project. In many cases, data is unavailable and simply impossible to collect with current technology. For example, it would be very informative to have a detailed composition of different cell types of the human body over time, annotated with external factors of an individual's life (i.e., eating, sleeping, and exercising habits). Not to mention the privacy and ethics of such a data collection endeavor, this kind of data is nowadays unobtainable. Nonetheless, data collection activities are at all-time highs and are only expected to grow in the coming years since researchers and companies worldwide are aware of their usefulness. In natural sciences, this trend is also noticeable, with a number of databases available with ever-growing amounts of data: PubChem contains information regarding over 100M chemical structures [275]; GenBank has more than 200M genetic sequences, that number has doubled approximately every 18 months [276]; at UniProtKB/TrEMBL it is possible to get information regarding 200M protein sequences [277]; Protein Data Bank (PDB) has just surpassed the 190k mark of experimental protein structures deposited [278]. Despite the vast amount of data available, it is not nearly enough for some applications. For example, while there are almost 200k structures in the PDB, less than half correspond to non-redundant proteins, only around 5k structures exist for antibody-antigen complexes [279], and even less (1.5k) of RNA molecules. For some proteins, it might be possible to reconstruct them with homology modeling, requiring only that there are known structures with a reasonable degree of sequence similarity. Furthermore, ML models, such as AlphaFold, have surpassed the accuracy of traditional methods at predicting 3D structures of proteins [252, 280]. Recently, a database of AlphaFold generated structures for several proteomes has been released [281, 282]. In the realm of pK_a values, there is a grimmer scenario, with only 1.5k experimental data points collected in the most extensive database [95] and no significant updates since its publication.

Despite the data shortage, several groups have developed ML models for pK_a predictions. Chen *et al.* have trained four tree-based machine learning models with experimental pK_a values [155]. The eXtreme Gradient Boosting (XGBoost) algorithm yielded a very low RMSE, outperforming PROPKA by 37% and DelPhiPKa by 15%. The model developed by Gokcan *et al.* was also trained with experimental pK_a values. However, instead of using hand-crafted features, it uses pre-trained embeddings as atomic quantum mechanical representations [156]. This model also beats PROPKA and DelPhiPKa in addition to the Null model which consists on the set of known pK_a value in water for each residue type. However, considering the severely reduced training set (e.g. less than 25 examples for Cys and Tyr) and that there are many similar proteins in the experimental data set, there is a clear risk of overfitting in both models. Hence, it is not easy to evaluate if the performance reported is an accurate depiction of the model's predictive ability. Cai *et al.* took an arguably more robust approach and trained their convolutional neural network (CNN) on pK_a derived from GB-based CpHMD simulations [157]. Furthermore, they excluded proteins with similar sequences from their test and validation sets. It was shown that this model performs better than PROPKA on the selected test set and has an RMSE similar to that of CpHMD. Still, the correlation between DeepKa and CpHMD is not perfect ($R^2 = 0.79$), and the RMSD between both predictions is also quite high (0.79). All existing ML pK_a predictors are targeting experimental pK_a values, and there are currently no models that can predict titration curves or microscopic pK_a values, which are useful for, among other things, accelerating CpHMD simulations.

1.6 Scope and Goals

The main goal of this Ph.D. plan is to push the boundaries of pK_a predictions by making them faster and accessible to more users. To achieve our objective, we have designed and implemented multiple methods and tools with different levels of theory. In this thesis, we present six of those projects grouped into three chapters according to their purpose:

Chapter 2 Extensible and user-friendly Poisson–Boltzmann-based predictors

Objective: Create solid foundations for future developments

- PypKa: an easy to use and extend Poisson–Boltzmann-based pK_a predictor
- PypKa-MD: a simple to use CpHMD implementation leveraging PypKa

Chapter 3 Accelerating pK_a calculations with Artificial Intelligence

Objective: Obtain speed-ups of several orders of magnitude

- pKAI: a machine learning model that predicts pK_a values with comparable accuracy to that of a physics-based method up to 1000x faster
- pKAI-MD: a graph neural network model for pK_a predictions to be integrated into a constant-pH molecular dynamics framework

Chapter 4 Beyond the command line

Objective: Provide readily available calculations

- pKPDB: a 12M pK_a values database for 120k experimental protein structures
- PypKa Server: a web app to run pK_a calculations in the cloud

Initially, we focused on developing an easy-to-use and extend Poisson–Boltzmann-based pK_a predictor, PypKa, the cornerstone of all subsequent work. By using PypKa's API, it is possible to add parallelizable pK_a calculations to existing protocols with the addition of a few extra lines of code. We then created PypKa-MD, a new stochastic titration constant-pH molecular dynamics (CpHMD), around this extremely flexible tool. In both PypKa and PypKa-MD, described in Chapter 2, usability and extensibility concerns were highly prioritized, as our objective was to create easy-to-use tools that could serve as a robust core to power future features and applications.

After its validation, a large database of 12M pK_a values, named pKPDB, was created by running PypKa over 120k proteins from the Protein Data Bank. pKPDB saves time and resources spent on repeated calculations by allowing users to instantly retrieve precomputed results. For the less computer-savvy users, a web application was made available, in which pKPDB can be queried, and pK_a calculations run in the cloud. In Chapter 4, we present these tools in greater detail.

The primary motivation for generating the pKPDB database was to facilitate the training of artificial intelligence models that could mimic pK_a predictors rooted in physics at a fraction of their computational cost. Chapter 3 is devoted to developing physics-informed deep learning models that can dramatically accelerate pK_a calculations. pKAI was the first of such a class of remarkably efficient pK_a predictors. The unrivaled speed-up offered by pKAI was seen as a solution for the hefty computational penalty of CpHMD methods. Accordingly, we have adapted pKAI to be a drop-in replacement for PypKa within the CpHMD framework.

Chapter 2

Extensible and User-friendly Poisson–Boltzmann-based predictors

2.1 PypKa: A Flexible Python Module for Poisson–Boltzmann-Based pK_a Calculations

Pedro B.P.S. Reis, Diogo Vila-Viçosa, Walter Rocchia*, and Miguel Machuqueiro*

2.1.1 Context

The first goal of this thesis was to develop a rigid body pK_a predictor that could be used as a standalone application and that could be effortlessly interfaced with other applications. Furthermore, we wanted this program to run the Poisson–Boltzmann and Monte Carlo calculations without having to write nor parse the input/output files of DelPhi [164], or PETIT [170]. Although the main motivation for this work was to obtain an API that could power an ecosystem of applications, like a new CpHMD implementation (Chapter 2.2), a significant effort was made to have a tool that could also handle experimental structures. This work was published in the Journal of Chemical Information and Modeling (DOI: 10.1021/acs.jcim.0c00718).

Abstract

The protonation of titratable residues has a significant impact on the structure and function of biomolecules, influencing many physicochemical and ADME properties. Thus, the importance of the estimation of protonation free energies (pK_a values) is paramount in different scientific communities, including bioinformatics, structural biology, or medicinal chemistry. Here, we introduce PypKa, a flexible tool to predict Poisson–Boltzmann/Monte Carlo-based pK_a values of titratable sites in proteins. This application was benchmarked using a large data set of experimental values to show that our single structure-based method is fast and has a competitive performance. This is a free and open-source tool that provides a simple, reusable and extensible Python API and CLI for pK_a calculations with a valuable trade-off between fast and accurate predictions. PypKa allows pK_a calculations in existing protocols with the addition of a few extra lines of code. PypKa supports CPU parallel computing on solvated proteins obtained from the PDB repository, but also from MD simulations using three common naming schemes: GROMOS, AMBER, and CHARMM. The code and documentation to this open-source project is publicly available at https://github.com/mms-fcul/PypKa

2.1.2 Introduction

Over the years several structure-based protein pK_a predicting tools have been developed. One of the most popular class of pK_a estimation methods relies on solving the Poisson–Boltzmann (PB) equation.[209, 283, 284] In this model the protein is described as a low dielectric medium containing fixed charges immersed in a high dielectric solvent. The protein dielectric constant (ε_{prot}) is an empirical parameter that represents the contributions that are not explicitly taken into account in the model such as electronic polarization. These methods usually rely on the calculation of the pK_{half} value on a single conformation, and assume this is a good estimate for the macroscopic pK_a. In fact, this approximation is reasonable when the protein structure is representative of both protonation states. When this condition is not met, a bias towards the preferred state of the structure can be propagated onto the predicted pK_a value.

Several methods have been developed to take advantage of the PB calculation coupled with Monte Carlo (MC) sampling of the protonation states. Some approaches modulate the ε_{prot} depending on amino acid types [285, 286], or the different regions of the protein [164, 194], and, more recently, the use of a smooth Gaussian function to capture the heterogeneity between protein and the water phase [185]. The underlying rationale is that increasing the dielectric response accounts for rearrangements due to the local electric field and increase the screening of individual interactions between the titrating groups. To improve their overall accuracy, other methods include explicit protein motions, like side-chain flexibility [187], hydrogen bond orientation and tautomers [170, 287].

More computational expensive alternatives that average pK_{half} values over multiple conformations have also been proposed. One way to perform conformational averaging is to sample protein side chain rotamers according to the Boltzmann distribution at a given pH and use the probability of each conformer to estimate the pK_a values. [171, 288] Alternatively, with the linear response approximation (LRA) method it is possible to estimate the pK_a value of a site by averaging over the pK_a values of the protonated and deprotonated states obtained from regular MD sampling. LRA has been successfully used to estimate pK_a values in peptides and proteins [175, 289, 290], however, it used on single sites and is only useful while there is a significant overlap between the conformational ensembles of both states. The so-called constant-pH molecular dynamics were developed to fulfill the need to capture the coupling between the protonation and conformational spaces[99, 177, 178, 215, 219, 228, 230, 237, 240, 241, 243, 244, 246–248, 291–300]. By leveraging the complementarity between molecular dynamics and continuum electrostatics, CpHMD methods have been used to study, among other properties, the titration curves of proteins [293, 301] at a higher computational cost.

The number of methodologies that nowadays rely on PB/MC calculations is remarkable and clearly highlights the need for fast and reliable PB solvers that can be easily incorporated in their workflows. Here, we propose a new wrapper tool, called PypKa, that brings together the

PB solver DelPhi v.5[164] and MC calculations[170] to predict pK_a values of titratable sites in proteins. PypKa was developed to be a highly scalable and flexible PB-based pK_a predictor with proton tautomers that is easy to integrate into a pipeline and to be used as a Python application programming interface (API) or command-line interface (CLI). Also, it can be easily extended to perform LRA calculations, while a CpHMD implementation is in development. The current version of PypKa can predict pK_a values of a protein with 40 sites in about 2 minutes on a single Intel Xeon E5-2620 type processor. PypKa is written in Python and Cython. Source code is freely available at https://github.com/mms-fcul/PypKa under the LGPL-3.0 license. The package can be installed from PyPi (https://pypi.org/project/pypka/). Documentation is available on Read the Docs (https://pypka.rtfd.io/).

2.1.3 Methods

Basic Usage

PypKa provides a simple Python API for the calculation of pK_a values of titratable sites from a PDB file (See Code 1). The core of PypKa is the Titration class which runs the calculations upon instantiation and the returned object can then be used to retrieve the results (Supplementary Figure A.1 and Code A.1). Currently, PypKa supports both the PDB and GROMACS (.gro) input formats and one can submit structures compliant with the naming scheme of most popular atomistic force-fields (AMBER, CHARMM & GROMOS) or directly from the Protein Data Bank. All canonical amino acids and also some lipids are supported (DMPC, POPC, POPE and cholesterol). In the default configuration, PypKa will calculate the pK_a values of all titratable sites in the PDB file. However, the user can set the site list, the number of parallel processes, and change most parameters of the calculation, such as the grid size and resolution or the convergence criterion. A complete list of all available parameters can be found in the online documentation. A Jupyter notebook with a case study and basic usage is also available at the GitHub repo.

The DelPhi4py module and the pK_a calculations

PypKa is a Python package which calculates pK_a values of titratable residues in biomolecules using a thermodynamic cycle approach and the PB model[209]. The PB equation is numerically solved by DelPhi [164] which has been integrated as a standalone Python wrapper (DelPhi4py). DelPhi4Py itself is distributed under a LGPL-3.0, while DelPhi is a proprietary software developed at Honig's Lab that is free for academic and research purposes.

PypKa takes full advantage of this python wrapper by keeping all the data structures in memory and minimizing the number of created files. Without DelPhi4py, PypKa would create $3 \times n$ input files for DelPhi while DelPhi would output $6 \times n$ files, where n is the number of states

```
>>> from pypka import Titration
>>>
>>> # Set user defined parameters
>>> params = {
>>>     'structure'      : '4lzt.pdb',
>>>     'pH'             : "0,14",
>>>     'ncpus'          : 1,
>>>     'epsin'          : 15,
>>>     'ionicstr'       : 0.1,
>>>     'pbc_dimensions': 0,
>>>     'convergence'    : 0.1
>>> }
>>> sites = {
>>>     'A': ['1N', '1', '7', '129C']
>>> }
>>> # Run calculation on a list of titratable sites
>>> tit = Titration(params, sites=sites)
Start Preprocessing
Start PB Calculations
PB Runs Ended
MC Runs Ended
Results
Chain  Site   Name      pK
   A      1    NTR    7.0
   A      1    LYS    10.45
   A      7    GLU    3.59
   A    129    CTR    2.6
API exited successfully
```

Listing 1: Usage example of running a PypKa simulation on the python API. For information on the parameters and information on the use of the CLI please refer to the online documentation.

(tautomers+1) of all titratable sites in the protein. For each state, a PB calculation is performed on the model compound while a calculation using a two-step focusing procedure is applied on the protein (for example, a Glu residue has 1 charged state + 4 neutral tautomeric states – the syn/anti conformations in each oxygen relative to the opposing carbonyl – which means that 15 calculations are performed). All operations on the atomic coordinates, charges and radii are performed in PypKa and passed to Delphi4py as pointers, while parameters and DelPhi outputs are interchanged by value, removing the need to perform a huge amount of I/O operations.

PypKa uses an established framework for estimating pK_a values which couples PB calculations to MC sampling of protonation states[170, 198, 209, 210, 302]. This approach requires solving a thermodynamic cycle containing model compounds (typically fragments of amino acid residues) whose pK_a values in solution (pK_{mod}) can be calibrated[293].

Although the PB-derived energies could be used to retrieve the distribution of probabilities for each ionization state, the analytical solution can only be calculated for a small number of sites. For most proteins, with many titratable sites, it is necessary to recover it numerically. PypKa includes a high performance MC algorithm written in Cython to sample protonation states with explicit inclusion of tautomerism, which is based on PETIT software [170].

There is an ever increasing number of experimental pK_a values that can be used to test and refine pK_a predictor methods. In fact, a database containing $\sim$ 1500 pK_a values has been recently published [95]. However, this opportunity to improve our computational approaches can not be fully seized if one is not able to automate them over hundreds of proteins. The proteins used in this work were taken from the PKAD database[95] (Supplementary Table A.1). Besides experimental pK_a values, this database includes experimental conditions and relative solvent accessible surface areas (SASA), defined as the percentage of SASA in the protein compared to a free residue in water. For more details on structure preparation and benchmark settings, please see the Methods section of Supplementary Information (A).

2.1.4 Benchmark

We used PypKa with different settings to calculate pK_a values for a dataset with 521 residues[95] and several specific structures of HEWL, which is a common test case for pK_a predictors. We evaluated the speed performance of PypKa on the 4LZT structure of HEWL. This calculation comprised 38 titratable residues (including SER and THR), resulting in 158 tautomers and took $\sim$130 s to calculate in one core using an Intel(R) Xeon(R) E5-2620 v4 @ 2.10GHz CPU (Supplementary Figure A.2). This process could be parallelized by increasing the number of CPU cores with almost linear scaling, due to the fact that PB calculations on each site and MC calculations at a given pH value were treated as independent embarrassingly parallel tasks.

The dielectric constant of the protein (ε_{prot}) is a key parameter in most PB-based pK_a calculation methods and often needs to be adjusted. Even though a dielectric constant is a well-defined property in physics, in the context of continuum models and especially when used for pK_a calculations, this is no longer the case. Here, the ε_{prot} is increased from 2, which accounts for electronic polarization, by a number of factors, including compensations for the lack of structural reorganization in rigid protein structures.[283, 303] Consequently, as it is done in many approaches, we optimize the ε_{prot} over the larger dataset (Supplementary Figure A.3A). We performed this ε_{prot} scan on our 521 residues dataset and observed that values higher than 8 are required to outperform the Null model (0.95), which assumes that the pK_a values in the protein remain the same as in water. Please note that the solution pK_a values used were the ones measured with Ala-based pentapeptides in Nick Pace's Lab,[19] which already include some effects from the main-chain. Higher ε_{prot} values attenuate the role of the electrostatic interactions, resulting in pK_a predictions closer to the Null model. Generally, the less solvated residues are harder to predict and require a combination of a lower ε_{prot} and a favorable conformation of the amino acid side chain and its environment. Since this requirements are often not met, the RMSE values obtained for less solvated residues are usually higher (Supplementary Figure A.3B). The analysis showed that ε_{prot}=15 is the one that minimizes the RMSE (0.82 in full set and 1.24 in the sub set) and it will be adopted for the remaining calculations in this work. The RMSE value obtained on such a large and heterogeneous data set is encouraging, especially since we observed a lower mean absolute error (MAE: 0.57) and standard deviation (STD: 0.78) compared to the Null model (MAE: 0.63; STD: 0.95).

We propose that the less solvated residues in the large data set are a real challenge for our method and probably this holds true also for other methods relying on fixed structures. To illustrate this effect, we calculated the RMSE values grouped by different SASA values (Supplementary Figure A.4). It is clear that we only obtained RMSE values significantly lower than the Null model in a few bins with low SASA. This can be explained mainly by two factors: (1) the Null model RMSE values are already quite low, especially for the more solvent exposed sites; and (2) the less exposed residues are difficult to predict since not all have a structure with a balanced electrostatic environment, which allows the correct pK_a prediction. In fact, the less solvent exposed residues are contributing the most to the observed MAE improvement (19% for SAS $<$ 30% vs. 8% for SAS $>$ 30%), compared to the Null Model. In summary, if the challenge is too easy (more solvent exposed groups), the Null model also performs well, if the group is more buried, the difficulty increases, which requires the use of a lower ε_{prot} and a protein crystal structure that captures both the protonated and deprotonated electrostatic environments. This is probably the main limitation of performing pK_a calculations on rigid experimental structures, since many of them are biased to the solution pH used in the experiment and/or to the force field used in the electron density fitting procedure. Many authors devised strategies to circumvent this problem, by modulating the ε_{prot} [164, 185, 194, 285, 286] or by including explicit protein motions [170, 171, 187, 287, 288]. However, in many cases, these approaches only

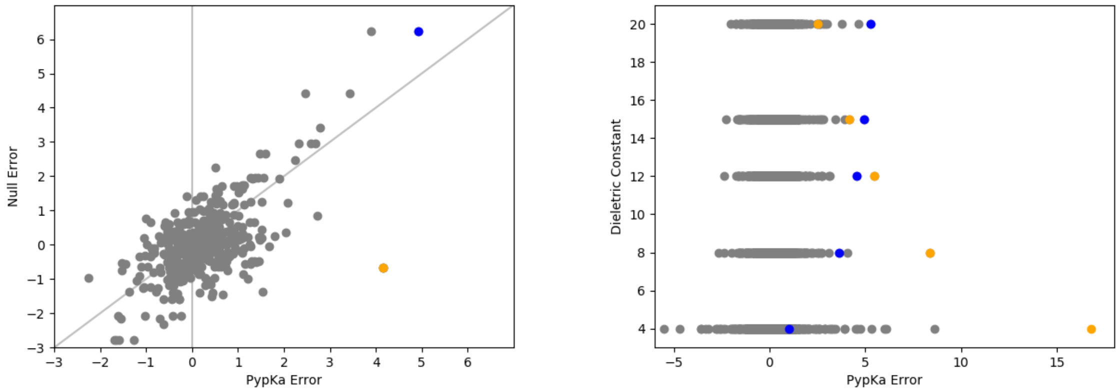


Figure 2.1: Scatter plots of PypKa errors vs Null model errors (A) and Dielectric constant of the protein (ε_{prot}) (B). The plotted errors were obtained by subtracting the predicted to the experimental values. The data shown in subplot A was obtained with ε_{prot}=15. Two data points (blue: Asp26_1TRW and orange: His46_1STN) were selected to illustrate the role of ε_{prot} and the representativeness of the experimental structure.

attenuate the fundamental problem in which there is no clear single conformation representing both protonation states.

To illustrate the shortcomings of performing pK_a calculations on rigid structures, we have plotted the PypKa errors vs. the Null model errors (Figure 2.1A). The selected cases (blue and orange data points) illustrate two different cases where our method struggled. In one case (Asp26_1TRW, blue point), lowering the ε_{prot} improves significantly the prediction, while in the other case (His46_1STN, orange point), the opposite is observed (Figure 2.1B). In the crystal structure (PDBID:1TRW), we observe that Asp26 is significantly buried and interacting strongly with a neighboring serine (Supplementary Figure A.5A). This environment leads to a strong desolvation effect, which increases the experimental pK_a value of this group by several units. An increase in ε_{prot} attenuates this desolvation contribution, hindering the methods' predictive ability. Looking at the structure of His46_1STN (Supplementary Figure A.5B), we observe that despite the presence of neighboring carboxylic acids, the strong interaction with the main chain N-H groups of Lys48 and Lys49 is overwhelmingly stabilizing the neutral form. This results in artificially low pK_a values, especially at lower ε_{prot}. However, it is easily envisaged a scenario where this His residue can have both protonation states stabilized by neighboring groups. Alternatively to the neutral form, when protonated, a small conformational change towards the two carboxylic groups (Glu43 and Glu52) will lead to a significant stabilization. The proper weighting of these two states should result in more balanced pK_a values and closer to the Null model.

We performed a detailed analysis of RMSE and MAE values per residue type from our full data set (Supplementary Table A.2). Our PypKa method improves the pK_a predictions for almost

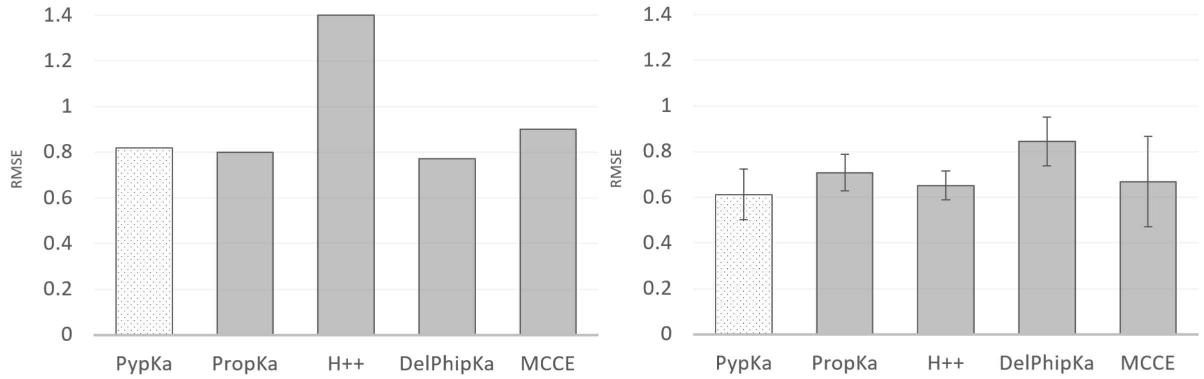


Figure 2.2: pK_a RMSE value comparison between PypKa and other predictors (PropKA[149], H++[169], DelPhiPKA[304], and MCCE[213]). The RMSE values were obtained from the original publications and performed on different data sets (A) or calculated in this work using a HEWL-based data set (2LZT, 3LZT, 4LZT, 2VB1, and 6RT3) (B).

all commonly titratable residue types, with the exception of glutamic acids. In this data set, the Glu and Lys residues present remarkably low Null model MAE values, which make them harder to improve. In overall, our method predicts pK_a values with very low MAEs (<=0.63) and a strong performance with Asp and Tyr residue types. The apparent different performance between Asp and Glu residues in our method is not really noticeable (the obtained MAE values are very similar) and is probably related with the specific difficulty of this data set and not that much with our method.

The comparison of the pK_a prediction ability between the different methods (PropKA[149], H++, DelPhiPKA[304], and MCCE[213]) is not trivial. Most authors benchmark their tools with specific data sets, which differ significantly both in size and difficulty (Figure 2.2A). Furthermore, not all authors use the same set of experimental values to calculate their pK_a shifts and there is some variability among the values obtained for the same residues. This can arise from different techniques, [138] measurement conditions, and experimental errors due to pH-induced protein denaturation.[305] It seems that the better performing methods obtained RMSE values around 0.8 pK units, a commonly observed value in many pK_a calculations [170, 171, 187, 210, 293, 301, 306–308]. The H++ tool showed a higher RMSE value, which can be explained by a more difficult data set containing many sites (33%) with a large pK_a shift[169]. To normalize this comparison, we need to define a data set that can be used with all pK_a predictors and maintaining similar settings. An application to our large data set would be hard to automatize and computationally expensive. Furthermore, it would require several calibration and optimization steps to allow for a fair comparison between methods. Therefore, we used a simple test set comprised of several HEWL crystal structures with similar settings and compared all methods performances (Figure 2.2B). Interestingly, in such a controlled benchmark, it seems that all pK_a predictors perform similarly. Furthermore, the most relevant conclusion from the data is that no method seems to lower the RMSE below an apparent limiting value. This RMSE limit is data set specific (∼0.8 in the large data set and ∼0.6 in the HEWL data set) and its magnitude is correlated with their overall difficulty. For these methods, which rely on a single structure, the difficulty to predict pK_a values is related to the structure representativeness. In this work, we showed two amino acid sites (Supplementary Figure A.5) illustrating this problem where a change in ε_{prot} has an opposite effect on the estimation. Therefore, in data sets containing similar cases, all pK_a predictors, which rely on single conformations and continuum electrostatic models, are expected to struggle to overcome the above-mentioned RMSE lower limit.

In summary, we showed several examples where our method can successfully estimate the pK_a values from rigid-body pK_half calculations on single crystal structures. On the other hand, there are also many cases where we could not accurately predict the experimental pK_a values from those structures. Their lack of representativeness requires the addition of conformational sampling to the PypKa calculations. Nevertheless, our tool is an excellent option to streamline pK_a calculations and to be incorporated into complementary methodologies which account for conformational sampling.

2.1.5 Conclusions

We introduce a new tool to calculate pK_half values on protein structures. PypKa is an object-oriented API which can easily be used, extended and modified, providing both a high-level protocol set-up capability and a reliable approach to estimate pK_a values. The rapid development is supported by a preprocessing module, validated charges, radii, and PB parameter defaults. Since the settings can be adjusted, the calculations can be optimized either for speed or accuracy, depending on the user's needs. Since most tasks are trivially parallelizable, PypKa shows a high scalability and allows for the calculation of the whole large data set in ∼2 hours per available CPU core.

We have benchmarked PypKa using a large data set of experimental values to show that the method has a competitive performance, compared with other approaches that also rely on pK_half calculations. The main advantage of using this application is its speed, simplicity, and robustness in the calculation of pK_half values. Interestingly, these features are key for the implementation of a new stochastic titration CpHMD method aiming at high computational speed and a user-friendly interface. Some of them have already been explored in this work, namely: multiprocessing, structure pre-processing, proton tautomers and multiple chains treatment. Additionally, the code also supports other features that will be useful in future applications: inclusion of DNA bases, explicit ions, and lipid patches with periodic boundary conditions in the PB calculations. PypKa also supports different naming schemes (compatible with GROMOS, CHARMM, and AMBER force fields) and a simplified procedure to add extra blocks for new molecules. This tool is also being developed in the context of a new Constant-pH MD implementation of the stochastic titration method. [178, 215] The high computational speed, scalability, and simplicity, makes PypKa an attractive solution to be extended and integrated into more complex schemes and to target large data sets of proteins. Due to its open-source nature, it provides the opportunity to constantly upgrade a flexible API that has applicability across a wide range of fields. The code is publicly available at https://github.com/mms-fcul/PypKa

2.1.6 Acknowledgements

We acknowledge financial support from FCT through grants SFRH/BD/136226/2018 and CEECIND/02300/2017 and projects PTDC/BIA-BFS/28419/2017, UIDB/04046/2020, and UIDP/04046/2020.

2.2 PypKa-MD: future-proofing the stochastic titration constant-pH molecular dynamics method

Pedro B.P.S. Reis*, Filipe E. P. Rodrigues, Telmo G. D. Silva, João G. N. Sequeira, Miguel Machuqueiro*

2.2.1 Context

After the development of PypKa (Chapter 2.1), the next milestone was to obtain a new CpHMD implementation that solved the usability and fragmentation issues plaguing previous versions. The new implementation needed to have a simple and flexible architecture that could harbor future extensions. As initially planned, with PypKa, we had a tool that ran Poisson–Boltzmann and Monte Carlo calculations, a significant part of the stochastic titration CpHMD workflow. Thus, the majority of the complexity of the new CpHMD implementation was dealt with, which greatly accelerated the development. We have also overhauled the philosophy of the CpHMD titratable residue blocks to make them more intuitive and to streamline the creation of new blocks. We leveraged the simplified process to include the support to the CHARMM36m force field, which we validated in another work [6]. Although PypKa-MD is available and can be used, this work has not yet been published as we want to validate it on a more extensive set of proteins.

2.2.2 Abstract

Constant-pH molecular dynamics (CpHMD) simulations are an effective tool for studying pH-dependent phenomena. By coupling conformational and protonation sampling, CpHMD methods confer more realism over traditional MD. However, mostly due to usability issues and the computational overhead, CpHMD is still not a go-to solution for most research groups. Here, we present PypKa-MD, an easy-to-use implementation of the stochastic titration CpHMD method with a simplified modular architecture to accommodate future developments. PypKa-MD supports GROMOS and CHARMM force fields, as well as modern versions of GROMACS. The new implementation was validated on alanine-based tetrapeptides with closely interacting titratable residues and four commonly used benchmark proteins. The pK_a values predicted by PypKa-MD are highly similar and correlated to the ones of a previously validated implementation. Both versions also showed a comparable performance at reproducing experimental values. PypKa-MD's code can be inspected and downloaded at https://github.com/mms-fcul/PypKa-MD.

2.2.3 Introduction

The structure, stability, and function of proteins are usually pH-dependent. However, these pH effects are often ignored in molecular dynamics (MD) simulations due to the difficulty of sampling correct protonation states. Over the last 28 years, many constant-pH MD (CpHMD) methods have been developed to address these limitations [4, 99, 176–178, 215, 218–220, 226–228, 230, 232, 234, 235, 240, 241, 243, 244, 246–248, 291–300, 309–319]. The different strategies employed can be distinguished mainly by: (i) the type of protonation (continuous vs. discrete); (ii) the force field (AMBER, CHARMM, GROMOS, OPLS, MARTINI, etc.), and its level of detail (all-atom, united-atom, and coarse grain); and (iii) the approximations used to deal with charge fluctuations in the simulation box, often related with the use of counterions and the long-range electrostatics treatment (reaction-field vs. Ewald summation methods). With only a few exceptions [176, 293], most methods are implemented in only one software package and for a single force field, hindering performance comparisons between methods and/or force fields. Nonetheless, there have been attempts to perform comparative studies of pK_a predictors, such as the pK_a Cooperative [179, 180], that included several CpHMD methods. The blind predictions highlighted many problems with the available methods at the time. Similar initiatives should be promoted by the community to assess the evolution of the field in recent years.

Originally developed by Baptista *et al.* [178], the stochastic titration method is a seminal discrete protonation CpHMD. In this methodology, the MD simulation is periodically updated with protonation states sampled with Monte Carlo (MC) from Poisson–Boltzmann-derived energies. The stochastic titration CpHMD has been extensively validated in a number of systems including well-solvated proteins [292, 301, 320], peptides [215, 321–323], membrane proteins [99, 324–326], membranes [240, 241], and dendrimers [236, 237]. Despite its demonstrated accuracy, CpHMD is not yet preferred over traditional MD for most studies. We believe the lack of adoption by the community is related to two issues: usability and computational cost. Like the one maintained by António Baptista, the implementation developed by our group (L-CpHMD) only supported the GROMOS force field family, most notably, the 54A7 [327, 328]. There are ongoing efforts to validate CHARMM [6] and AMBER extensions, which would greatly amplify the force field choice for users. However, the current implementations still face a fragmentation problem, as there is no single version in which all features are available. Furthermore, to run an L-CpHMD simulation in its current version, a steep learning curve needs to be overcome. Multiple avenues have been pursued to decrease the computational cost. In Baptista's implementation, which uses MEAD [166] for the Poisson-Boltzmann (PB) calculations, a reduced titration scheme was introduced. By keeping the protonation state of residues with a very low likelihood of changing its state (i.e., less than 0.1%) fixed for a few cycles, it is possible to avoid a significant amount of PB runs. We have opted to use the PB solver Delphi [329] in L-CpHMD, allowing for a less stringent, and thus less computationally demanding, convergence criterion while maintaining an equivalent accuracy [330]. We have also validated the usage of GROMOS with particle mesh Ewald (PME) [5], instead of reaction field (RF) with which it was parameterized, to be able to leverage GROMACS's [331, 332] GPU acceleration for MD simulations (unfortunately, GROMACS GPU code does not support the charge groups needed for RF).

The stochastic titration CpHMD method derives all parameters required for the PB calculations from the underlying force field. However, the treatment of model compounds retains some of the theoretical vagueness present in PB methods concerning their molecular definition and pK_a values [293]. Instead of being a molecule featuring the same chemical group with a known experimental pK_a value, the model compound of a site is a non-physical fragment defined as a portion of the amino acid residue (usually the complete side chain). The pK_a value of a model compound (pK_{mod}) is then calibrated using experimental data of simple systems [293]. For the amino acid side chains, we have used the pK_a values measured from NMR data on alanine-based pentapeptides (AAXAA, where X is a titratable residue) [19, 94]. By performing a calibration, it is possible to offset systematic errors introduced by the PB parameters. Unfortunately, a new calibration procedure is required whenever important PB-related parameters are changed.

One of the most critical stages of the development cycle of a new computational method is its validation. In this work, we are not introducing a new CpHMD method but rather a new implementation of the stochastic titration CpHMD method, which, as previously mentioned, has already been abundantly validated. Nevertheless, it is still indispensable to validate it, and to compare it against a previous implementation, L-CpHMD, to determine whether some of the changes introduced undesired effects. To test how the new implementation performed on closely interacting residues, we have used alanine-based tetrapeptides with two central adjacent glutamate and histidine residues. With just three small peptides (AEEA, AHEA, AHHA), it is possible to test strong interactions between anionic and cationic sites.

The systems used to benchmark new developments in the CpHMD field often include the same proteins [4, 176, 177, 219, 231, 232, 246, 291, 293, 295, 301, 333]. The hen egg-white lysozyme (HEWL) is arguably the most widely used test system for pK_a predictors [4, 177, 194, 219, 231, 232, 246, 291, 295, 301, 333] due to the large number of residues with available experimental data [138, 334, 335], many of which with highly shifted pK_a values (mainly in the acidic range). Another important protein is the *Staphylococcus aureus* nuclease (SNase), which is very stable and has a significant amount of experimental data [162, 336] with several unexpectedly hard-to-predict residues. Unfortunately, both of these proteins lack titratable cysteine residues. In order to add this important residue to our validation, we have also included two thioredoxin proteins, the human form (^{h}Trx) and another from *Escherichia coli* (^{Ec}Trx), which has two reduced cysteine residues.

In this work, we present our new implementation of the stochastic titration CpHMD method – PypKa-MD – that consolidates several features from different development forks. A significant effort was made to simplify its usage so that anyone that knows how to use GROMACS can easily set up and run CpHMD simulations. In the new Python-based implementation, PB and MC simulations are abstracted by PypKa's API [1] which significantly simplifies its architecture. PypKa-MD is meant to serve as the base upon which future developments will be built. Here, we performed the pK_{mod} calibration and preliminary validation of the new implementation on tetrapeptides and a few commonly used proteins.

2.2.4 Methods

CpHMD settings

MD simulations were interrupted at regular time intervals ($\tau = 20$ ps) and new protonation states were obtained from Monte Carlo (MC) calculations using Poisson-Boltzmann (PB) derived free-energy terms [4, 178, 215]. After the topology update, and prior to the next production MD segment, a very short (0.2 ps) solvent relaxation step (with frozen solute) is performed [178, 215]. All CpHMD simulations were performed using either the G^{54A7} force field for 25 ns. Three replicates were used to simulate the pH values ranging from 1–12 with a step of 1.0. All MD simulations were performed with an integration step of 2 fs using GROMACS 5.1.5 [331]. The SPC water model was used. The non-bonded interactions were treated with a single cutoff of 1.4 nm, updated every 5 steps in G^{54A7} simulations. Beyond the cutoff, all van der Waals interactions were truncated, and the Coulombic were treated with the generalized reaction field method [337] with a single cutoff of 1.4 nm.

A minimization procedure was applied to all systems. The steepest descent minimization algorithm was used with no constraints in the first step, while in the second the p-LINCS [338] and SETTLE [339] were turned on for solute and water molecules, respectively. Each replicate was initialized for 50 ps in NVT with an integration step of 1 fs, followed by another 50 ps in NPT with an integration step of 2 fs. In the NVT ensemble, the v-rescale thermostat [340] was used to keep the temperature at 310 K (coupling constant of 0.05 ps). In the NPT ensemble, the v-rescale thermostat [340] (coupling constant of 0.1 ps) was used in combination with the Parrinello-Rahman barostat [341] (coupling constant of 0.5 ps and an isothermal compressibility of 4.5×10^{-5} bar^{-1}).

PB calculations were performed using DelPhi v5.0 [329] using a two focusing procedure [342]. The dielectric constants were 80 and 2 for solvent and protein, respectively [178, 215]. During the focusing procedure, a grid space of 0.1 nm was used for the larger grid, which was reduced to 0.025 nm in the smaller grid. Both grids contain 81 grid nodes on each side and are centered in the titratable group. The convergence threshold was set to 0.01 k_bT/e [330]. The MC runs were performed for 10^5 cycles. In each cycle, random attempts are made to change the protonation state of every titratable group and of all pairs of sites with an interaction larger than 2 pK units.

Table 2.1: Proteins used as test systems in this study, their PDB codes, number of residues, and number of water molecules in the simulated system

System Name	Protein name	PDB ID	# AAs	# waters
HEWL	*G. gallus* Lysozyme	4LZT	129	6.4k
SNase	*S. aureus* Nuclease	1STN	149	8.5k
^{h}Trx	*H. sapiens* Thioredoxin	1TRW	105	3.8k
^{Ec}Trx	*E. coli* Thioredoxin	2TRX	108	3.9k

System setup

Four protein systems were prepared to test the C^{36m} implementation (Table 2.1): Lysozyme (HEWL) [343], *Staphylococcus aureus* nuclease (SNase) [344], human thioredoxin (^{h}Trx) [345] and *E. coli* thioredoxin (^{Ec}Trx) [346]. The experimental structure of SNase (PDB: 1STN) is missing 5 and 8 residues in the N- and C- terminus, respectively, which were completed. In the ^{Ec}Trx structure (PDB: 2TRX) the repeated chain was removed. All systems were solvated in a rhombic dodecahedral box with periodic boundary conditions.

2.2.5 Results and discussion

Table 2.2: Examples of features in different stochastic titration CpHMD implementations. $\text{CpHMD}_{\text{ITQB}}$ stands for the implementation developed and maintained at Baptista's group.

	$\text{CpHMD}_{\text{ITQB}}$	L-CpHMD	PypKa-MD
Poisson–Boltzmann	MEAD[166]	DelPhi[164]	PypKa (DelPhi)
Monte Carlo	PETIT[170]	PETIT[170]	PypKa[1]
GROMACS versions	4.0.7	4.0.7, >5.1.4 >2020	>5.1.4 >2020
Force fields	GROMOS	GROMOS CHARMM	GROMOS CHARMM
pdb2gmx-independent topology update	Yes	No	Yes
Centering procedure	fixbox	trjconv	fixbox
Reduced titration	Yes	No	Yes
Necessary input files	.mdp, .gro, .top .pHmdp, .sites .pdb, .mgm, .ogm	.mdp, .gro, .top .pHmdp, .sites .pdb, .pbp	.mdp, .gro, .top

Implementation

PypKa-MD is a novel implementation of the stochastic titration CpHMD (st-CpHMD) method originally developed by Baptista *et al.* [178]. In this method, the protonation state of the protein in a molecular dynamics (MD) simulation is periodically updated. The cyclic workflow of st-CpHMD is illustrated in Figure 2.3 and can be divided into four stages:

Poisson–Boltzmann A series of PB calculations using a single conformation are run to determine pH-independent energies like the intrinsic pK_a (pK_{intr}) and the interaction between sites.

Monte Carlo A MC simulation is performed to sample protonation states from the pH-independent energies. The final state of this simulation is selected.

Solvent Relaxation A short MD simulation is run in which the protein position is kept fixed with freeze groups, allowing the solvent to adapt to the new charge configuration of the protein.

Production MD A new segment of production MD is produced. The final conformation is used in the next PB step, starting a new CpHMD cycle.

The output of a CpHMD simulation is a trajectory of concatenated production MD segments and the protonation state of the titrating molecule over time.

PypKa-MD is the result of an effort to consolidate the best features of separate implementations at Baptista's and Machuqueiro's labs (Table 2.2). Arguably one of the most distinctive features of the new version is the abstraction of the PB and MC steps, which is handled by PypKa [1]. This design decision massively simplifies the complexity of our code. Furthermore, some desired characteristics of PypKa are automatically inherited, such as the usage of the PB solver DelPhi [164] and its parallelizable MC routine. Compared to the previous PB/MC procedure based on I/O intensive file creation and manipulation, PypKa communicates with the PB solver and the MC routine through memory pointers, increasing both computational efficiency and the numerical precision of the results. Adding extra residues to PypKa is also much easier thanks to a helper script that derives radii and charges from force field residue blocks and bonded

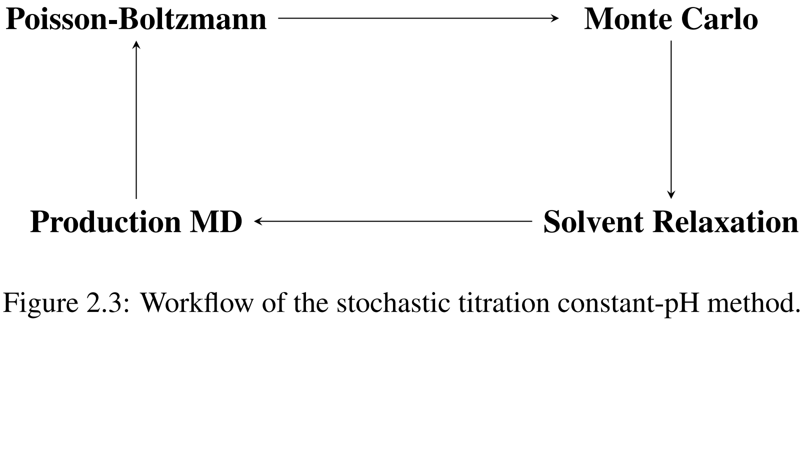


Figure 2.3: Workflow of the stochastic titration constant-pH method.

parameters files (.rtp and ffbonded).

The philosophy of the CpHMD titratable residue blocks was also simplified in PypKa-MD in order to streamline the creation of new titratable blocks in the force field. Previously, each tautomer had its own uniquely named residue block in the .rtp file. Currently, for each residue, there are only two residue blocks representing the protonated and deprotonated states, and the name of these blocks is the same as in the original force field. For example, the 5 tautomeric blocks (GL0, GL1, GL2, GL3, and GL4) representing the glutamate residue have been replaced by only two (GLU and GLUH). The difference between a regular GLU residue and a CpHMD GLU is the inclusion of four tautomeric hydrogens. Like in the older versions, in the deprotonated block all tautomeric hydrogens have no charge, and in the protonated residue, only one of the hydrogens is present (charged). Despite the fewer residue blocks, PypKa-MD supports all the same tautomers as before. The rules to convert between tautomers are stated in a new CpHMD-specific file. The format of this file has been inspired by a recently developed version at Baptista's group that does not make use of GROMACS's pdb2gmx to update the topology after PB/MC. Although implemented differently, this feature has been imported to PypKa-MD as well. In fact, topology manipulation is one of the most essential operations performed by PypKa-MD.

Until recently, all available implementations only supported the GROMOS G54A7 force field. While GROMOS is used by many groups, it is not one of the most popular force fields, which limits the adoption of the stochastic titration CpHMD method. Furthermore, the GROMOS force field parameters have been parametrized with the reaction field method using charge groups and a twin-range cutoff scheme, and from version 2016 onward, GROMACS discontinued the support for charge groups. Luckily, GROMOS is also compatible with an atomistic single cutoff (verlet), and PME [5]. Nevertheless, we have recently validated the L-CpHMD implementation on the CHARMM36m force field [6]. Moreover, in the last years, some L-CpHMD forks were adapted to take advantage of the accelerations provided by recent GROMACS versions, including the ability to run MD in GPUs. As the basis of all future developments, PypKa-MD already supports GROMOS 54A7 and CHARMM36m, as well as all modern GROMACS releases. In the near future, we plan on adding support to AMBER.

A key goal of the development of PypKa-MD was to make CpHMD significantly easier to use. Ideally, it should be as easy to use as traditional MD. Our usability concerns lead us to drastically reduce the number of CpHMD-specific input files. Actually, using PypKa-MD requires no additional files compared to running MD in GROMACS. It is now a matter of choosing a CpHMD-compatible force field during the system setup with pdb2gmx and then running the PypKa-MD executable with a modified .mdp file (Listing 2). The required .mdp file is still a valid input for GROMACS as the assignment of each CpHMD parameter is a GROMACS comment. If the comments start with two semicolons, they are also ignored by PypKa-MD. It

```
;;; CpHMD Parameters ;;;

; GROin = ../initial/init2.gro     ; input structure file
; TOPin = ../../box_min/topol.top  ; input topology file
; sysname = protein_x              ; output file name prefix

; ffID = CHARMM36mpH  ; CHARMM36mpH or G54A7pH
; sites = all         ; titrate all sites or specify
                      ; i.e. 1N 12 33 36C

; nCycles = 50    ; number of CpHMD cycles
                  ;   50 * 20ps (tau_prot) = 1ns
; nCPUs = 8       ; number of CPU cores to use
; GPUID = False   ; GPU slot to be used in MD
                  ; i.e 0,1,2,3 for GPU slot, False for no GPU

; GroDIR="/opt/gromacs-2021.5/bin/" ; GROMACS binary folder

; pH = 7.2            ; pH value of the simulation
; ionicstr = 0.1      ; ionic strength (M)

;;; OPTIONAL ;;;

;; DATin = protein_x.dat     ; fixbox input file
;; NDXin = protein_x.ndx     ; input index file
;; titrating_group = Protein ; titrating index name

;; pypka_ffID = CHARMM36mpH ; set of PB parameters for PypKa
;; pypka_nlit = 100         ; number of PB linear iterations

; reduced_titration = True  ; reduced titration state
; rt_cycles = 10            ; reduced titration cycle duration

;;; SLURM SETTINGS ;;;

;SLURM segments = 50  ; number of consecutive CpHMD runs
;SLURM partition = MD24, MD16 ; slurm partitions argument
;SLURM requeue = 1            ; slurm requeue argument
;SLURM pythonEXEC = python3.8 ; python executable with PypKa-MD
```

Listing 2: Example of PypKa-MD and Slurm parameters in a .mdp file.

is also possible to provide directives to Slurm, a widely used queue manager, by marking the line with the special comment "*;SLURM*". Fixbox is a very robust centering tool developed by António Baptista used to preprocess the MD output structures before the PB calculations. This program requires an input file with instructions about the order of assembly of existing chains. For protein systems, the automatically generated file is sufficient to achieve proper centering. Unfortunately, in more complex systems such as micelles, it is necessary to provide those instructions explicitly in a file defined in the *DATin* parameter. The reduced titration schema can also be activated and tuned in the same .mdp file. If turned on (*reduced_titration*), the protonation state of all titratable sites specified in *sites* will only be sampled with PB/MC every *rt_cycles* CpHMD cycles. This schema can dramatically increase the computational efficiency of a CpHMD by keeping fixed all titratable sites whose probability of changing their current protonation state is found to be less than *rt_limit* (typically 0.001).

Before running PypKa-MD, one is required to install it. The code can be inspected and downloaded from its GitHub repo https://github.com/mms-fcul/PypKa-MD. Alternatively, it can be installed from PyPI (the Python Package Index) with Pip. There are also two programs and respective dependencies that need to be installed in the system: PypKa and GROMACS.

pK_{mod} Calibration

A model compound contains the same chemical group as a protonatable site in the protein whose pK_{a} value is presumably known [293]. In our case, the model compounds are fragments of titratable amino acid residues, and their pK_{a} values (pK_{mod}) require calibration. We have calibrated these pK_{mod} values for L-CpHMD and PypKa-MD by employing a previously reported protocol [99, 293] that uses experimental data of Alanine-based pentapeptides ($\mathrm{Ala_2}$-X-$\mathrm{Ala_2}$, where X is a pH titrating residue – i.e. Asp, Cys, Glu, His, Lys or Tyr) [19, 94]. These peptides are often used to study the effect of the protein environment on the protonation behavior of titratable residues [99, 293]. All N- and C-termini were capped with an acetyl or amino group, respectively, except for when these termini were titrating. In these cases, an Ala pentapeptide was used. The calibration procedure consisted in running CpHMD simulations (3× 50 ns) at pH values near an initial pK_{mod} guess. After capturing the complete pH titration curve of the pentapeptide, a pK_{a} shift is obtained, which can be compared to the experimental pK_{a} values to generate the final pK_{mod} (Table 2.3). The final values show some significant shifts that can be attributed to the model compound definition. In fact, the most sensitive sites are the termini, which are defined as quite simple chemical groups connected to the peptide main chain.

We expected minor differences in the pK_{mod} values of the two versions since, as detailed in 2.2.5, there are several differences between the two. Nevertheless, dramatic differences in pK_{mod} values would likely mean an error in PypKa-MD. We observed an RMSD of 0.03 between L-CpHMD and PypKa-MD, a similar RMSD between the pK_{mod} values of the two

Table 2.3: pK^{mod} of all titratable residues for the Null model, L-CpHMD, and PypKa-MD. The Null model values were obtained from experimental pK_{a} measured with the AAXAA pentapeptides, where X is a titratable amino acid [19, 94].

Residue	Null/Exp.	L-CpHMD	PypKa-MD
C-ter	3.67	3.64	3.65
ASP	3.94	3.94	3.98
GLU	4.25	4.21	4.24
HIS	6.54	6.49	6.48
N-ter	8.00	7.91	7.97
CYS	8.55	8.60	8.61
TYR	9.84	9.83	9.87
LYS	10.4	10.37	10.36

implementations and the experimental pK_{a} of the residues in water (0.04 for PypKa-MD; 0.05 for L-CpHMD).

Validation

The stochastic titration CpHMD method, and, in particular, the L-CpHMD implementation have been extensively validated over the years [99, 215, 236, 237, 240, 241, 292, 301, 320–326]. Here, we validate our new implementation, PypKa-MD, by comparing it against the L-CpHMD. All MD simulations in both implementations were run on the same GROMACS version with the same parameters. Thus, any discrepancies observed must be explained either by the differences in the code and consequent choice of protonation states or by random variations in replicates due to lack of sampling. The latter divergences could be diminished by running long simulations and by increasing the number of replicates. Nevertheless, any severe errors introduced by the new implementation should be able to be easily spotted in the analysis performed of the selected systems.

The pK_{mod} calibration ensures that all highly solvated titratable residues exhibit their pK_{a} value in water. In this process, it is possible to offset systematic errors of an implementation. However, the calibration does not consider interactions between titratable residues. Alanine-based AXXA tetrapeptides in which the two X central residues are glutamate or histidine residues are a good system to test these close interactions. With only three peptides, AEEA, AHEA, and AHHA, one can study strong interactions between all combinations of anionic and cationic residues. As expected, the pK_{a} values obtained from L-CpHMD and PypKa-MD simulations (Table 2.4) are very similar (RMSD: 0.07; MAE: 0.06; R^2: 1.00). Despite some observable differences in the pK_{a} values, these are always inferior to $\sim$ 0.1 pH units. Furthermore, for all

Table 2.4: pK_a values and corresponding 95% confidence interval of capped tetrapeptides AXXA – where X is a glutamate or histidine residue – calculated from simulations of L-CpHMD and PypKa-MD.

System	Residue	L-CpHMD	PypKa-MD	Diff
AEEA	Glu-2	4.33 ± 0.12	4.26 ± 0.04	0.06
	Glu-3	4.35 ± 0.12	4.29 ± 0.04	0.05
AHEA	His-2	6.70 ± 0.07	6.61 ± 0.07	0.09
	Glu-3	4.08 ± 0.06	4.05 ± 0.08	0.03
AHHA	His-2	6.30 ± 0.15	6.42 ± 0.05	0.11
	His-3	6.48 ± 0.08	6.51 ± 0.14	0.03

residues, their 95% confidence interval overlap.

We have selected four proteins commonly used to benchmark pK_a predictors to further validate PypKa-MD. Here, the reduced titration scheme (RT) was active to accelerate the simulations. With RT, new protonation states are not computed for residues in a very likely protonation state (greater than 99.9%). Thus, the speedup provided by RT is dependent on the pH and protein. For example, in SNase the average number of titrating residues fluctuates between 1.6 and 22.2 in the pH range of our simulations (Supplementary Table B.1). This translates into a speedup of the PB/MC stage ranging from 2.7× to 12.5× (using 8 Intel Xeon CPU E5-2620 v4 @ 2.10GHz). The acceleration provided by RT to PB/MC is quite impressive, considering that a similar speedup (13×) is given to MD by moving the simulations from CPU to GPU [6].

Before calculating an equilibrium property such as a pK_a value, it is necessary to check the stability of the systems at all pH values. To this effect, we have calculated for all replicates and pH values the Cα RMSD (Supplementary Figure B.1), and secondary structure profiles (Supplementary Figure B.2). Even at extreme pH values, most properties converge within 5-15 ns. Regardless, to better compare the protonation state preferences of each CpHMD implementation which would naturally drift as the sampled conformational space diverged, we have chosen not to discard the non-equilibrated section and to use all simulation time to compute the pK_a values for all HEWL, SNase, ^{h}Trx, and ^{Ec}Trx shown in Supplementary Table B.2, Table B, Table B.4, and Table B.5 respectively.

Overall the pK_a values obtained with L-CpHMD are in agreement with those of PypKa-MD for all proteins tested (Table 2.5). In addition to the low discrepancy (MAD: 0.20; RMSD: 0.30), the predicted pK_a values by both versions also display a very high correlation (R^2: 0.99). Moreover, the standard deviation between the replicate averages of the two implementations (0.23) is inferior to the average standard deviation between the replicates of PypKa-MD (0.32) and L-CpHMD (0.30). The consistently positive, albeit low, mean deviation value indicates that, compared to L-CpHMD, PypKa-MD tends to underestimate pK_a values. This systematic

Table 2.5: Coefficient of determination (R^2), mean deviation (MD), mean absolute deviation (MAD), and root mean squared deviation (RMSD) between L-CpHMD and PypKa-MD pK_a predictions for each system.

System	# Residues	R^2	MD	MAD	RMSD
HEWL	21	1.00	0.05	0.12	0.17
SNase	52	0.99	0.05	0.20	0.28
^{h}Trx	33	0.98	0.07	0.24	0.49
^{Ec}Trx	33	0.99	0.05	0.18	0.25
All	139	0.99	0.05	0.20	0.30

Table 2.6: Mean absolute error (MAE) and root mean squared error (RMSE) of L-CpHMD and PypKa-MD experimental pK_a predictions for each system.

System	# Residues	L-CpHMD		PypKa-MD	
		MAE	RMSE	MAE	RMSE
HEWL	17	0.65	0.91	0.63	0.90
SNase	20	0.97	1.15	0.98	1.20
^{h}Trx	17	0.60	0.76	0.72	0.98
^{Ec}Trx	6	1.07	1.39	1.02	1.37
All	60	0.79	1.03	0.81	1.17
All w/o outliers*	57	0.77	1.00	0.73	0.95

* Excluding: ^{h}Trx Asp-26, and Asp-58; SNase His-46.

deviation warrants further investigation and could be explained by the truncation of an energy contribution, such as the background interactions in L-CpHMD. The analysis of individual residue types (Supplementary Table B.6) is less robust due to the scarce amount of examples of some types and higher sensitivity to outliers. Although it is still possible to observe the overall trend of good correlation and similarity, upon a closer inspection, a few residues that exhibit large fluctuations between replicates can be identified as outliers (SNase His-46 and ^{h}Trx Asp-26 and Asp-58). As previously stated, it is likely that for these more difficult to converge cases, it would be necessary to have more replicates and longer simulations. Finally, we compare the performance of the CpHMD implementation at reproducing experimental values (Table 2.6). As expected, PypKa-MD and L-CpHMD display a closely matched accuracy, with one of the implementations outperforming the other in half of the systems. The performance gap observed when analyzing all residues dissipates when one removes the previously identified outliers.

2.2.6 Conclusions

In this work, we have introduced PypKa-MD, a novel implementation of the stochastic titration CpHMD method, inspired by the best features of all the current forks, which will serve as the basis for future development. Due to the use of PypKa to handle the PB and MC stages, PypKa-MD's architecture is markedly simplified. Besides the future-proofing of CpHMD, this implementation focused on two key aspects holding back the adoption of previous versions: ease of use and speed. Running a CpHMD simulation with PypKa-MD is now essentially equivalent to running traditional MD with GROMACS, requiring only the use of the GROMOS or CHARMM CpHMD-compatible force field during system setup and a modified .mdp file with CpHMD-specific parameters. In the future, we will extend the support to the AMBER force field and provide an automated system neutralization procedure to add the correct number of counterions to minimize system charge at different pH values as required by PME. The compatibility with current and upcoming versions of GROMACS ensures that any future accelerations in MD simulations offered by the GROMACS suite can be taken advantage of in CpHMD. The reduced titration scheme implemented in PypKa can speed up large systems by more than an order of magnitude. Further performance improvements are expected from replacing all PB calculations with pKAI, a machine learning model that predicts pK_a values up to 1000× faster than PypKa [3]. Moreover, enhanced sampled methods such as replica exchange and umbrella sampling implemented in L-CpHMD will also be ported to PypKa-MD.

After performing the calibration of the model compounds' pK_a value with alanine-based pentapeptides, we have validated PypKa-MD by comparing its accuracy against L-CpHMD, a previous implementation of the method. First, three tetrapeptides were used to verify the comparable performance of PypKa-MD in a small system of closely interacting titratable residues. Then, we selected four commonly used proteins for benchmarking pK_a predictors as a preliminary validation. In this set of systems, PypKa-MD was found to output extremely similar and correlated pK_a values to those of L-CpHMD. Furthermore, the results confirmed that both implementations display a comparable accuracy at predicting experimental pK_a values. A more complete benchmark with more proteins will be executed soon, as well as an optimization of the charges and radii used in the PB calculations.

2.2.7 Acknowledgements

We thank António Baptista, Sara S. Campos, and Paulo Martel for fruitful discussions. We acknowledge financial support from Fundação para a Ciência e a Tecnologia through grants SFRH/BD/136226/2018, 2021.05909.BD, and CEECIND/02300/2017, and projects UIDB/04046/2020 and UIDP/04046/2020.

Chapter 3

Accelerating pK_a calculations with Artificial Intelligence

3.1 pKAI: A fast and interpretable deep learning approach for accurate electrostatics-driven pK_a predictions

Pedro B.P.S. Reis*, Marco Bertolini, Floriane Montanari, Walter Rocchia, Miguel Machuqueiro*, and Djork-Arné Clevert*

3.1.1 Context

At the start of this PhD there were no machine learning models to predict pK_a values in proteins, nor did we plan to train one. However, as artificial intelligence became an integral presence in more scientific fields, we noticed the opportunity to obtain pK_a estimations with an associated error similar to a PB-based method at the computational cost of a faster empirical predictor. We did not train any model directly on experimental data for several reasons: the sheer number of examples is limited and it has not grown in the last years; the examples are very skewed towards well-solvated and with small pK_a shifts; macroscopic pK_a predictions could not be used to accelerate CpHMD. Therefore, we set out to create a model that mimics PypKa (Chapter 2.1) and outputs pK_a estimations of a single conformation. To do so, we had first to generate a database with examples (Chapter 4.1). This work was published in the Journal of Chemical Theory and Computation (DOI: 10.1021/acs.jctc.2c00308).

Abstract

Existing computational methods to estimate pK_a values in proteins rely on theoretical approximations and lengthy computations. In this work, we use a data set of 6 million theoretically determined pK_a shifts to train deep learning models that are shown to rival the physics-based predictors. These neural networks managed to infer the electrostatic contribution of different chemical groups, and learned the importance of solvent exposure and close interactions, including hydrogen bonds. Although trained only using theoretical data, our pKAI+ model displays the best accuracy on a test set of $\sim$750 experimental values. Inference times allow speedups of more than 1000 times faster than physics-based methods. By combining speed, accuracy and a reasonable understanding of the underlying physics, our models provide a game-changing solution for fast estimations of macroscopic pK_a from ensembles of microscopic values as well as for many downstream applications such as molecular docking and constant-pH molecular dynamics simulations.

3.1.2 Introduction

Many biological processes are triggered by changes in the ionization state of key amino acid side-chains [283, 347]. Experimentally, the titration behavior of a molecule can be measured using potentiometry or by tracking free energy changes across a pH range. For individual sites, titration curves can be derived from infrared or NMR spectroscopy. Detailed microscopic information can be quickly and inexpensively obtained with computational methods, and several *in silico* pK_a calculations have become widely used to provide insights about structural and functional properties of proteins [149, 169, 171].

In Poisson–Boltzmann-based (PB) methods, the solvent is implicitly described while proteins are represented by point charges in a low dielectric medium [1, 169, 171, 304]. These continuum electrostatics (CE) methods assume that the pK_a^{single} (the proton binding affinity for a chemical group in a given conformation, often called pK_{half} in theoretical calculations) is a good estimate for the macroscopic pK_a value. This assumption holds when the protein structure is sufficiently representative of the conformational ensembles corresponding to both protonation states. Experimentally determined structures exhibit conformations at a minimum energy state, which, in turn, is related to a specific protonation state. However, biomolecular systems can explore different energy basins, which may exhibit alternative protonation states. Energy minima can be affected by experimental conditions, such as temperature, ionic strength, and pH. Inaccuracies in pK_a predictions due to limited conformational rearrangements can be reduced by increasing the protein dielectric constant from its default value (2-4), which only accounts for electronic polarisation. The dielectric constant can be used as an empirical parameter mimicking the effect of the response mechanisms to the local electric field that is not explicitly taken into account in the model [164, 181, 185, 194, 285]. A more computationally expensive approach is to explicitly include protein motion by sampling conformers via Monte Carlo (MC) or molecular dynamics (MD) simulations and applying conformational averaging [170, 171, 187, 287]. Finally, by coupling the sampling of protonation states at given pH and conformations, constant-pH MD methods [4, 177, 178, 231, 295] provide greater insight into pH-dependent processes [99, 237, 348–350].

As larger data sets of experimental pK_a values have become available, a new class of purely empirical methods has been developed. These models rely on statistical fits of empirical parameters weighting the different energetic contributions into simplified functions. PROPKA [149] is arguably the most popular of such methods [351] and has been shown to perform competitively even when compared to higher-level theory methods [1, 352]. The empirical methods are much faster than the physics-based ones, although at the cost of providing less microscopic insights, and their predictive power is unknown on mutations and/or proteins dissimilar to those composing the training set.

The accuracy of most predictors is bound to the estimation of the same quantity, the so-called

$\Delta\text{p}K_\text{a}$. This is the free energy of transferring the ionizable residue from the solvent to the protein compared to its neutral counterpart. Since $\text{p}K_\text{a}$ values for all amino acids in water have been experimentally determined, the $\text{p}K_a^{solvent}$ term can be fixed, and, in practice, it can also be adjusted to incorporate systematic errors. The $\Delta\text{p}K_\text{a}$ can be regarded as a sum of mostly electrostatic contributions stemming from the residue microenvironment. Therefore, an accurate prediction of $\text{p}K_\text{a}$ values for a given conformation requires a correct description of the residue interactions with the surrounding protein charges and with the solvent.

At their core, deep learning (DL) models are complex non-linear empirical functions fitted to best map input variables to output properties. Considering chemical properties, such as $\text{p}K_\text{a}$ values, which are dictated by molecular configurations, and provided that enough examples are presented, it is possible to train a model to map this relationship without the need to solve non-linear equations in 3D or to sort through the massive space of possible states.

In this paper, we have developed two DL-based $\text{p}K_\text{a}$ predictors: pKAI and pKAI+, for $\text{p}K_\text{a}^\text{single}$ and experimental $\text{p}K_\text{a}$ values, respectively. These models have been trained on a database with $\sim$6 million $\text{p}K_\text{a}$ values estimated from $\sim$50 thousand structures using a continuum electrostatics method, PypKa [1]. pKAI+ displays an unrivaled performance at predicting experimental $\text{p}K_\text{a}$ values on a $\sim$750 members data set. Also, pKAI exhibits an accuracy comparable to the PB-based predictor used to generate the training set while being approximately 10–1000× faster. By applying explainable artificial intelligence (XAI) analysis, we show that these simple models are able to implicitly model most of the required energetic contributions, such as Coulomb interactions, desolvation, and hydrogen bonding. Therefore, the presented models feature the best characteristics of CE-based methods – accuracy and interpretability – with the speed provided by empirical approaches.

3.1.3 Methods

Data set

To train our DL models, we used a large publicly available data set of estimated pK values – the pKPDB database [2]. This data set of $\sim$6M $\text{p}K_\text{a}$ values was created by running the PypKa tool with default parameters [1] over all the protein structures deposited on the Protein Data Bank. The PB solver DelPhi [164] was used with a dielectric constant equal to 15 and 0.1M of ionic strength. A two-step focusing procedure was employed with a coarser grid spacing of 1Å and the subsequent calculation using a finer grid with 0.25Å between nodes. Monte Carlo sampling was used to sample protonation microstates and tautomers.

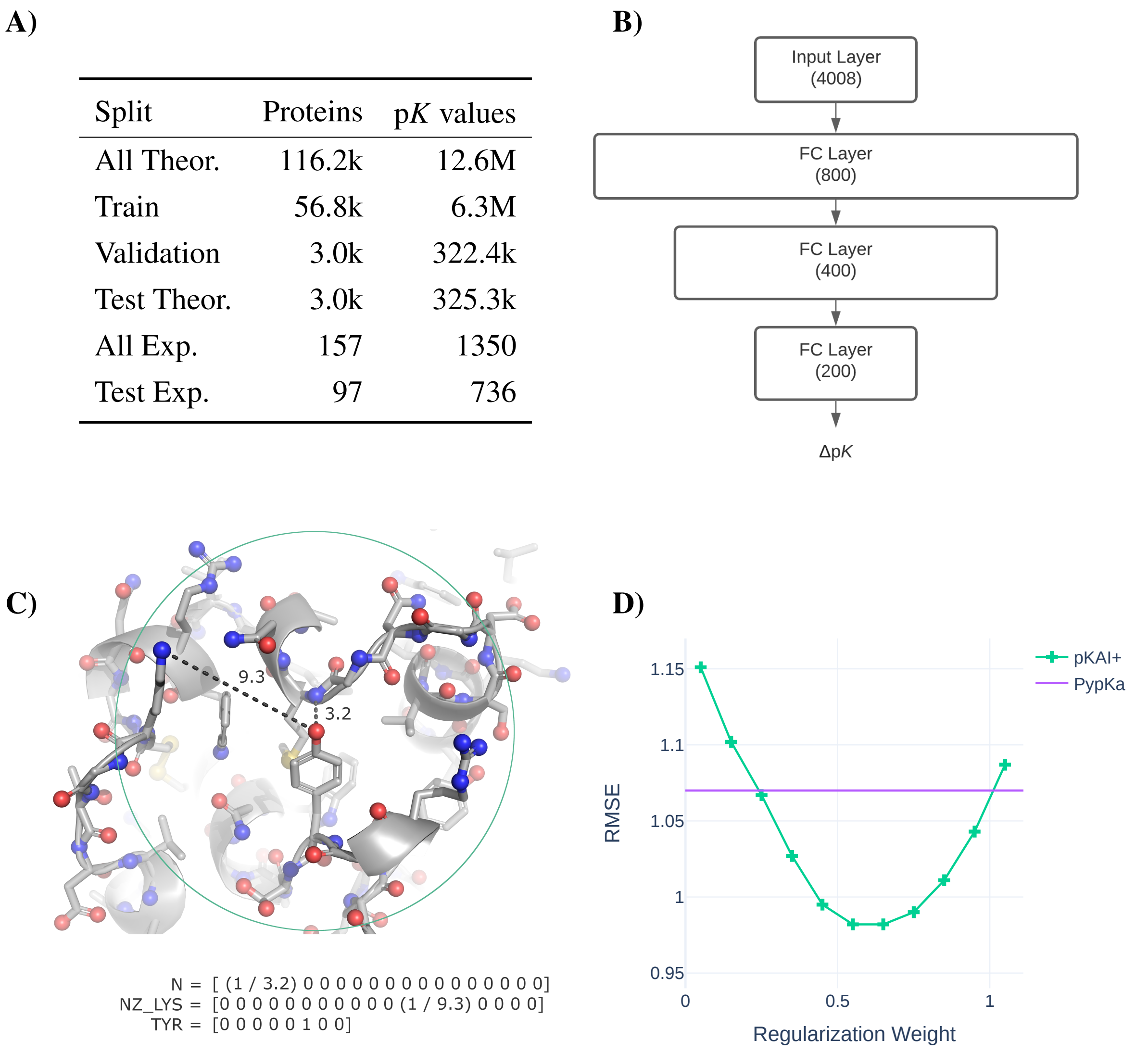
The target values to be fitted by our model are theoretical $\text{p}K_\text{a}^\text{single}$ values estimated with a PB-based method. This implies that pKAI will inherit the assumptions and limitations of this class of predictors. Our approach contrasts with the one usually adopted for training empirical predictors, which entails using experimental values to fit the model's parameters. The main advantage of this novel approach is that we can train models with significantly more parameters, such as deep learning ones since there is now a much larger abundance of training data. As a comparison, in PROPKA3 only 85 experimental values of aspartate and glutamate residues were used to fit 6 parameters [149]. Recently, traditional ML models have been trained on $\sim$1k experimental pK_a values [155, 156]. However, testing the real-world performance of such methods is difficult as there is a high degree of similarity among available experimental data. Our larger data set translates into more diversity in terms of protein and residue types and, more importantly, a wider variety of residue environments. It also helps our models to steer away from the undesired overfitting. Furthermore, the relation between a structure and our target property is deterministic, contrary to that of experimental pK_a values, which suffers from the lack of entropic information.

The ultimate goal of these methods is to accurately predict experimental pK_a values, and thus, we have assessed the model's performance with $\sim$750 experimental pK_a values taken from the largest compilation of experimentally determined pK_a values of protein residues reported in the literature – the PKAD database [95]. The 97 proteins in the experimental test set are reported in the Supplementary Table C.1. We compare our experimental results with a null model (attributing to each titratable group the corresponding pK_a value in water), PypKa (the method used to generate the training set), and PROPKA with default settings (the empirical method of reference).

Before training our models on our data set, we applied a curated data split (Table 3.1A) to ensure that the training, validation, and test sets did not contain proteins with a high degree of similarity and prevent overfitting. First, we randomly selected 3k proteins from the full data set of $\sim$120k proteins as our holdout test set of theoretical pK_a values. The program mmseqs [353] was then used to exclude all proteins containing at least one chain similar to any of the chains found either in the experimental or in the theoretical test sets. Chains were considered to be similar if they presented sequence identity over 90%. From the remaining set of proteins, 3.000 more were randomly assigned to the validation set, while the rest became the training set. Finally, we have excluded similar proteins to those of the validation set from the training set. In the experimental data set, we have excluded all duplicated proteins, non-exact pK_a values (e.g., $>$12.0), and residues for which PypKa or PROPKA failed to produce an estimate.

Model architecture and implementation

pKAI is implemented and trained using PyTorch v1.9.0 [354] and PyTorch Lightning v1.2.10 [355]. The model has a simple architecture comprised of 3 fully-connected hidden layers in a pyramidal configuration fitted to the pK_a shifts of titratable amino acids (Figure 3.1B). The simplicity of the architecture is intentional. pKAI is meant to serve as a proof-of-concept that



A)

Split	Proteins	pK values
All Theor.	116.2k	12.6M
Train	56.8k	6.3M
Validation	3.0k	322.4k
Test Theor.	3.0k	325.3k
All Exp.	157	1350
Test Exp.	97	736

Figure 3.1: A) Overview of the data split and similarity exclusion performed on the pKPDB, and PKAD databases [2, 95]. B) pKAI model architecture. C) Illustration of the titratable amino acid environment encoding. Only Nitrogen, Oxygen, and Sulfur atoms (shown as spheres) within a 15 Å cutoff (green circle) are included, while all carbon (shown as sticks) and hydrogens (omitted) are ignored. The included atoms are represented by the inverse of their distance to the titratable residue in an OHE vector featuring 16 categories of atom classes (Supplementary Table C.6). The titratable residue is represented by an OHE vector comprised of 8 classes. D) Performance of pKAI+ with different regularization weights on the experimental test set.

deep learning models can capture the effect of electrostatic interactions in the pK_a of titratable residues. Recent work has shown that it is possible to have an ML model that accurately predicts electrostatic solvation energies of proteins [356]. However, pK_a estimations are even more complex, requiring at least 2 PB calculations per residue state for the physics-based counterpart (e.g., in PypKa, each carboxylic acid has 5 states, hence 10 PB calculations are required for each Asp/Glu residue).

The encoding of the environment of each titratable residue has been simplified to retain only the most important electrostatic descriptors (Figure 3.1C). Considering the decay rate of the electrostatic potential, we decided to truncate the contributions to the environment of a residue by applying a cutoff of 15 Å around the labile atom(s) of the titratable residue. In practice, this cutoff is slightly smaller for some residue environments as the necessary input layer size normalization resulted in the truncation of the closest 250 atoms. It is expected that the bigger the protein, the higher the occurrence of residues with a cutoff less than 15 Å. Nevertheless, the truncation only excludes quite distant atoms, and 14.85 Å was the minimum cutoff value observed in the test set. A further approximation was made by considering only highly charged atoms as they perform the strongest electrostatic interactions with the titratable site and assuming that solvent exposure can be inferred from the titratable residues distances to nearby atoms (similar to half-sphere exposure [357]). This simplification can be slightly compensated by using atom classes instead of charges or element names, as they implicitly provide information about adjacent atoms. The atoms were one-hot encoded (OHE), and in order to reduce the input layer size, chemically similar atoms were assigned to the same category (Supplementary Table C.6). While carboxylic oxygen atoms (C-termini OXT; aspartates OD1 and OD2; glutamates OE1 and OE2) and primary amine atoms (arginines NH1 and NH2) atoms have been merged, others with similar names but different chemical properties were separated (glutamines OE1 and NE2 from glutamates OE1 and histidines NE2, asparagines OD1 from aspartates OD1; main chain N from N-termini N).

The final 4008-sized input layer consisted of 250 atoms represented by 16 OHE classes concatenated to an 8-dimension OHE vector corresponding to the titrating amino acid. Each atom's OHE was multiplied by its reciprocal distance to the titrating residues so that this valuable information could be included without increasing the size of the input layer.

pKAI is freely available as a python module that can be installed via pip. The source code can be found at https://github.com/bayer-science-for-a-better-life/pKAI.

Training

Training was performed with mini-batches of 256 examples, and the Adam optimizer [358] with a learning rate of $1e^{-6}$ and weight decay of $1e^{-4}$. Dropout regularization was applied to all fully-connected layers with the exception of the last one. Hyper-parameter optimization was performed with Optuna [359] using the performance in the validation set. Training these models takes approximately 10 minutes on an NVIDIA Tesla M40 24GB, using 16bit precision and an early stopping strategy on the minimization of the cost function with a delta of $1e^{-3}$ and patience of 5 steps.

The pKAI model was trained on an MSE cost function while for the pKAI+ we have added a regularization parameter α to penalize $\Delta\text{p}K_\text{a}$ predictions (y). Thus, the loss function of pKAI+ becomes

$$J(y_i, \hat{y}_i, \alpha) = (1-\alpha)(y_i - \hat{y}_i)^2 + \alpha \hat{y}_i^{\,2} \tag{3.1}$$

where y_i is the true value and $\hat{y}_i$ the estimation. Different regularization weights were tested to check for overfitting (Figure 3.1D). While we have selected an α of 50%, any value in the 40–70% range would lead to a similar improvement. Moreover, the same trend is observed when dividing the experimental test into 5 folds (Supplementary Figure C.1).

XAI Methods

For each input atom feature $\widehat{a} = (a, r_a)$, where a indicates the atom class and r_a the corresponding distance to the liable atom(s) of the titrating residue, we compute the corresponding attribution $I(\widehat{a})$ with the Integrated Gradients (IG) algorithm,[360] as implemented in the `shap` package [361]. $I(\widehat{a})$ measures the sensitivity of the network output with respect to changes in the input $\widehat{a}$. A large absolute value of $I(\widehat{a})$ indicates that the network assigns high importance to this feature while the sign of $I(\widehat{a})$ indicates whether the feature contributes positively or negatively to the output. Given that the most important contributions to the $\Delta\text{p}K_\text{a}$ are of electrostatic nature, one can try to explain the model inferred charges for each atom class a by computing the distant-independent score C:

$$C(a) = \mathbb{E}\left[r_a^{-1} I_-(\widehat{a})\right] - \mathbb{E}\left[r_a^{-1} I_+(\widehat{a})\right], \tag{3.2}$$

where I_- and I_+ are negative and positive I values, respectively. The C score of an atom class is thus the difference between the distance weighted average of examples with negative and positive I values over a large subset (10000 samples) of the test set. The sign of $C(a)$ in equation 3.2 resembles the charge that the network, on average, assigns to a given atom type. For example, if an atom class is being perceived by the model as contributing negatively to the $\Delta\text{p}K_\text{a}$ ($\mathbb{E}\left[r_a^{-1} I_-(\widehat{a})\right] > \mathbb{E}\left[r_a^{-1} I_+(\widehat{a})\right]$ hence $C(a) > 0$), this would mean that the network learned that this particular atom stabilizes the deprotonated state, which is characteristic of positively charged groups.

The solvent-accessible surface area (SASA) values shown in Supplementary Table C.2 and in the XAI subsection have been taken from pKPDB [2].

3.1.4 Results

The main goal of pKAI is to mimic the pK_a predictive ability of PB-based methods with a significant computational performance improvement. Our training set is comprised of pK_a values calculated using PypKa on a large number of proteins taken from the Protein Data Bank [2]. An elaborate data split was performed to minimize data leakage from the training set to the validation and test sets (see Methods). pKAI was designed to be a simple and interpretable model using the minimum structural features that still capture the electrostatic environment surrounding every titratable residue. The model has been trained on ΔpK_a values rather than on absolute values. The pK_a shift is, in fact, a more appropriate quantity to learn, less dependent on the chemical peculiarities of individual amino acids, and more sensitive to the local electrostatic environment. For example, residues that share a common side-chain chemical group (such as glutamate and aspartate sharing a carboxylic acid) are influenced by the same environment in a similar way.

We wanted our model to capture the electrostatic dependence between the environment of a residue and its consequent pK_a shift while keeping the input layer as small as possible (see Methods). By ignoring all carbon and hydrogen atoms, we are greatly reducing the dimensionality of our input layer while retaining most of the information regarding charged particles. There is, of course, a significant loss of topological information, although much can be inferred from the positions of the included atoms. In fact, there is no performance gain when adding solvent exposure measurements (e.g., SASA, residue depth) to the environment embedding. Considering that solvent exposure entails topological information and that the model is not able to extract additional information from it, we conclude that it was already estimating, to some degree, these molecular properties (see Model Explainability subsection).

pKAI: predicting theoretical pK_a

The performance of the model on the test set is reported in Supplementary Table C.2 and Figure 3.2A. The null model used for comparison consists of the reference pK_a value in water for each residue type and corresponds to 0 in the ΔpK_a scale. Overall, pKAI reproduces the PB-based ΔpK_a values with an MAE value of 0.31, an RMSE of 0.52 and an R^2 of 0.93. However, in this case, we are only predicting theoretical values with a well-defined relation between structure and pK_a^{single} (pK value of a single conformation). Experimental pK_a estimation is a much more complex task since the pK_a^{single} values corresponding to the different conformations spanned by the protein should be weighted according to their occurrence probability at equilibrium. The performance of pKAI is impressive considering the high complexity of the dependence between pK_a and the site electrostatic environment, illustrated by the high RMSE value of the Null model (1.89). Some residues are easier to predict (e.g., LYS and termini residues) while others are more challenging (e.g., CYS and TYR). This can be explained by

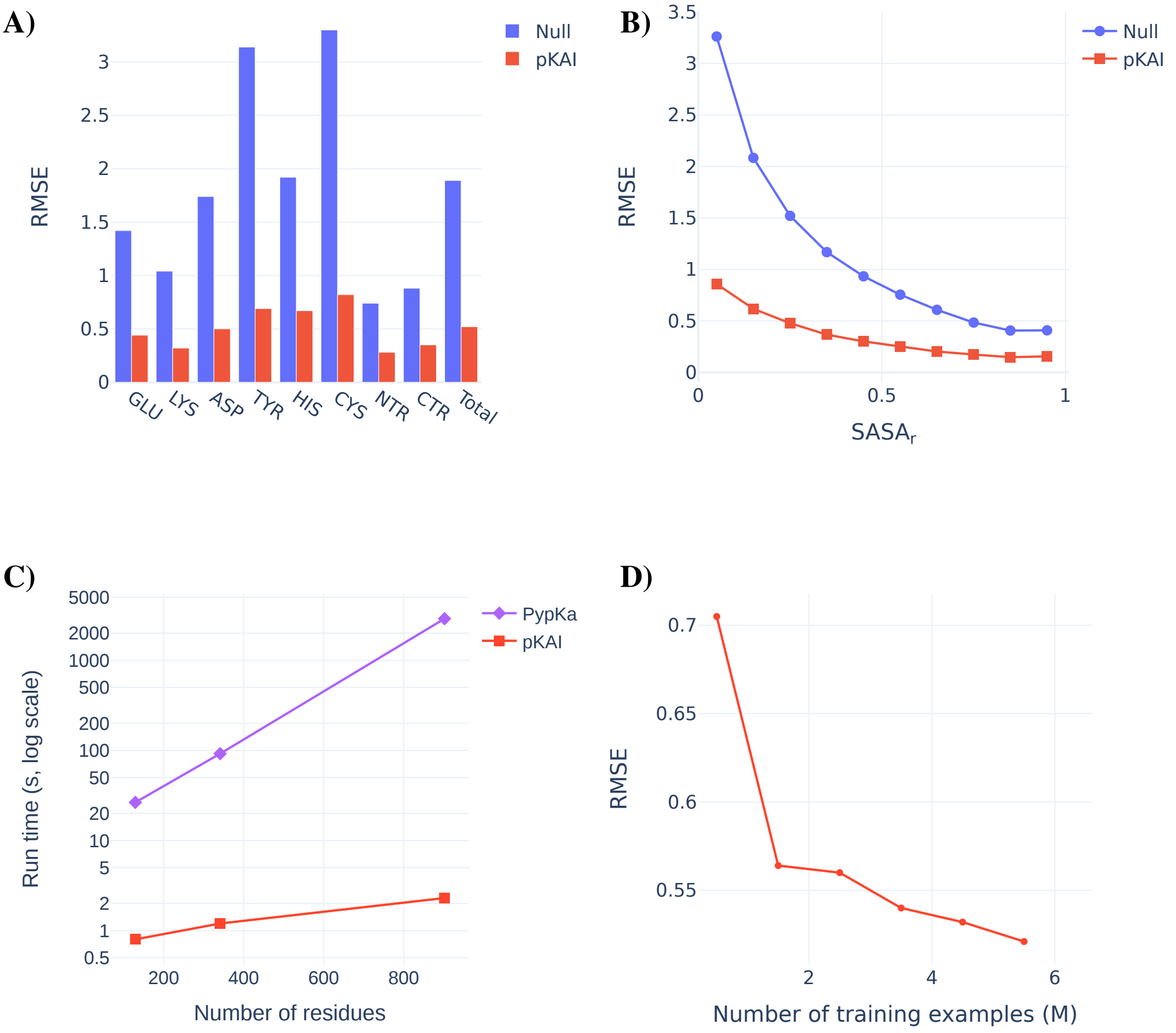


Figure 3.2: A) Comparison between Null model and pKAI RMSE values (values shown in Supplementary Table C.2). The Null model is defined as the pK_a values of the residues in water taken from Thurlkill *et al.* [19]. B) Performance at predicting pK_a^{single} values dependency on the magnitude of solvent exposure (SASA). The calculations were performed for pKAI and Null model using the PypKa predictions as reference. C) Execution time comparison between PypKa and pKAI (values shown in Supplementary Table C.3). This benchmark was executed on a machine with a single Intel Xeon E5-2620 processor. D) Effect of the size of the training set in the model performance on the validation set.

their solvent exposure distribution (Figure 3.2B): well-solvated residues exhibit small ΔpK_a values, while more buried ones are more affected by the desolvation effect and establish more interactions with other residues causing their pK_a values to shift. There is a clear dependency between the solvent exposure of a residue, its ΔpK_a value, and the prediction difficulty (Supplementary Figure C.2). The excellent performance of pKAI is also demonstrated by the fact that most predictions (81.2%) exhibit an error below 0.5 pK units, which is sufficient for most use cases.

The main advantage of DL models is the potential speedup they can provide. Since continuum electrostatics (CE) pK_a estimations need to sample thermodynamic equilibrium microstates, several iterative simulations have to be performed on each protonation state and on the environment of every residue. On the other hand, pKAI merely needs to apply its learned function over each residue and, as such, is remarkably faster (Figure 3.2C). Moreover, the convergence of the CE simulations is harder to achieve as the protein size increases. Consequently, in PypKa, as the protein size increases, so does the time required to estimate each pK_a value. In contrast, the run time of pKAI's DL model has a different dependence on the protein size. Since the bigger is the protein, the larger is the amount of calculations that can be performed simultaneously, then the less significant becomes the model loading cost and the faster the average per-residue execution time. This results in sublinear scaling performance and in a speedup over its CE counterpart that can exceed over a thousand times. As such, pKAI is a particularly valuable tool for dealing with very large systems with thousands of residues where the only added computational cost stems from the prepossessing of the structure.

Another important factor contributing to the high accuracy obtained is the considerable size of the training set. Despite using the largest repository of experimental protein structures and the largest pK_a database available [2], we show that there is still a correlation between the number of examples in the training set and the accuracy of the model (Figure 3.2D). This indicates that our model can still be improved by providing further examples of pK_a values.

pKAI+: Predicting experimental pK_a values

The main goal of pK_a predictors, such as PypKa, is to estimate the macroscopic pK_a value for the titratable residues using structures (usually experimental ones). Since pKAI aims at reproducing the pK_a^{single} calculated with PypKa at a fraction of the computational cost, it is not expected to outperform the PB-based method in predicting experimental values. When using PB to predict experimental pK_as, a higher dielectric constant for the solute is often adopted to compensate for the lack of conformational flexibility in the method and the lack of representativity of the experimental input structure. A similar approach can be implemented in pKAI by introducing a regularization weight to the cost function (pKAI+). This regularization penalizes the magnitude of the ΔpK_a prediction. In practice, this procedure biases our estimates towards

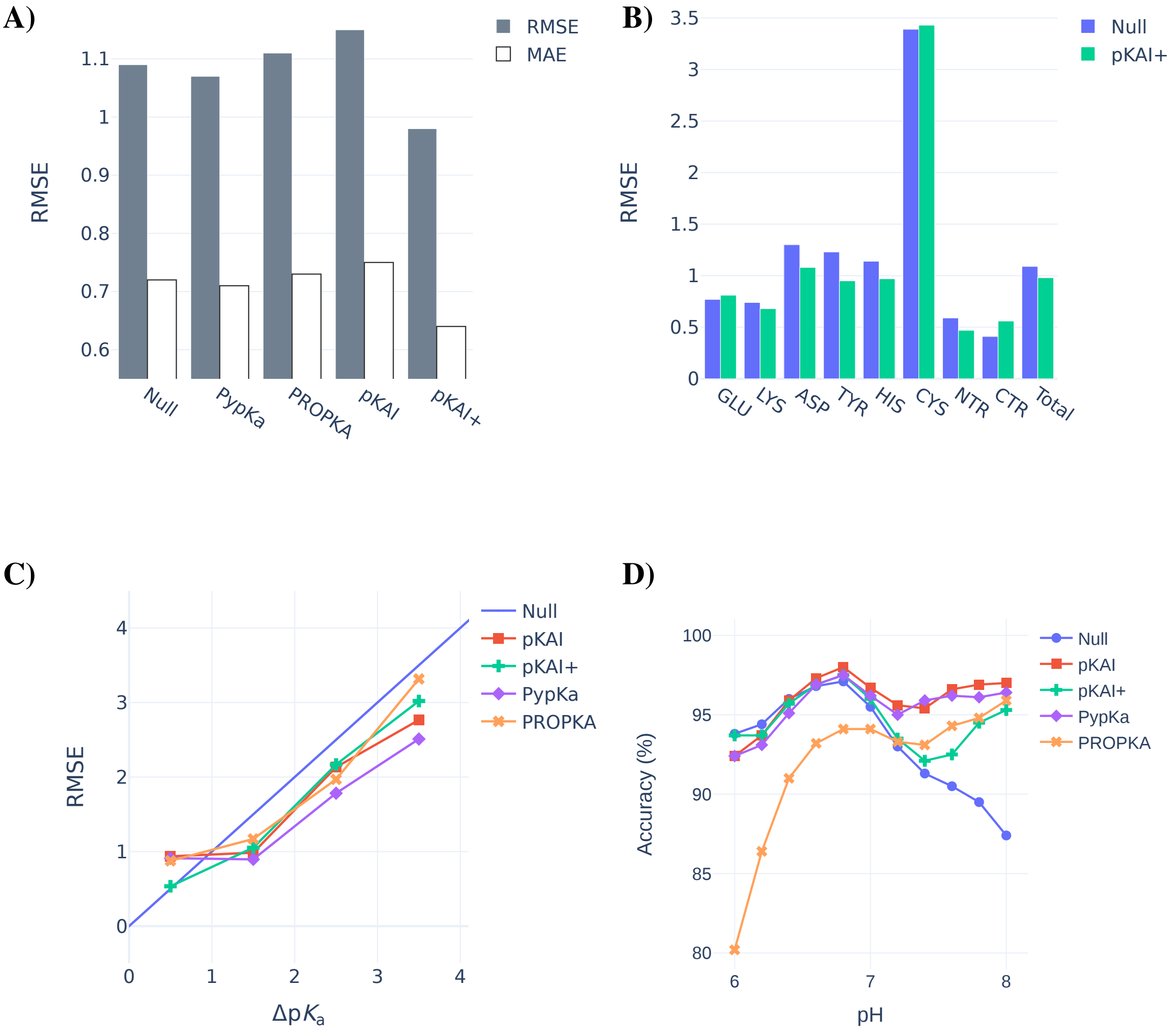


Figure 3.3: A) Experimental pK_a benchmark of several methods on a data set of 736 residues from 97 proteins (values shown in Supplementary Table C.5). The null model values are the pK_a values of each amino acid substituted in an alanine pentapeptide (Ace-AA-X-AA-NH_2) [19, 94]. B) Comparison between Null model and pKAI+ performance by residue type. C) Prediction errors of the different models given the experimental pK_a shift (ΔpK_a). D) Accuracy of several methods at predicting representative protonation states derived from experimental pK_a values. Residues at a pH within 1.5 units of the experimental pK_a are considered not to have a representative protonation state.

the pK_a values in water, similarly to what is done by the increased solute dielectric constant in PB-based approaches. However, the analogous effect is applied evenly to all residues, independently of the solvent exposure. Thus, adding the regularization penalty is different from training pKAI with a data set generated with a higher protein dielectric constant. Furthermore, we have previously benchmarked PypKa on a range of dielectric constants (4 – 20) and shown that there is no benefit in increasing the dielectric constant to values greater than 15 [1]. It should be noted that pKAI+ has not been trained on experimental pK_a, but rather on the same training set as pKAI.

To evaluate the performance of our model, we have benchmarked it using a data set of 736 titratable residues in 97 proteins with experimentally determined pK_a values (Figure 3.3A). Remarkably, pKAI+ (RMSE of 0.98) is able to outperform both PypKa (RMSE of 1.07) and PROPKA (state-of-the-art empirical pK_a predictor, RMSE of 1.11). Furthermore, the improvement over the other methods is significant for most residue types (Figure 3.3B) and can be quantified using metrics that are more (RMSE, 0.9 quantile) or less (MAE, error percentage under 0.5) sensitive to the presence of outliers (Supplementary Table C.4). Cysteine residues are particularly difficult to predict because they naturally occur less frequently and are more buried than all other titratable residues. This leads to an under-representation of these residues in the training set while exhibiting the largest pK_a shifts. To illustrate the difficulty of this data set, note that some methodologies are not able to improve on the null model (RMSE of 1.09). The reported deviations are specific to this data set. Even though our benchmark is one of the largest ever used to validate a pK_a predictor, it is likely still insufficient to quantify the true accuracy of these methods. Furthermore, besides being limited, these test sets used for validating new pK_a predictors tend to always be different. This makes it very hard to compare methods without rerunning them. In this benchmark, PypKa represents the PB-based methods like DelPhiPKa[304] or H++[169]. More computationally expensive methods such as MCCE[213] or constant-pH MD are not represented here. These methods are expected to outperform PB-based methods, which rely on a single structure, although the exact improvement on this test set is hard to predict. DeepKa is a recently published convolutional neural network trained on theoretical pK_a values from constant-pH MD (CpHMD) simulations [157]. As expected, the CpHMD implemented in the Amber suite[176] (RMSE of 1.02) outperformed PROPKA (RMSE of 1.12) in their test set, which only includes the 4 residues (Asp, Glu, His, and Lys) predicted by DeepKa (RMSE of 1.05).

The difficulty of estimating pK_a values is not the same for all residues. pK_a predictors are usually a valuable tool to predict residues in which the shift is significant. For example, if a residue is completely exposed to the solvent and performs no other interactions, its pK_a will be equal to its known value in water. To assess our model's performance while avoiding cherry-picking, no particular cases were analyzed. Instead, we have classified the residues according to their solvent exposure level (Supplementary Figure C.3) and the magnitude of the experimental pK_a

shifts. pKAI+ shows comparable RMSE values to PypKa for both the most solvent-exposed and buried residues. Interestingly, it is also able to surpass the PB-based model for partially exposed residues. Notably, pKAI+ only improves the PypKa predictions for pK_a shifts smaller than 1 pK unit (Figure 3.3C). This indicates that pKAI+ corrects the pK_a values of partially exposed residues which are establishing non-representative interactions in the experimental structure. Since there is a large number of residues with these characteristics in the test set [2], the overall performance improvement is significant (Supplementary Table C.5).

From the pK_a value of a residue, it is possible to derive its most likely protonation state at a given pH. To perform this conversion, one must assume that the Henderson–Hasselbalch (HH) equation can describe its protonation behavior, implying that no other titratable residues influence its titration. According to the HH equation, at a pH equal to the pK_a value, the protonated and deprotonated species exist in the same proportion. Hence, at this pH value, there is no most probable protonation state. At a pH value that is 1.0 unit away from the pK_a value, the least likely protonation state still occurs 30% of the time. To account for this fact and to alleviate the aforementioned approximation, when calculating the most representative protonation state of a residue from pH 0 to 12, at each pH value, only residues with an experimental pK_a at a minimum distance of 1.5 units were considered. The 1.5 pH cutoff is arbitrary, but the same trend was observed when slightly different values (0.5 – 2) were used. The most abundant protonation states obtained from pKAI predictions are in good agreement with those derived from experiments and outperform those of PROPKA in a wide range of pH values (Supplementary Figure C.4). Moreover, pKAI is the best model at assigning a fixed protonation state to a protein at biologically-relevant pH values (Figure 3.3D), arguably the most common task pK_a predictors are used for. In contrast to the poor performance of the Null model and PROPKA at the physiological pH range, both models outperform pKAI and PypKa in pH values inferior to 4.0. In the acidic region, most of the Glu and Asp residues, which make up for around 60% of the experimental test set, are titrating. PROPKA was trained on some of these Glu and Asp residues [149], which may have resulted in an over-optimistic evaluation of its performance at lower pH values. pKAI+ is biased to predict pK_a values between those of pKAI and the Null model. This bias has granted the model an edge on experimental pK_a estimations. However, in tasks in which the Null model does not perform well, pKAI+'s ability is also affected. This can be seen in the biological range at the more basic pH values.

Model Explainability

The main driving force for pK_a shifts in proteins is electrostatic in nature. In our model, each atom of the environment represents the contribution of a chemical group or part of a residue. This individual contribution towards the final ΔpK_a prediction can be estimated (see XAI in the Methods section for further details), and it is shown in Figure 3.4A. Remarkably, although our model has been given no information about atomic charges, it assigns contributions that are in

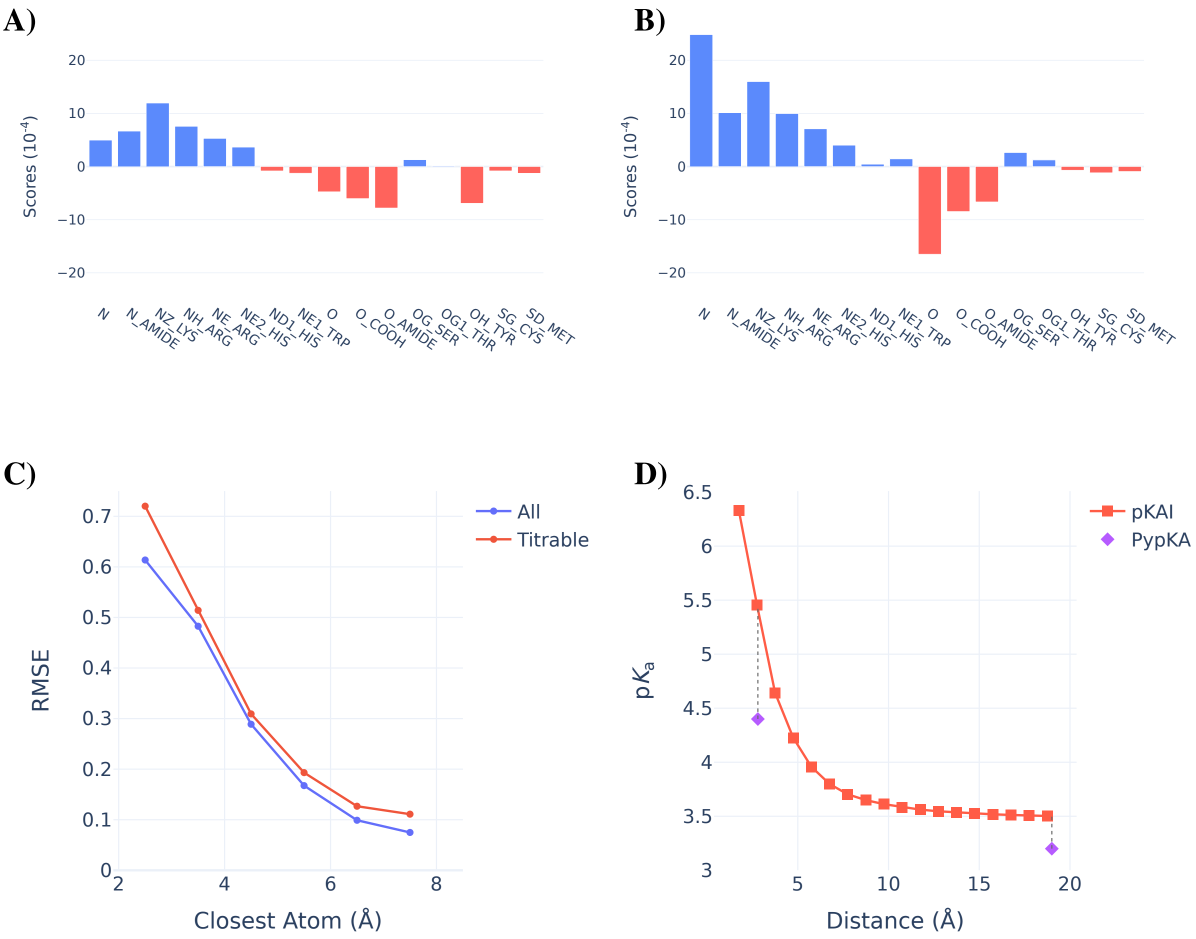


Figure 3.4: Charge scores attributed by pKAI to all considered input atoms classes (Supplementary Table C.6) of all atoms (A) and atoms closer than 6 Å (B). C) Closest atom influence on pKAI performance. D) Impact of changing the distance of the closest atom on pKAI predictions of residue TYR-315 from structure 2BJU. For reference, we have included PypKa predictions of the same residue in the state presented in the experimental structure (closest distance 2.8 Å) and in a modified structure in which the closest atom is absent.

agreement with the expected overall charge of the atom class. Cationic amine groups (NZ_LYS; NH_ARG; NE_ARG; NE2_HIS) are clearly assigned positive scores (i.e., destabilize the protonation of the titratable residue) and are easily distinguishable from the anionic carbonyl groups (O_COOH from Glu, Asp, and C-termini residues). These scores provide a general insight into the network's interpretation of each atom and should not be used for more quantitative analysis. Since the atom score is an averaged measure across the test set, an imbalance of closely interacting atoms of a specific class can dramatically skew its median contribution.

Hydrogen bonds are one of the strongest interactions found in proteins, and, as such, their proper description is crucial to obtain accurate pK_a predictions. By comparing Figures 3.4 A and B we can observe marked differences between the atom scores at close proximity and those farther away from the titrating residue. For example, the average score of the very abundant classes of primary amines (N; N_AMIDE) and carbonyl groups (O; O_AMIDE) is greatly diminished when compared to their short-range contributions, where these become hydrogen donors and acceptors, respectively. The anionic Tyr residue is perceived to have an overall negative contribution, except when it is close to another titratable residue; in this case, there seems to be no preferred state as it can act both as a donor and as an acceptor – like any titratable residue. On the other hand, the contribution of neutral non-titrating alcohol groups (OG_SER; OG1_THR) is almost exclusively attributed to their potential to form hydrogen bonds at short range.

Beyond the general understanding shown before, hydrogen bond contributions are hard to account for compared to other interactions. As shown in Figure 3.4C, the closer another residue (blue curve) is to the titrating one, the harder for the model is to correctly describe their interaction. The difficulty of the prediction increases dramatically at the typical distance of hydrogen bonds (2.5-3.2Å). This is even more marked if one considers interactions established between two titratable residues (red curve). In this case, the network has to solve for the pK_a of both residues simultaneously, and in many instances, it is unable to do so. Hence, predicting the contribution of the remaining environment is easier than that of a single hydrogen bond. This is illustrated in Figure 3.4D, in which the agreement with the physics-based method is much higher when the closest atom is removed from the structure rather than when it is kept in its original position. Although many other profiles can be observed (Supplementary Figure C.6), this trend is generally conserved. Considering that the model did not receive explicit information about hydrogen bonds, it is quite remarkable that it was able to correlate this type of interaction with larger pK_a shifts.

Solvent exposure is another property that is usually a key contributor to pK_a shifts. The models are trained without explicit knowledge of the 3D structure of the protein and deprived of information regarding carbon atoms. Nevertheless, they seem to learn about the solvent exposure contribution. We compared the correlations (Pearson correlation coefficient r and Spearman's rank correlation coefficient ρ) between the calculated SASA and the pK_a shifts over the entire test data set. Using the known ΔpK_a, we obtained $r_{\Delta pka} = -0.68$; $\rho_{\Delta pka} = -0.60$, while using the predicted ΔpK_a, we got $r_{pred} = -0.66$; $\rho_{pred} = -0.62$. The similarity between these values indicates that the model has learned the correct correlation between SASA and the pK_a shift. Additionally, we tested different solvent exposure metrics as an additional input and observed virtually no performance improvement (Supplementary Table C.7).

Finally, it is worth mentioning that the XAI analysis was a driving factor in the development of pKAI. In fact, the importance that the model assigns to each atom class (similar to Figure 3.4) was pivotal in selecting the final set of atom classes aimed at describing the surrounding environment residues.

3.1.5 Discussion

We have introduced pKAI and pKAI+, two deep learning models to predict theoretical and experimental ΔpK_a values, respectively. pKAI offers unprecedented efficiency, exhibiting a remarkable trade-off between accuracy and computational speed, its performance rivaling that of CE-based methods, such as PypKa. pKAI could be used as a replacement for such methods, especially when dealing with large proteins or applications requiring multiple CE calculations, like constant-pH MD simulations [4, 177, 178, 231, 295]. Considering the latest advances in sequence to structure predictions [252], faster methods, such as pKAI, will likely be of use as exponentially more structures become available. Furthermore, when optimizing new structures for binding to specific targets (e.g., design of enzymes and/or antibodies), it is vital to have an accurate prediction of the protonation states.

While we strive for optimal accuracy, we are aware that many applications will only require a binary decision (hence a qualitative prediction of pK_a shifts would be sufficient). For example, when selecting the most likely protonation state of a protein for running MD simulations, one only needs to predict whether each pK_a is larger or smaller than the pH value of interest. As intended, pKAI shows a performance similar to that of a PB-based model. Furthermore, it significantly surpasses PROPKA and the Null model in the physiological pH range.

Several other applications only require an estimation of the proton binding affinity using a fixed conformation. This quantity, termed pK_a^{single}, renders a good prediction of the macroscopic pK_a when averaged over a representative ensemble of conformations. From pK_a^{single} values, the most abundant/representative protonation states for a particular conformation can be calculated, improving the realism of methods such as molecular dynamics [4, 177, 178, 231, 295] and molecular docking [362]. pKAI is nearly perfect at mimicking representative protonation states given by PypKa, being particularly effective at physiological pH, achieving an astounding accuracy of 99.4% (Supplementary Figure C.5). In a conformational ensemble, there are always many representative protonation states which differ significantly from the one calculated using the macroscopic pK_a values. Therefore, coupling pK_a^{single} calculations with conformational sampling techniques is very appealing in theory but difficult in practice due to their computational cost. By using pKAI instead of PypKa (or any other PB-based method), one would drastically decrease the computational overhead (up to 1000×).

pKAI does not handle all residues with the same performance. Difficult cases are caused by low representation in the training set, low solvent exposure, and/or close-by residues providing H bond interactions. These peculiar environments usually present a high ΔpK_a which is not handled very well by the method. One clear way to improve our models would therefore be to introduce more training examples. Furthermore, the inclusion of more training data with rare environments would definitely enhance performance. To avoid limiting the scaling rate by the availability of new experimental protein structures, we plan to generate new uncorrelated protein structures using conformational sampling methods, such as MD and MC. Another advantage of using computational methodologies is guiding the protein conformational sampling to achieve electrostatic environments that are underrepresented in the training set. To better handle interactions with neighboring titratable groups, a change of environment encoding would be needed. One approach to be explored in future work would be to represent the whole protein as a graph and use graph neural network algorithms to learn the ΔpK_a values.

Although pKAI excels at predicting $pK_\mathrm{a}^\mathrm{single}$ values, its performance is modest when estimating experimental pK_a values. Inspired by the observation that increasing the dielectric constant in PB-based methods improves their agreement with experimental results, we have introduced a regularization parameter into the cost function. Similar to the dielectric constant, this regularization weight biases all predictions towards the residue's pK_a values in water. The new model, pKAI+, outperforms all methods tested in this work, including PypKa, which was used to create the training set. However, this improvement, while significant for partially exposed residues that would otherwise exhibit overestimated pK_a shifts, penalizes the accuracy of more shifted residues.

In this work we made the conscious decision of training our models solely on theoretical pK_a values, and to use all the available experimental data as a test set. The reason for this choice is twofold. First, there are not enough experimental data points to successfully train large models like deep learning ones. This issue could be circumvented with pretrained embeddings, assuming these representations hold the necessary information for the new task. Gokcan *et al.* have used molecular representations encoding quantum mechanical information to obtain a neural network model with an RMSE of 0.5 – 0.75 for most titratable residues [156]. The second problem with this approach is that the available data is quite limited in variability. Since a model trained on experimental data will not be exposed to a wide variety of environments, in real-world applications it will likely need to extrapolate in many cases. Both these issues contribute to the risk of model overfitting, and poor generalizability. Chen *et al.* trained tree-based machine learning models, such as XGBoost or LightGBM, on experimental data and their best model exhibited an RMSE of 0.69 [155]. To compare pKAI with these models, and illustrate the data leakage problem at hand, we have refined our pKAI model by training it on same data split reported in reference [155]. This new model seems to have an unparalleled performance (RMSE of 0.32; MAE of 0.21). However, this level of accuracy is unlikely to be expected for a rigid body calculation due to the missing entropic information. Furthermore, at the moment there are only 18 and 23 experimental pK_a values reported for Cys and Tyr residues, respectively. Even considering some degree of information transfer from other residue types, it is extremely unlikely that a few dozens of residues are able to convey enough information to create a model with a robust predictive ability at inference. Contrarily, pKAI was trained on millions of environments, and as such, we believe that the reported performance estimates are a much better reflection of its predictive ability. Finally, it must be noted that experimental data (both structures and pK_a values) should not be taken as absolute truths with no associated errors. In fact, old measurements of a popular benchmark protein (hen egg-white lysozyme) have been evaluated with modern NMR spectroscopy, and discrepancies of more than one pH unit have been found [138]. It is reasonable to assume that at least some of the $\approx$1k available experimental values have comparable errors, which only reinforces the importance of blind prediction exercises such as the pK_a Cooperative [179].

With pKAI and pKAI+, we are introducing the first deep learning-based predictor of pK_a shifts in proteins trained on continuum electrostatics data. The unique combination of speed and accuracy afforded by our models represents a paradigm shift in pK_a predictions. pKAI paves the way for accurate estimations of macroscopic pK_a values from ensemble calculations of pK_a^{single} values, overcoming previous computational limits. By design, the models were trained using a very simplified view of the surroundings of the titratable group, accounting only for residues within a 15 Å cutoff and ignoring all carbon and hydrogen atoms. This informed design choice allowed for the models to stay small and fast. Explainability methods confirmed that this input information was enough for the model to capture crucial features such as electrostatics, solvent exposure, and environment contributions. The models' initial success introduces several opportunities for further research, including problem encoding, accounting for conformational flexibility, interactions with other molecule types (i.e., small molecules, nucleic acids, lipids), and adding further target properties that could be of interest for other applications.

3.1.6 Acknowledgements

We would like to thank Paulo Martel and Diogo Vila-Viçosa for the fruitful discussions, as well as the attendees of the Protein Electrostatics 2021 meeting (www.proteinelectrostatics.org). We thank Artemi Bendandi for proofreading the manuscript.

PR and MM acknowledge financial support from FCT: SFRH/BD/136226/2018, CEECIND/02300/2017, UIDB/04046/2020, and UIDP/04046/2020. This work benefited from services and resources provided by the EGI-ACE project (receiving funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017567), with the dedicated support from the CESGA and IN2P3-IRES resource providers. MB and FM acknowledge funding from the Bayer AG Life Science Collaboration ("Explainable AI").

3.2 pKAI-MD: Towards AI-accelerated Constant-pH Molecular Dynamics

Pedro B.P.S. Reis*, Tuan Le, Floriane Montanari, Djork-Arne Clevert, and Miguel Machuqueiro*

3.2.1 Context

Even though our first machine learning pK_a predictor, pKAI (Chapter 3.1), proved to be quite successful, we knew from the beginning that it would not be suitable for integration into a CpHMD implementation (Chapter 2.2). Nevertheless, our simple proof-of-concept model demonstrated that the opportunity to accelerate CpHMD with machine learning was valid. Then, we set out to address two of the most severe limitations of pKAI, hindering its usage in CpHMD: pK_a values as the target property, which would lead to incorrect protonation state sampling; the inability to deal with closely interacting titratable residues. This work is not yet published as the integration with the CpHMD's MC routine is not yet finished.

3.2.2 Abstract

In classic molecular dynamics simulations, the protonation state of the system is fixed. Hence, conformational sampling is performed at an undefined pH value. Constant-pH molecular dynamics (CpHMD) methods solve this problem by coupling conformational and protonation sampling. Unfortunately, this method can be considerably more computationally expensive. Here, we propose a new GNN-based protein pK_a predictor suitable for faster CpHMD simulations. This model estimates pH-independent energies to be used in a Monte Carlo routine to sample representative microscopic protonation states. While developing the new model, we explored different graph representations of proteins using multiple electrostatics-driven properties. We also illustrate the benefits of using GNNs over feedforward neural networks.

3.2.3 Introduction

Constant-pH molecular dynamics (CpHMD) methods are powerful tools for studying pH-dependent systems [178, 214–217]. Several of these methods combine molecular dynamics (MD) and continuum-electrostatics (CE) to couple the sampling of conformations and protonations and bind them to a specific pH value. This approach has been used extensively to provide molecular insights about many biologically-relevant pH-dependent systems such as virus fusion peptides [326], neurodegenerative target proteins [363], transmembrane proton channels [364], and therapeutic antibodies [365].

In classic MD simulations, a protein is constrained to a single protonation state. However, at the physiological pH range, a protein transitions between dozens to hundreds of states. For example, at pH 7, a small protein like lysozyme, with only 29 titratable residues, can display 24 unique protonation states, and at pH 4, as many as 222 microstates can be observed [172]. Thus, in many MD simulations, the protein is simulated at undefined pH values with its conformational sampling not representative of any meaningful physical condition. Hence, the importance of CpHMD methods is not limited to the so-called pH-dependent phenomena. Instead, it should be regarded as a revised or more realistic MD. Unfortunately, CpHMD has not yet reached mass adoption by researchers, as for most, the better accuracy does not compensate for the higher computational cost.

Recently, machine learning models have been trained to speed up pK_a calculations. Furthermore, two models have been developed to mimic theoretical predictors and used physics-based models' predictions as training data. DeepKa was developed to reproduce CpHMD-derived pK_a values [156]. However, this model was trained to predict macroscopic pK_a values from a single conformation. Therefore, it can not be used to differentiate between individual states. pKAI is another deep learning model trained on CE-derived pK_a values [3]. Like the original method, pKAI can predict the pK_a value of different conformations. In fact, the physics-based method (PypKa [1]) used to train pKAI is the same one powering a CpHMD implementation (PypKa-MD). pKAI replicates theoretical pK_a values at a fraction of the computational time, with speedups up to 3 orders of magnitude. It was shown that pKAI has a reasonable understanding of the underlying physics and extrapolation ability. However, this model struggles with closely interacting residues, as it treats the environment of one residue at a time. Furthermore, it is not possible to use pK_a values alone to accurately derive protonation states of coupled residues. In several CpHMD methodologies, the protonation states are sampled with Monte Carlo (MC) simulations from the PB-derived energies: the intrinsic pK_a value (pK_{intr}), a pH-independent quantity representing the pK_a of a residue when all others are at their reference state; and, the interaction energies between titratable residues.

Some of the limitations exhibited by pKAI could be solved, at least partially, by using graph neural networks (GNN). Proteins have long been depicted as graphs in which nodes represent atoms and edges describe covalent bonds. Instead of covalent bonds as a criterion for edge attribution, one could use the closest atoms for better information passing from and with the surrounding environment. Furthermore, during training, the model considers nearby titrating residues at once. Also, labile hydrogen atoms can be included in the graph, which along with the 3-dimensional information unavailable to pKAI, should improve the model's internal representation of residues involved in hydrogen bonding. Recently, there has been a growing number of work applying GNN to proteins [366–370]. However, in most cases the side chain is not accurately described, which may be beneficial when dealing with macroscopic properties.

In this work, we developed a new GNN-based protein pK_a predictor suitable for CpHMD integration. This model estimates pK_{intr} and interaction energies and will be coupled with an MC algorithm to provide the representative microscopic protonation states required by CpHMD. Before training this model, we explored how to best encode a protein for a GNN using multiple electrostatics-driven properties, and show the performance benefits of using GNNs over feedforward neural networks (FNN) for pK_a predictions.

3.2.4 Methods

In this work, all models are based on the same E(n) equivariant graph neural network (EGNN) core architecture [274]. The inputs of the EGNN are the 3D coordinates and labels of the protein atoms. We explored which atoms to include, as well as the labeling strategy. The default input atoms and labels are taken from previous models that predict pK_a values in proteins, the pKAI 1.0 model [3]. In these models, only charged atoms like nitrogen, oxygen, and sulfur are included, and the labels are set according to the chemical group. To these, we have added the labile hydrogens in different tautomeric positions. The EGNN's output atomic embeddings are then used directly or with pooling to train FNN for different tasks.

In the "Protein encoding for GNNs" section, the model is trained end-to-end on multiple tasks spanning different resolutions: at the atomic level, the titration curve of individual tautomeric hydrogens (P(H)); at the residue level, the pK_a value of titratable residues; at the protein level, the isoelectric point (pI). The details of each task are reported in Supplementary Table D.1. For the isoelectric point FNN, the input is the average of all node embeddings. To predict pK_a values, we tried pooling different combinations of residue-belonging atoms, such as the charged atoms (N, O, and S), the hydrogen atoms (H), and the carbon atoms (C_α, C_β) (Figure 3.5). Like the pKAI models, the NN was trained on ΔpK_a values instead of the absolute pK_a values. ΔpK_a can be obtained by subtracting the pK_a value of a residue in water from its value in a particular protein environment. The predictions of the hydrogen atom's occupancy at different pH values (ranging from -6 to 20 with a 0.5 step) were performed using the embedding of the corresponding H atom. All tasks were trained simultaneously, and to evaluate the performance of the multi-task model at training time, we have defined the loss function as the average of the

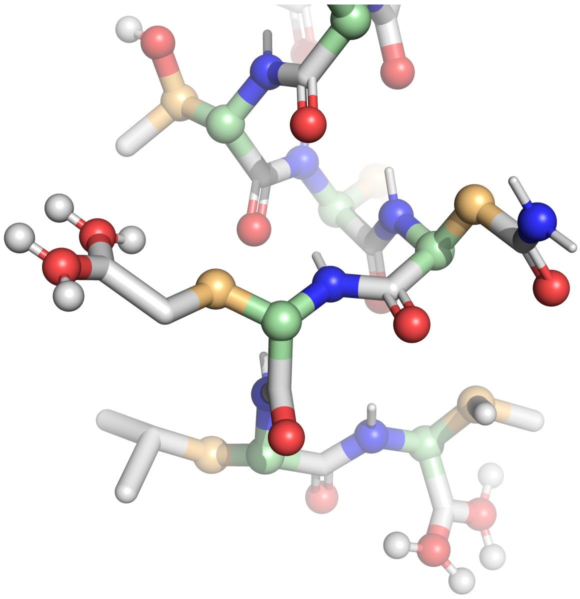


Figure 3.5: Illustration of a protein structure as sticks with atom included in the graph representations highlighted as spheres: hydrogen nodes in white, charged atom nodes in red and blue, C_α nodes in green, and C_β nodes in orange.

individual task losses. For the pI and pK_a tasks, the L2 loss was selected, and the P(H) task used the binary cross-entropy loss.

In the "GNN-based pK_a predictions" section, three models are trained for separate tasks: pK_a values, the intrinsic pK_a (pK_intr), and interaction energies between titratable sites. The model trained on pK_a values is referred to in this work as pKAI 2.0. The intrinsic pK_a value is the pK_a of residue with all other residues in a reference state. As a pH-independent quantity, it is a useful quantity to combine with the interactions between titratable residues to define the energy shift of a protonation change in site n of a protein with N titratable sites,

$$\Delta\Delta G_{n1\rightarrow n2} = 2.3k_BT\left[a_{n1}\gamma_{n1}\left(pH - \mathrm{p}K_\mathrm{intr}(n2)\right) - a_{n2}\gamma_{n2}\left(pH - \mathrm{p}K_\mathrm{intr}(n1)\right)\right] + \sum_{j\neq n}^{N}\Delta G_{n1\rightarrow n2,jx} \tag{3.3}$$

where a_n is the ionization state (0 for neutral; 1 for ionized), and γ_n is the charge (+1 for cationic; -1 for anionic) of site n. The difference in the interaction energy of site n with site j in fixed protonation state x due to the protonation change of n from $n1$ to $n2$ is given by,

$$\Delta G_{n1\rightarrow n2,jx} = G_{n2,jx} - G_{n1,jx} \tag{3.4}$$

Having pK_intr for all titratable residues and all possible interaction energies, it is possible to sample different protonation states with Monte Carlo (MC) using the Metropolis criterion [167, 212], analogously to that is already done in the stochastic titration CpHMD method to periodically update the MD simulation. All models were trained using an L2 loss.

The DL models presented in this work have been implemented with PyTorch v1.9.0 [354], PyTorch Lightning v1.3.8 [355] and PyTorch Geometric v2.0.3 [371]. The node embeddings of the EGNN had a size of 256, and 3 graph convolutional layers were used. Each node was connected to the 32 nearest nodes. All FNNs were comprised of two fully connected linear layers of input sizes 256 and 64, using a SiLU activation function [372]. During training, we used a batch size of 16 and an Adam optimizer with a learning rate equal to 10^{-4}.

The pKPDB database contains around 12M theoretical pK_a values and 120k isoelectric points [2]. These values were estimated from experimental data structures deposited in the Protein Data Bank with PypKa [1] – a Poisson–Boltzmann-based method – using default values, as described elsewhere [3]. To easily compare our results with pKAI 1.0 while ensuring that proteins with a high degree of similarity did not appear in the training and validation sets, the data split adopted to train pKAI 2.0 is the same used for the previous model [3] (Supplementary Table D.2). The multi-task models were trained on a subset of proteins with one chain to expedite the data preprocessing and training processes (Supplementary Table D.3). Part of the data necessary to develop the models presented in this work was not available in the pKPDB. The original database was extended with the titration curves of individual tautomeric hydrogens. These calculations were performed by PypKa using the same default parameters used for the other properties. An extra data set comprised of pK_{intr} values and interaction energies was created. In these calculations, instead of the default protein dielectric constant 15 used thus far to calculate all other properties, a dielectric of 2 was set, as required for usage in CpHMD. This data set of pK_{intr} and interacting energies is not yet finished (Supplementary Table D.4), thus the results presented here are only preliminary.

3.2.5 Results & Discussion

Protein encoding for GNNs

Protein structures can easily be represented as graphs, and in recent years there has been a growing number of work applying graph neural networks (GNN) to protein-related tasks [366–370]. However, virtually all GNN applications encoding protein structures aim at predicting macroscopic properties, such as binding affinities, from a single conformation. In these models, it is common to depict the protein as a collection of residues represented by the C_α, analogously to the 2D residue distance matrix traditionally used in convolutional neural networks (CNN). One can argue that for macroscopic properties, providing only information regarding the backbone position and modeling the side chain implicitly might be enough since the property depends on a number of side chain positions. Nevertheless, if the objective is to obtain energies with which to perform the ensemble averaging, or if the property to be predicted depends solely (or significantly) on a single conformation, considering the side chain positions explicitly becomes essential.

To compare different approaches to encoding a protein for a GNN, we trained several models in which the atoms chosen to represent the protein were changed and tested their average performance at predicting three electrostatics-driven properties. Even though the tasks are related, they are predicted using embeddings of distinct resolutions. The network uses the embeddings of individual atoms to predict the probability of a hydrogen atom occurring at different pH values (P(H)). pK_a predictions require a residue embedding, and an embedding representing the entire protein is needed for isoelectric point (pI) estimations. The baseline graph features nodes of all nitrogen, oxygen, and sulfur atoms, as well as all labile hydrogen atoms. This representation is inspired by the pKAI model that obtained a good accuracy at predicting pK_a values, despite the relatively simple residue environment characterization of only nearby charged atoms [3]. The performance of our baseline model is shown in Figure 3.6A. Despite target pI values having a broader range compared to ΔpK_a values, the former task is easier since pI predictions benefit from error cancellation effects. Furthermore, by definition, probabilities can only take values between 0 and 1, resulting in a considerably lower RMSE for the P(H) task. Thus, the average cost value of all tasks used for training is less sensitive to variations in the performance of P(H). In order to compare the performance of the different models, it is helpful to check the cumulative improvement of the individual tasks shown in Figures 3.6B-D.

In our protein graph, each node represents an atom, and it is connected to the 32 nearest nodes. Although no edge attributes are required to distinguish between nodes, defining at least one node feature is necessary. In the baseline model, atoms were labeled according to their chemical groups, following the same nomenclature as that of the pKAI model. The proposed atom classes perform better than the more common approaches of using the atom or element name (Figure 3.6B). By classifying atoms according to their element name, a considerable amount of information is lost. It becomes extremely hard for the network to differentiate between certain atoms, such as a nitrogen in the main chain and one in the side chain of a lysine. Contrarily, when labeling atoms by their atom name, the network cannot transfer information between chemically identical atoms as easily. For example, in the baseline model, any oxygen belonging to a carboxylic acid has the same atom class instead of the multiple atom names found for oxygens in aspartates (OD1, OD2), glutamates (OE1, OE2), and C-terminal (O, OXT) residues.

Which atoms to include as nodes is arguably one of the most important decisions when constructing the molecular graph. The overall shape of a protein is given by its backbone, commonly depicted by C_α atoms. However, using a protein graph composed solely of C_α atoms for electrostatics-related properties yields subpar performance (276.2% and 772.4% worse at pK_a and pI tasks, respectively). Moreover, complementing our graph of charged atoms with either C_α or C_β atoms results in a minor performance improvement (Figure 3.6C). C_β atoms contain more information regarding side chain position compared to C_α, which might explain the discrepancy in performance. Furthermore, the aliphatic residues, which would otherwise contain no side chain nodes, seem to contribute the most to the performance gain of the C_β addition (Figure 3.6D). Since all amino acids contain a C_α, it can be interesting to have it in

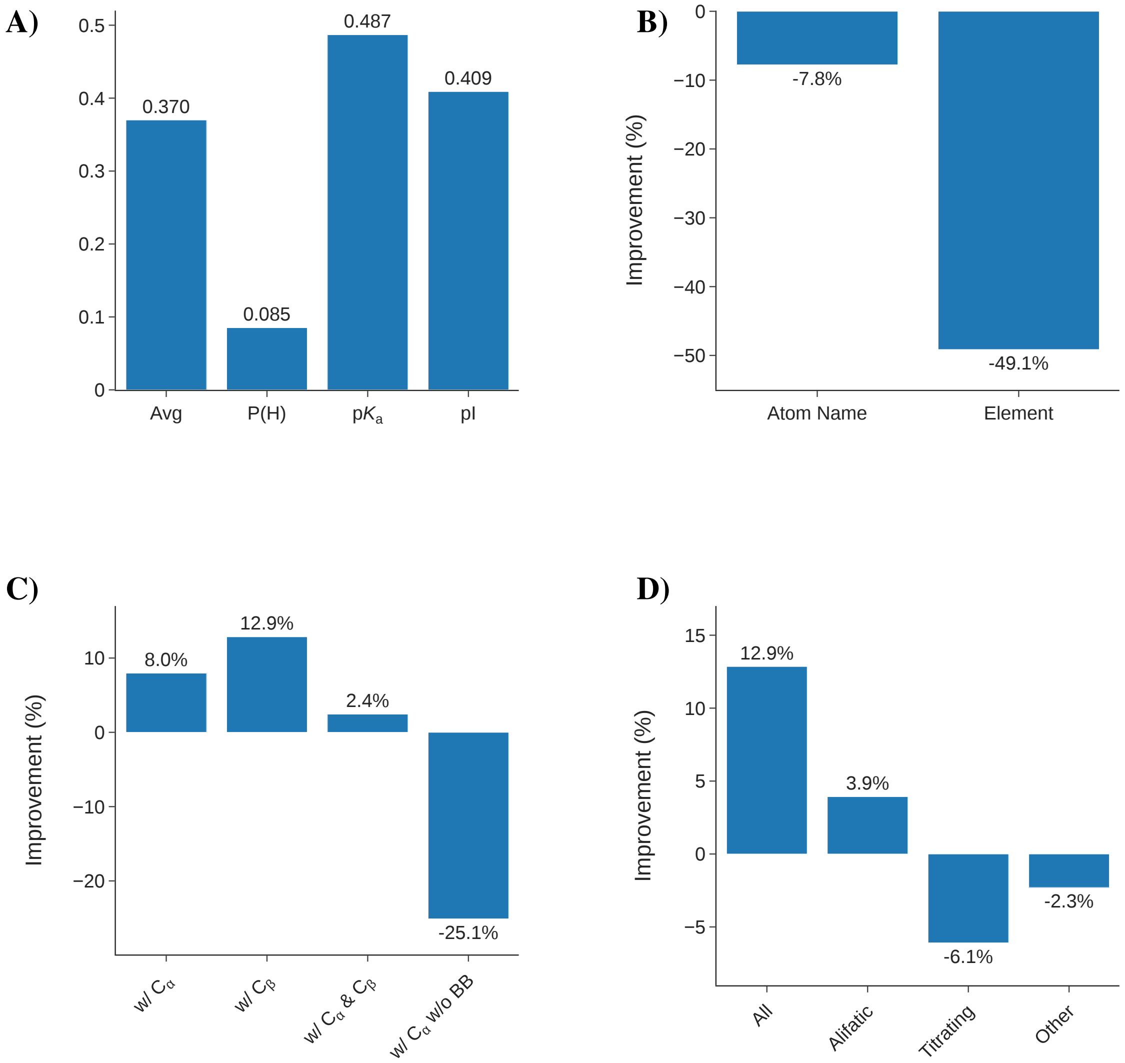


Figure 3.6: A) Multi-task GNN cost and individual tasks RMSE. B) Multi-task performance improvement of classifying nodes by atom names or element, over custom atom classes (Supplementary Table D.5). C) Multi-task performance improvement of including different atoms (C_α, C_β) over the default of having a graph comprised only of polar heavy atoms (Pl). BB stands for backbone polar atoms (N, P), omitted in Pl + C_α - BB. Individual task performance shown in Supplementary Table D.6. D) Multi-task performance improvement of including all C_β atoms, only C_β of alifatic residues, only C_β of titratable residues, and only C_β of non-alifatic nor titratable residues (Other). Individual task performance shown in Supplementary Table D.7.

Table 3.1: Performance of different residue embeddings for the prediction of multiple electrostatics-related tasks.

	Avg	P(H)	pK_a	pI
Charged	0.362	8.36×10^{-2}	0.469	0.407
CB	0.402	7.58×10^{-2}	0.580	0.377
Hydrogen (H)	0.419	9.38×10^{-2}	0.542	0.472
Charged + CB	0.370	8.59×10^{-2}	0.479	0.417
Charged + H	0.354	8.12×10^{-2}	0.457	0.401
CB + H	0.367	8.18×10^{-2}	0.464	0.427
Charged + CB + H	0.366	8.63×10^{-2}	0.472	0.416

the graph as a single node representing the whole residue. Alternatively, a virtual node that connects only to other atoms from the same residue could be added. Interestingly, adding both C_α and C_β nodes results in poorer performance. A possible interpretation is that both atoms describe roughly the same information and consequently dilute the more relevant message from charged atoms. To test whether the backbone could be approximated by the C_α, we trained a model without the main chain oxygen nor nitrogen atoms. Its performance was significantly worse (33.8%) than the model with the charged atoms of the main chain. This result shows that for electrostatics-related tasks, the often overlooked main chain oxygen and nitrogen atoms should not be replaced by C_α nodes.

To predict a property of an atom, it is straightforward to take its corresponding node embedding of the GNN last layer as the input feature. However, dealing with residue properties is slightly more complicated since there are multiple nodes associated with each residue. The two most obvious solutions are to use the only single node in all residues, the C_α, or to create a residue embedding by averaging all its atom embeddings. Instead of the C_α, we have used the C_β since it was deemed to contain more relevant information. Interestingly, neither of the two approaches leads to the best performance (Table 3.1). A residue embedding comprised of only the charged and hydrogen atoms seems to be the most informative. Furthermore, using such an embedding for the residue task positively impacts the atomic and protein predictions. In other words, the other tasks benefit from improving the internal learned representation of the charged and hydrogen atoms. It should be stated that residue embeddings combining two or more types of nodes display comparable performances (4% difference between the best and the worse). However, there is a significant 12% improvement to be gained from using charged and hydrogen atoms rather than the C_β alone. Also, these results might be dependent on the tasks, even though similar relative performances are expected for estimations of other electrostatics-driven properties.

There is an abundance of protein and electrostatics-related tasks with limited available data.

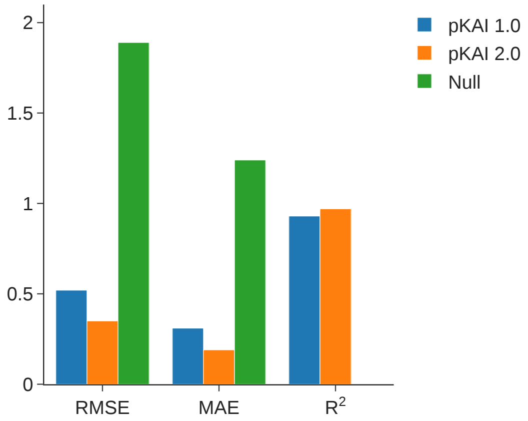


Figure 3.7: Null model, pKAI 1.0, and pKAI 2.0 performance. For more details, see Supplementary Table D.8.

The internal representation of the protein learned by the models trained in this work may be a valuable embedding for such tasks. We are currently investigating the applicability of these pre-trained embeddings and trying to deconvolute the information contained. Meanwhile, it is possible to try these protein graph-based embeddings as a python package available at github.com/bayer-science-for-a-better-life/pege.

GNN-based pK_a predictions

In the previous section, we tested several strategies to encode a protein for GNNs. We found that the model trained on a graph that included C_β atoms and all charged atoms yielded the best performance on several electrostatics-driven properties. We have also shown that by combining the node embeddings of the charged atoms and hydrogens into a residue embedding, the model could improve its internal representation of the protein. In this section, we used these approaches to predict only one property: pK_a values. The new model, pKAI 2.0, is an upgraded version of our previous FNN-based pK_a predictor pKAI [3]. In pKAI 1.0, each residue environment was described by the distances, and chemical group of atoms within a 15 Å cutoff. Conversely, in the graph representation it is possible to keep the 3D information of each configuration. Moreover, instead of training the model one residue at the time, all titratable sites are considered in the GNN backward pass.

The novel graph-based model achieved an impressive accuracy at predicting pK_a values, with an RMSE of 0.35, an MSE of 0.19, and, an R^2 of 0.97 (Figure 3.7). To evidence the degree of difficulty of the data set, the null model – the pK_a of a residue in water – is also shown (RMSE: 1.89; MAE: 1.24; R^2: 0.00). As expected, the pKAI 2.0 model outperforms the already

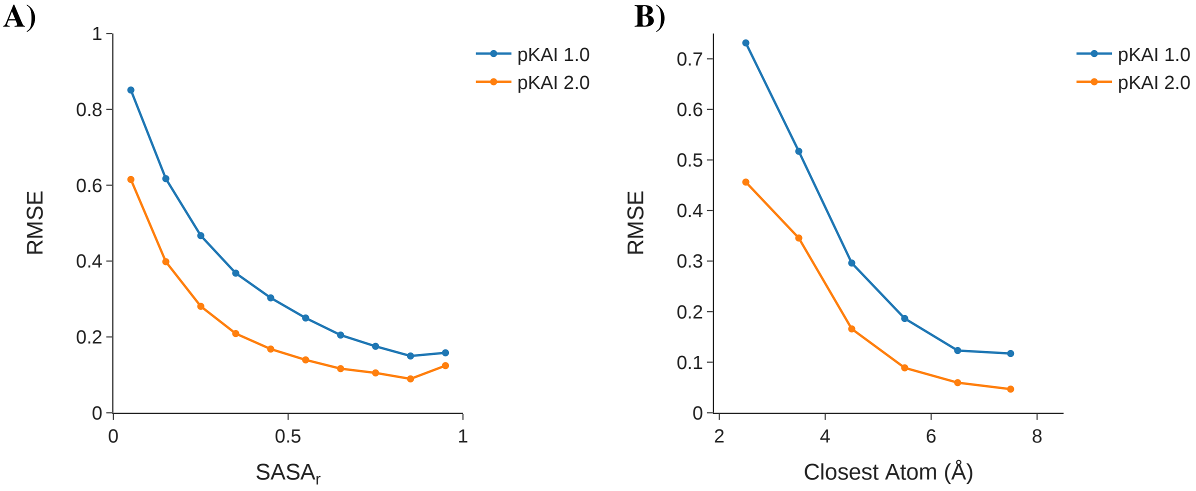


Figure 3.8: pKAI 1.0 and pKAI 2.0 performance dependence on the solvent accessible surface area (SASA) (A) or the closest interacting titratable residue (B).

quite competitive predecessor. Furthermore, the performance gain occurs in all amino acid types, with the single exception of the N-terminal (Supplementary Table D.9). Considering the predicting ability in the similar Lys and C-terminal residues, the N-terminal discrepancy is surprising and warrants further investigation. Unfortunately, the higher accuracy of pKAI 2.0 is likely the result of a more computationally costly calculation. An extensive speed benchmark will also be performed once pKAI 2.0 is integrated into the available pKAI module.

From the user point of view, the most valuable predictions are usually of buried residues significantly deviated from their pK_a values in water. In contrast, solvent-exposed sites tend to interact mostly with water and, thus, exhibit small shifts. As such, it is important to investigate the solvent exposure influence on the performance of a pK_a predictor. Figure 3.8A shows that the improvements introduced by the graph-based model are useful for the end user since pKAI 2.0 is consistently better in all degrees of solvent exposure, and the difference between the two models is more marked for highly shifted residues (Supplementary Figure D.1). Strong pK_a shifts are usually caused by desolvated environments and by the presence of closely interacting residues. So, naturally, the same trend observed for the performance dependence on solvent exposure can be seen for the proximity of the nearest atom (Supplementary Figure D.2). In addition, residues with neighboring titratable residues that were particularly challenging for the original model have been greatly improved in pKAI 2.0 (Figure 3.8B). This impressive accuracy increase is likely due to the GNN's ability to be trained on all protein titratable residues simultaneously.

Unfortunately, the limitations when dealing with coupled titratable residues have only been partially addressed in pKAI 2.0, which restricts its use in a CpHMD framework. In each CpHMD cycle, it is necessary to sample a protonation state from the correct ensemble. So far, our models have only been trained to predict pK_a values. Calculating protonation states from pK_a values implies large approximations. A major one is that sites do not influence each other's titration. Hence, this approach, which has already been pursued by other groups [235], is not suitable for a proper CpHMD method. In a typical CpHMD with discrete protonation states, several Poisson–Boltzmann calculations are performed to derive the pK_{intr} values of each site and the interaction energies between them. These quantities are then used in a Monte Carlo (MC) calculation to sample the required protonation states. Currently, we are exploring the possibility of training a GNN to predict these energies and later integrating it in an MC routine. The preliminary results are very encouraging as it was possible to train models that presented low RMSE values of 2.17 (MAE: 0.88), and 0.17 (MAE: 0.03), for the pK_{intr} values, and the interaction energies, respectively. Once more training data is available, the performance of these models is expected to improve further. In this data set, a protein dielectric constant of 2 is used instead of the usual 15 for pK_a calculations. The higher value of the internal dielectric constant accounts for the conformational flexibility absent in rigid body pK_a calculations. In a CpHMD simulation, this effect is explicitly modeled, and thus, a dielectric of 2 is used to account for the missing contribution of electronic polarization [178]. The lower dielectric constant leads to higher magnitude pK_a shifts and, consequently, to a more challenging data set for an ML model. There is an ongoing effort to integrate our model into a CpHMD framework (PypKa-MD) and test its performance and accuracy. In the future, this model may be employed to accelerate other applications, such as MMPBSA.

3.2.6 Conclusions

In the first part of the work, we explored different approaches to encode protein structures for GNN. To evaluate the success of the strategies, we trained the models to predict three electrostatics-driven properties and assessed their performance. Several graphs were tested by changing the atoms included as nodes and their classification. The combination of atomic nodes into residue embeddings was also investigated. The overall best approach was the inclusion of all charged, labile hydrogen, and C_β atoms in the graph, using as residue embedding the average of corresponding charged and hydrogen nodes. In the future, we will apply the internal representation learned by the model to other electrostatics-related tasks with less available data.

With the best protein graph, we trained a GNN on theoretical pK_a values. This model (pKAI 2.0) exhibited a remarkable performance, beating our previous FNN-based model (pKAI 1.0). The accuracy increase of the GNN-based pK_a predictor can be explained by the extra 3D information of the input and by the possibility of training the model on all titratable residues of a protein simultaneously. This hypothesis is supported by the substantial improvement of the predictions in residues that strongly interact with other surrounding titratable sites.

Finally, we have reported preliminary results of a GNN trained on pK_{intr} values and interaction energies between all titratable sites. Predicting these energies instead of pK_{a} values is required for accurate sampling of protonation states for CpHMD. We are currently working on integrating this model into a CpHMD package. The ML-accelerated CpHMD method will then be subject to extensive testing and validation.

3.2.7 Acknowledgements

We would like to thank the Diogo Vila Viçosa, Paulo Martel, and Machine Learning Research group for the fruitful discussions.

PR and MM acknowledge financial support from FCT: SFRH/BD/136226/2018, CEECIND/02300/2017, UIDB/04046/2020, and UIDP/04046/2020. This work benefited from services and resources provided by the EGI-ACE project (receiving funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017567), with the dedicated support from the CESGA and IN2P3-IRES resource providers.

Chapter 4

Beyond the command line

4.1 pKPDB: a Protein Data Bank extension database of pK_a and pI theoretical values

Pedro B.P.S. Reis*, Djork-Arné Clevert, and Miguel Machuqueiro*

4.1.1 Context

In order to train machine learning models, like the ones presented in chapters 3.1 and 3.2, we needed to generate a large amount of training examples. Since we wanted to map the relation between a conformation and its pK_a value, we used PypKa (Chapter 2.1) to create the necessary examples. We ran it over thousands of structures from the Protein Data Bank so that the final database contained a diverse set of proteins. The precomputed values have been made available to download as a csv file or to query in the PypKa Server (Chapter 4.2). This work was published in Bioinformatics (DOI: 10.1093/bioinformatics/btab518).

Abstract

pK_a values of ionizable residues and isoelectric points of proteins provide valuable local and global insights about their structure and function. These properties can be estimated with reasonably good accuracy using Poisson–Boltzmann and Monte Carlo calculations at a considerable computational cost (from some minutes to several hours). pKPDB is a database of over 12M theoretical pK_a values calculated over 120k protein structures deposited in the Protein Data Bank. By providing precomputed pK_a and pI values, users can retrieve results instantaneously for their protein(s) of interest while also saving countless hours and resources that would be spent on repeated calculations. Furthermore, there is an ever-growing imbalance between experimental pK_a and pI values and the number of resolved structures. This database will complement the experimental and computational data already available and can also provide crucial information regarding buried residues that are underrepresented in experimental measurements. Gzipped csv files containing pK_a and isoelectric point values can be downloaded from https://pypka.org/pKPDB. To query a single PDB code please use the PypKa free server at https://pypka.org. The pKPDB source code can be found at https://github.com/mms-fcul/pKPDB.

4.1.2 Introduction

Ionizable residues play a central part in defining the structure and function of proteins. Changes in their preferred protonation state are usually associated with folding events, variations in enzymatic activity, and activation state transitions [283, 347]. The local environment of residues can be characterized by their pK_a values, while the global pH-dependence of the protein can also be inferred from its isoelectric point (pI) – the pH value at which the overall net charge of the protein is zero. These electrostatic properties, among others, provide helpful information to design experiments and create models of complex phenomena. As such, they are widely studied with both experimental and computational methods.

Recently, there have been growing efforts to compile experimental data, most notably the PKAD database [95] with $\sim$1500 pK_a values and the PIP-DB [373] with $\sim$5000 pIs. Experimental databases provide attractive benchmarking opportunities for computational methods. However, their size is modest in comparison with the number of available experimentally determined structures deposited at the Protein Data Bank (PDB) – $\sim$150 thousand as of 2020 [374]. This difference has been steadily increasing with time and could grow even faster with the development of novel AI structure predictors like AlphaFold2 [375]. pKPDB aims at bridging this gap by proving Poisson–Boltzmann-based theoretical pK_a and pI values for all structures in the PDB. A similar rationale has been used in the Proteome-pI database, which contains theoretical pIs of over 21 million protein sequences [376]. Nevertheless, the isoelectric points reported in Proteome-pI have been estimated using the Henderson–Hasselbalch equation with generic pK_a values for each residue type, which were fitted against experimental data. Unlike this approach, in pKPDB each isoelectric point is calculated from a theoretical titration curve where individual pK_a values are determined for all titratable residues of that protein.

4.1.3 Methods

pKPDB is a database of theoretical pK_a values calculated over the structures found in the PDB. At the moment, it is comprised of about 12.5 million pK_a values estimated from 120k protein structures. These correspond to the deposited proteins with less than 1000 residues; however, the database is currently being extended to include bigger proteins, and it will be updated as the PDB releases new structures. All information related to structures and proteins was retrieved using the PDB web services [377]. In order to calculate pK_a and pI values PypKa was used with default parameters [1]. This choice was motivated by the need for a pK_a predictor that could be easily automatized and with a balanced execution speed and accuracy ratio. PypKa exhibited an mean absolute error of 0.57 on a large benchmark [1], thus, the predictions featured in pKPDB should have a comparable accuracy. Sequence alignment clusters were also included in the database to easily identify structures of highly similar proteins. For each PDB structure, the corresponding cluster features all proteins with at least one similar chain. We considered chains

Table 4.1: Distribution of pK_a values by residue type and their relative solvent accessible surface area ($\mathrm{SASA_r}$).

Residue	p$K_\mathrm{a}$values (10^3)	pK_a values (%)	Average pK_a	$\sigma_{\mathrm{p}K_\mathrm{a}}$	Average $SASA_r$
GLU	3.1	24.83	5.52	0.94	0.43
LYS	2.8	22.44	9.93	0.54	0.47
ASP	2.8	22.00	5.46	1.06	0.40
TYR	1.8	13.92	10.55	0.83	0.19
HIS	1.2	9.34	6.67	0.64	0.29
CYS	0.5	3.92	10.54	0.87	0.11
CTR	0.2	1.82	3.99	0.81	0.73
NTR	0.2	1.73	7.68	0.49	0.75

as similar if they present a sequence identity greater than 0.9 (calculated with mmseqs2 [378]). Different solvent exposure metrics were determined with several biopython modules [379]: residue depth using msms [380]; relative SASA using DSSP [381]; half-sphere exposure with HSE [357]. pKPDB is a PostgreSQL database managed by Python3 and bash scripts.

4.1.4 Results and Discussion

pKPDB provides precomputed pK_a and pI values for all structures found in the PDB. It provides users with instantaneous results for their protein(s) of interest while saving countless hours and resources which would be spent on repeated simulations. Currently, the database contains results for just over 120 thousand structures (Supplementary Table E.1), which account for ∼67% of all protein structures available. The database does not yet include structures with complexes of protein and nucleic acids, although these will be added after simulating all protein-only structures. Unfortunately, there are also some protein structures too incomplete for PypKa to compute (about 2.5% of all attempts).

The distribution of the ∼8.5 million pK_a values shown in Table 4.1 follows the trend reported in [376] with the three most abundant titratable residues (glutamate, lysine, and aspartate) accounting for nearly 70% of all pK_a values in the database. Likewise, the 120 thousand isoelectric points distribution is compatible with those already reported [376] (Figure 4.1). In general, the average pK_a values of each amino acid are close to their values in water [19]), although some residues such as glutamates, aspartates, and cysteines show average shifts between 1 and 2 pH units. Not surprisingly, these are also the residues that exhibit the largest dispersion of pK_a values ($\sigma_{\mathrm{p}K_\mathrm{a}}$) which is probably related to them being exposed to the most heterogeneous environments. Water exposure is an environment property known to affect the magnitude of pK_a shifts. More buried residues lead to larger shifts (Supplementary Figure E.1), usually promoted

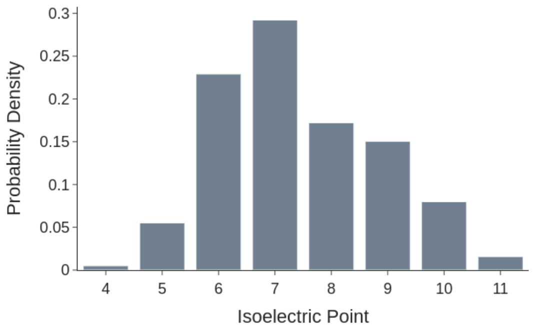


Figure 4.1: Probability density distribution of all isoelectric points in the pKPDB database.

by desolvation effects which favor the neutral states. The analysis of the residues solvent exposure yielded an intuitive correlation with hydrophobicity and natural distribution within the protein. The termini residues exhibit a high water exposure since they are located in the most flexible region of the protein. Contrarily, cysteine residues are often located in functional sites, and buried regions of the protein [114]. These are particularly underrepresented in experimental data, with only 20 pK_a values reported in wild-type proteins [95]. In fact, most environments and residues are underrepresented in experimental measurements, as evidenced by the different distribution profiles between the solvent exposure of the residues in pKPDB and the 1350 residues compiled in PKAD (Figure 4.2). Interestingly, there seems to be a shortage of highly buried and well-solvated residues. Since well-solvated residues display little to no pK_a shift, there is little motivation to spend resources on these measurements. On the other hand, the under-representation of buried residues can be explained by increased methodological difficulties. Hopefully, the release of this pKPDB database will contribute to expanding the knowledge regarding titratable residues, particularly these experimentally underrepresented ones, as well as to complement experimental and computational data already available for pK_a values and isoelectric points.

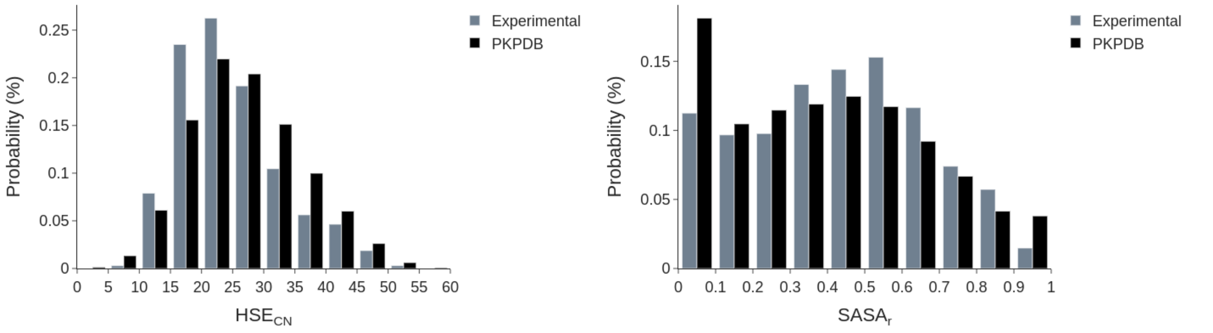


Figure 4.2: Probability of half-sphere solvent exposure (HSE_{CN}) and relative SASA ($SASA_r$) of all residues in pKPDB, compared with those found in the PKAD database [95].

4.1.5 Acknowledgements

We thank Diogo Vila-Viçosa, Floriane Montanari and Walter Rocchia for fruitful discussions.

This work has been supported by FCT: SFRH/BD/136226/2018, CEECIND/02300/2017, UIDB/04046/2020, and UIDP/04046/2020; and the EGI infrastructure: IN2P3-IRES, and NCG-INGRID-PT.

4.2 Online pK_a predictions and biomolecular structure preparation with precomputed results for the PDB

Pedro B.P.S. Reis*, Djork-Arné Clevert, and Miguel Machuqueiro*

4.2.1 Context

While CLI and API are useful types of interfaces for most researchers in computational fields, having an web application provides a graphical interface available for the already developed tools – PypKa (Chapter 2.1), pKAI (Chapter 3.1), pKPBD (Chapter 4.1) – that can be used by more users in any operating system (including mobile). This work is not yet published as there are key features currently being implemented such as the extension of the precomputed database with all the structures from AlphaFold, and the support to prepare structures for the CHARMM force field and CpHMD.

4.2.2 Abstract

When preparing biomolecular structures for molecular dynamics simulations, pK_a calculations are required to provide at least a representative protonation state at a given pH value. Neglecting this step and adopting the reference protonation states of the amino acid residues in water, often leads to wrong electrostatics and nonphysical simulations. Fortunately, several methods have been developed to prepare structures considering the protonation preference of residues in their specific environments (pK_a values), and some are even available for online usage. In this work, we present the PypKa server, which allows users to run physics-based, as well as ML-accelerated methods suitable for larger systems, to obtain pK_a values, isoelectric points, titration curves, and structures with representative pH-dependent protonation states compatible with commonly used force fields (AMBER, GROMOS). The user may upload a custom structure or submit an identifier code from PBD or UniProtKB. The results for $\sim$ 100k structures taken from the Protein Data Bank have been precomputed and their data can be retrieved without the need for extra calculations. We are currently working on adding data relative to all the proteins in AlphaFold DB. All this information can also be obtained from a REST API facilitating its usage and integration into existing pipelines as well as other web services. The web server is available at pypka.org.

4.2.3 Introduction

Before running popular molecular modelling methods like molecular dynamics (MD; e.g. GROMACS [331], NAMD [249], etc.), molecular docking (e.g. HADDOCK [382], AutoDock [383], etc.) or continuum electrostatics (e.g. APBS [165], DelPhi [164], etc.) it is necessary to prepare protein starting structures (experimental or models). It is often necessary to rebuild partially solved side-chain atoms in protein X-ray structures and add missing hydrogens, which are frequently absent. Furthermore, the conditions (like pH and ionic strength) at which the experimental structure was determined rarely coincide with the desired ones, which can affect significantly the ionization state of a protein [95, 209]. Thus, pK_a calculations are required to correctly place polar hydrogen atoms and to attain a representative protonation state at the user-defined conditions. pK_a estimations are useful not only to assign representative protonation states but also to provide molecular insights about the electrostatic environment of residues.

There are already several tools available that allow users to prepare their structures and/or obtain pK_a estimations. In the PDB2PQR web server [384] it is possible to process pdb files into clean pqr files with direct integration with APBS to easily run electrostatics calculations. To determine the most abundant protonation states of titratable sites, an empirical pK_a predictor (PROPKA [149]) is used, and in this server it possible to inspect the log files produced to find the pK_a predictions. Other pK_a calculator web servers, like H++ [169] and DelPhiPKa [186], rely on the use of continuum electrostatics instead of statistical method. Compared to PROPKA, these Poisson–Boltzmann-based predictors grant a more detailed description of the interatomic interactions at the cost of higher computational time. In H++, the pK_a and isoelectric point information is complemented with the generation of AMBER-compatible structures and topologies (with both implicit and explicit solvent) and there is even the option to neutralize the system with counterions. Also noteworthy are the IPC 2.0 and Proteome-pI 2.0 releases. In these online servers, users can input protein sequences and get pK_a values and isolectric points based on several different models as well as query isolectric points of 20k proteomes [385, 386].

The initial (and often rate-limiting) step to study biological systems with atomistic detail is to acquire a molecular structure. A common source is the Protein Data Bank (PDB) which at the present contains almost 200k experimentally determined structures [387]. However, many of these structures correspond to the same protein and there is no structure available for most biologically occurring proteins. In the case of the human proteome, the most represented in the PDB, just 35% of proteins have a PDB entry and many of these are only partially solved [281, 388]. Homology modelling is often employed when a specific protein structure is not available, but there is a good degree of sequence similarity with other known structures. In recent years machine learning (ML) alternative models have been proposed to predict 3D structures from sequences of amino acids with remarkable accuracy [252, 280]. Most notably, AlphaFold was applied to several proteomes and the predicted structures have been made available [281, 282]. These structures, similar to experimental ones, lack hydrogen atoms and require preprocessing before they can be used in a molecular modeling pipeline.

It is clear that the same PDB structures have likely been repeatedly prepared countless times by different research groups. To save computational resources and streamline structure preparation from the PDB, we have calculated the pK_a values on these structures using our continuum electrostatics-based pK_a predictor (PypKa [1]). With these pK_a values collected in the pKPDB database [2], it became possible to efficiently retrieve structures with representative protonation states. We are currently extending the pKPDB database to the complete AlphaFold DB. For very large proteins (more than 1400 residues), which are quite computationally demanding, we have alternatively used pKAI [3] which can mimic PypKa's results with a mean average error of 0.3 pH units (see Methods).

In this work, we present the PypKa server, an online web service to calculate pK_a values of proteins and prepare biomolecular structures for usage in downstream applications. The user may select a structure by its PBD identification code or UniProtKB accession number. In the query, if a previous result of the selected structure is found in the pKPDB, no calculation is required/performed, and the cleaned pdb file can be instantly accessed. The user can also upload a custom pdb file which will trigger a new pK_a calculation. Ultra-large structures, which would be very time and resource-consuming for PypKa, can now be easily handled by pKAI. In this server, it is also possible to obtain input structures compatible with the most popular MD force fields, like AMBER, and GROMOS. In the future, we will also support CHARMM, and constant-pH MD force fields. Furthermore, these structures, the pK_a values, and the isoelectric points, can be obtained from a REST API, making them easily incorporated into our user's own pipelines and other services.

4.2.4 Materials and Methods

Implementation

Front end The PypKa server is a progressive web app powered by the React-based GatsbyJS framework. The standard web technologies HTML5, CSS3, and JavaScript were used, and most modern web browsers support the web app. The Protein Data Bank Web Services is used to fetch information about the user-selected accession codes. The AlphaFold Protein Structure Database developed by DeepMind and EMBL-EBI is accessed to download the AlphaFold generated structures by their UniProtKB identifier. The communication with the back end is done mainly via a REST API, except for the submission results retrieval, which is done with a WebSocket to facilitate the real-time data transfer from the server. The titration plots shown on the results page are rendered by the plotly.js graphing library.

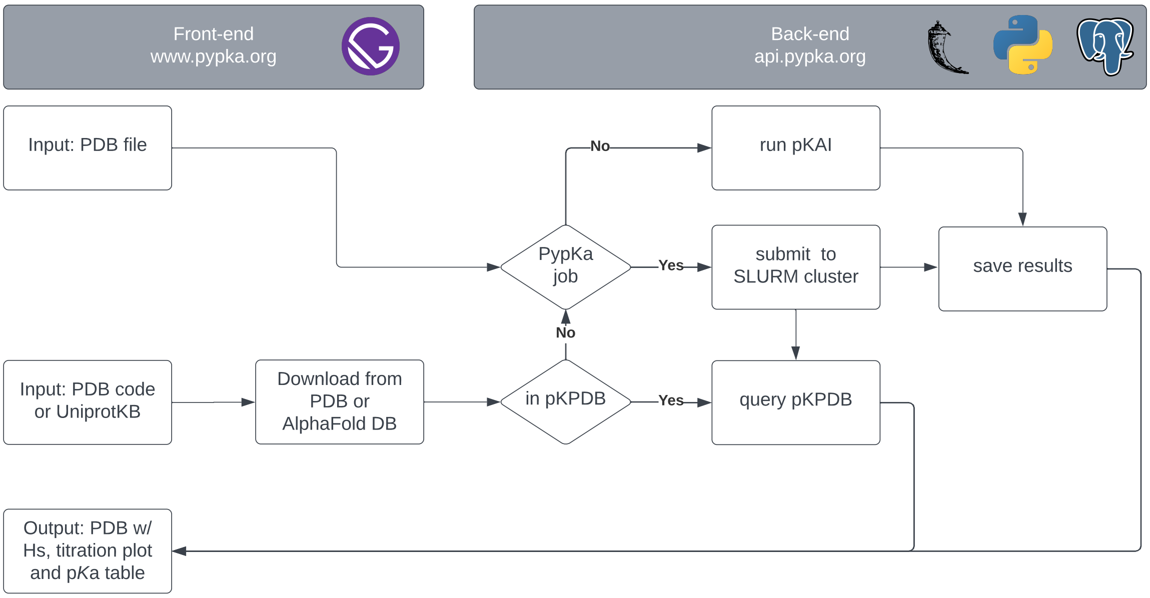


Figure 4.3: Overview of the PypKa web server workflow.

Back end The back end is hosted in a Ubuntu 18.04 instance and can be divided into several components: a REST API and WebSocket services, an HTTP server, resources and jobs manager, and a relational database. The REST API and the WebSocket services are Python3.8-based and run on top of an Nginx v1.14.0 server that handles the requests. We use Flask v2.0.1 for the API implementation and Tornado v6.1 for the WebSocket. Slurm v20.02.7 is used to manage the resources and PypKa jobs allocation efficiently. PypKa v2.7.2 [1] is pK_a predictor being used with default settings unless otherwise specified by the user. pKAI [3] is the ML model used to accelerate the pK_a calculations in large systems. The pdbmender library v0.4.1, which uses the PDB2PQR [384] to reconstruct missing atoms and deal with different naming nomenclatures, handles the input structures preprocessing and also creates the output structures with the correct protonation states, estimated by the pK_a predictors. The pKPDB [2] with precomputed pK_a values, isoelectric points and titration curves is implemented in PostgreSQL v12.9.

Workflow The PypKa server workflow is illustrated in Figure 4.3. At the front end, the user may submit a job using a pdb file, a PDB code, or a UniProtKB identifier. If a custom pdb file is uploaded, the user may select which pK_a predictor to use. By default, PypKa will be selected for structures with less than 1400 residues, and pKAI will be used for the remaining larger systems. While PypKa has shown to be highly scalable [1], it is still quite computationally expensive to run on these larger systems. Furthermore, a double-focusing procedure would need to be implemented to efficiently and accurately tackle such systems. The effectiveness of pKAI at replicating PypKa results and its remarkable efficiency [3] positions this method as an attractive alternative. If a PypKa prediction is selected, the job will be submitted to the

Table 4.2: REST API endpoints and parameters. The protein_id parameter can be any PBD identification code or UniProtKB accession number. The force field parameter ff must be one of the allowed options (amber, charmm, gromos, ambercphmd, charmmcphmd, gromoscphmd). For example, to obtain the structure of the PDB code 4LZT at pH 7.2 with the CHARMM nomenclature access: `api.pkpdb.org/pdb?protein_id=4lzt&ph=7.2&ff=charmm`

Method	Endpoind	Parameters	Description
GET	/pkas	protein_id	Gets pK_a values, titration curve and isoelectric point
GET	/pdb	protein_id, ph, ff	Gets the structure with a representative protonation state at a given pH value

Slurm cluster. After running the pK_a calculation, the results will be stored, returned to the front end, and presented to the user. If the user submits a job using a PDB or UniProtKB code, the pKPDB will be queried to check if it contains the corresponding results (true if default settings were used). If the query is successful, the results will be immediately returned to the user as there is no need to proceed with further calculations. In the opposite scenario, a new pK_a prediction will be performed using a structure downloaded from the PDB or, in the future, AlphaFold DB.

REST API It is possible to use the PypKa server without interacting with the web UI. The REST API allows users and other services to easily integrate pK_a calculations and structure preparation into their existing pipelines. At the moment, two endpoints are available to the general public (Table 4.2). In the /pkas endpoint, users can retrieve pK_a values, the isoelectric point, and the titration curve of a protein. In addition, with the /pdb, a structure with a representative protonation state will be returned in a user-specified nomenclature and pH. In both cases, the pKPDB will be queried for results. In the event of a submission made with a protein code that has not been previously computed and stored in the pKPDB, a very fast pKAI calculation will take place and its results will be returned to the user request.

4.2.5 Results

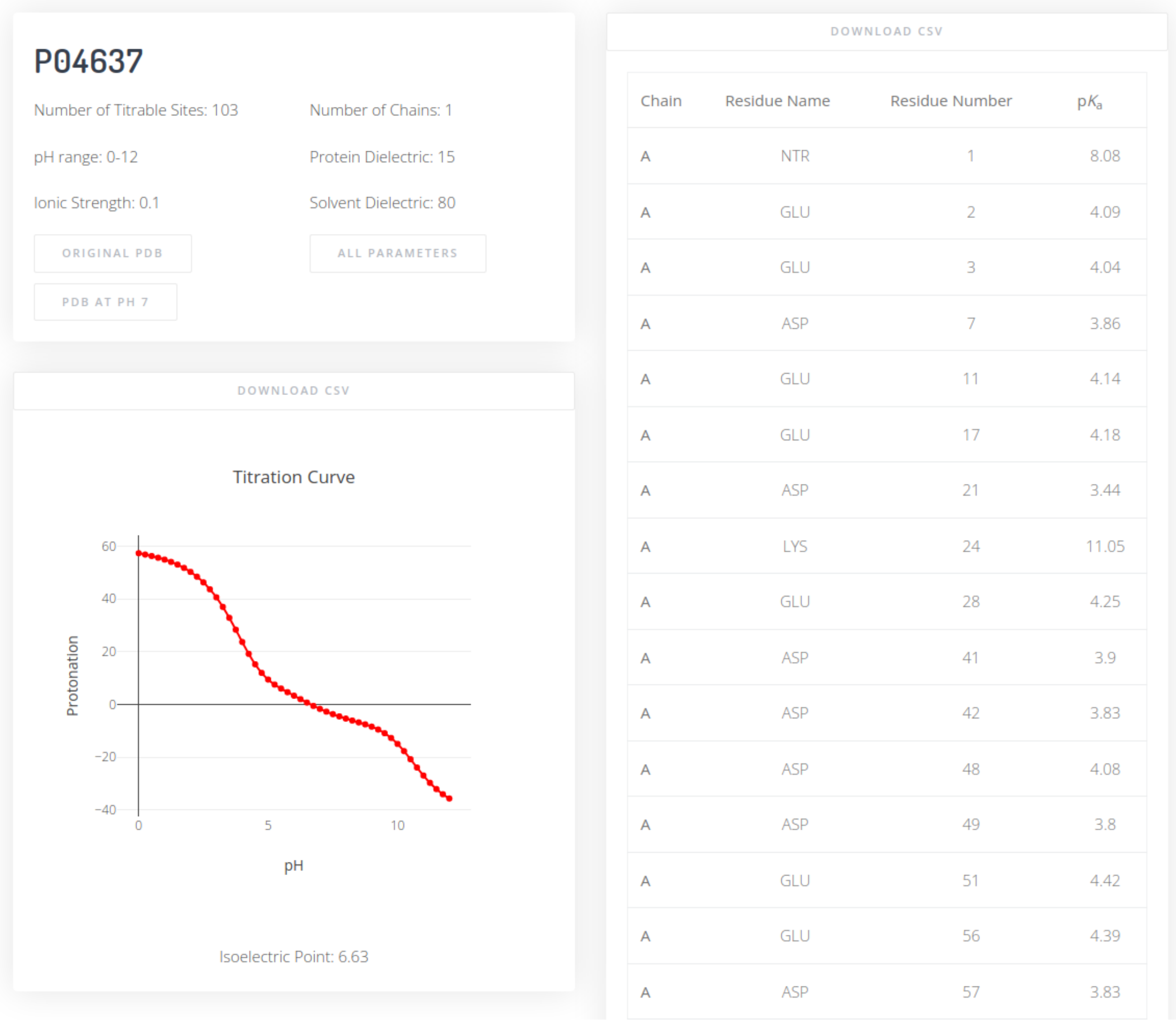
The PypKa server allows users to effortlessly run pK_a predictions and use them to prepare biomolecular structures for other molecular modeling methods. Although only proteins are titrated, nucleic acids, lipids, and ions can be included in the calculations and contribute as background charges. Three methods are offered to the user (PypKa, pKAI, and pKAI+), each with its own set of pros and cons. PypKa is the only physics-based method and the only one that can include other molecules in the calculations. It is thus the most reliable but also the most time-consuming (see Table 4.3). As expected, the computational cost increases with the size of the protein and its number of titratable residues. pKAI is an ML model trained to reproduce

Table 4.3: Typical user waiting times from job submission to final results. The reported times assume there is no latency in the communication with the server and thus do not consider the user's internet connection speed. PypKa runs are performed in parallel on 16 cores.

PDB Code	# Residues	# titratable	Time to Results (s)		
		Residues	PypKa	pKAI	pKPDB
1A1W	83	23	12s	1s	<1s
102L	163	39	18s	1s	<1s
16VP	311	65	43s	1s	<1s
15C8	430	101	1m07s	1s	<1s
1A01	574	162	2m02s	2s	<1s
1ACO	753	199	5m27s	2s	<1s
1A6D	1005	280	10m44s	3s	<1s
1A4Y	1166	308	16m08s	3s	<1s
1CB5	1359	405	24m06s	4s	<1s

PypKa predictions at a much lower computational cost. Therefore, it will be adopted when dealing with very large systems. pKAI+ is a variant of the pKAI that penalizes predictions with big pK_a shifts in an attempt to attenuate the effect of using a single (possibly unrepresentative) structure to describe the conformational ensemble of a protein. However, while pKAI+ yields a smaller error compared to pKAI or PypKa at predicting experimental pK_a values, the latter methods perform better at choosing the most representative protonation state at the physiological pH range [3]. The input pdb file for the calculations may have been experimentally solved or obtained from computational methods, such as MD or homology modeling, and the nomenclature of several popular force fields (AMBER, CHARMM, and GROMOS) is supported. The user may select a custom pdb file or an accession code from PDB or UniProtKB. In this case, and if the default parameters are used, a database with pK_a values and related quantities for 100k structures taken from the PDB will be queried and immediately returned with no extra calculations performed. If no results are found or if custom parameters are selected, a new pK_a prediction is submitted with the chosen method.

Once the results are returned to the front end, the user may visualize and download them. An example of the output page shown in the server is illustrated in Figure 4.4. The protein's isoelectric point value is shown and can also be inferred by observing the total titration curve plot. This titration curve can be downloaded as a csv file, while a similar feature is available for the table with the pK_a estimations. In this table a color code has been used to help identify residues that are markedly shifted compared to their reference pK_a values in water. A table displaying the coupled site network highlights the physical proximity and cooperativity between sites. As for downloadable output structures, the user can select between pdb and pqr file



P04637

Number of Titrable Sites: 103

Number of Chains: 1

pH range: 0-12

Protein Dielectric: 15

Ionic Strength: 0.1

Solvent Dielectric: 80

Isoelectric Point: 6.63

Chain	Residue Name	Residue Number	pK_a
A	NTR	1	8.08
A	GLU	2	4.09
A	GLU	3	4.04
A	ASP	7	3.86
A	GLU	11	4.14
A	GLU	17	4.18
A	ASP	21	3.44
A	LYS	24	11.05
A	GLU	28	4.25
A	ASP	41	3.9
A	ASP	42	3.83
A	ASP	48	4.08
A	ASP	49	3.8
A	GLU	51	4.42
A	GLU	56	4.39
A	ASP	57	3.83

Figure 4.4: Example of a PypKa server output. On the top left a small informative card displays details about the system and calculation parameters. In this card there is also a link to download the pdb file with a representative protonation state. A titration plot is shown below in which the isolectric point is highlighted. A table with the calculated pK_a values is shown on the left side. A user may download both the titration curve as well as the pK_a values as csv files.

formats with the titratable protons placed in the most likely configuration according to the pK_a calculations which include proton tautomerism. The nomenclature for the output file includes the same force field input options (AMBER, and GROMOS)

Unfortunately, there are PypKa jobs that fail, usually due to missing atoms that PDB2PQR cannot reconstruct nor repair. While pKAI jobs are able to handle most of these structures, users are strongly suggested to inspect these failed structures and fix them manually or using third-party software. Another known issue is related to the size of the input structure. The current settings of PypKa delimits its application to structures with less than ∼1400 residues, and users are recommended to use pKAI for larger systems.

4.2.6 Conclusion

Nowadays, there are several computational methods to predict pK_a values in proteins and/or to prepare biomolecular structures for molecular modeling pipelines. Some are even available as web servers for users to run these tools online. Our PypKa server stands out from these methods by providing the ability to quickly retrieve results for structures from commonly used repositories and to run extremely large systems with ML-accelerated models. Another valuable feature of this service is the REST API that allows users to integrate it into their existing scripts, protocols, as well as other web services to quickly have access to pK_a values, isoelectric points, titration curves, and pH-dependent structures.

4.2.7 Data Availability

The web server is freely available at pypka.org and does not require user registration or login. The REST API can be accessed at api.pypka.org. The code for both the front end and the back end is hosted in GitHub and can be inspected at mms-fcul/PypKa-Server-Front and mms-fcul/PypKa-Server-Back, respectively.

4.2.8 Acknowledgments

We would like to thank early users for their feedback and the Molecular Modelling and Simulation group for their suggestions. We thank Miguel Caballer for the help setting up the Slurm cluster instance at the European Grid Infrastructure (EGI). We also thank Telmo Silva for his contribution to the development of this webserver.

PR and MM acknowledge financial support from FCT: SFRH/BD/136226/2018, CEECIND/02300/2017, UIDB/04046/2020, and UIDP/04046/2020. This work benefited from services and resources provided by FCT (2021.09635.CPCA) and the EGI-ACE project (receiving funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017567), with the dedicated support from the CESGA and IN2P3-IRES resource providers.

Chapter 5

Conclusion and Future Perspectives

The main goal of this Ph.D. was to develop faster and more accessible protein pK_a predictors. This thesis presented six tools and methods that contributed to that objective. Although there is still a vast amount of features to be implemented in the current methods, the projects developed during this Ph.D. represent a significant leap in the state-of-the-art of the field. Thus, we believe the proposed goal to have been achieved successfully.

First, we released a Poisson–Boltzmann-based rigid body pK_a predictor, PypKa (Chapter 2.1). Despite the efficient trade-off between accuracy and speed exhibited, the highlight of this tool was its API that enables other applications to easily run pK_a calculations. In fact, this unique feature of PypKa has been exploited, directly or indirectly, by all of the subsequent projects developed in the context of this thesis. Besides the protein pK_a estimations, there are other useful modules within PypKa that can also be used independently by external programs, such as its preprocessing module of experimental structures and the Monte Carlo routine. As a central part of several other projects, PypKa is actively maintained, and new features are regularly added to fix reported bugs and support more use cases. In the future, we plan to, among others, have PypKa report an error estimation as well as the influence of surrounding residues on the predictions and replace DelPhi with a more portable and asymmetric grid-backed PB solver currently being developed at Walter Rocchia's lab.

One of the applications we originally envisioned to leverage Pypka was our new user-friendly CpHMD implementation, PypKa-MD (Chapter 2.2). Using PypKa to abstract the PB and MC calculations, it allowed PypKa-MD to stay lean, centered around a pdb2gmx-free topology update and focus on a better usability experience for the user. The new implementation already includes a selection of the best features from multiple forks and will serve as the base for future developments, preventing the fragmentation plaguing current versions. Currently, the top priority is attributed to implementing enhanced sampling methods such as replica exchange, umbrella sampling, and metadynamics. There is also an ongoing initiative to benchmark PypKa-

MD on a large set of proteins with experimental data while at the same time optimizing the radii of the PB calculations. Also in the works is the extension to the AMBER force field and a helpful script that automates the PME-required system neutralization with counterions.

In order to seize the opportunity to train machine learning models that could replicate our pK_a predictors and consequently gain otherwise unattainable speed ups, it was necessary to generate a substantial amount of examples. Thus, we create pKPDB (Chapter 4.1), a database of theoretical pK_a values estimated by running PypKa over thousands of experimentally resolved structures deposited in the Protein Data Bank. Even though the database contains millions of pK_a values, orders of magnitude more will be added in the future as we run PypKa on the AlphaFold DB structures. pKPDB can be downloaded as a single file, suitable for training ML models, or it can be queried in the PypKa Server (Chapter 4.2). The web app can be used to supply precomputed results virtually instantaneously or to submit PypKa jobs to the cloud. In the future, we aim to support calculations with charges and radii derived from AMBER, CHARMM, and GROMOS and also to output structures in representative protonation states with a nomenclature compatible with the same force fields.

The development of the pKPBD database facilitated the training of pKAI (Chapter 3.1), the first machine learning model trained to output pK_a values of a single conformation. While pKAI started out as a baseline model, it became quite successful at mirroring PypKa's predictions for a fraction of the computational cost (up to 1000× faster). Furthermore, a significant effort was made to ensure the accurate predictions resulted from a good understanding of the underlying physics. Considering that pKAI captures a considerable amount of information regarding the environment of a residue, we are currently exploring the latent space of its internal representation and its application to protein-related tasks with limited available experimental data like cryptic pockets and binding affinity predictions.

Much like with PypKa, the primary motivation to develop pKAI was the prospective integration into a CpHMD framework. However, the original form of pKAI was not suited for CpHMD as it struggled with nearby titratable residues, and it only predicted pK_a values from which sampling protonation states are inadequate. Hence, we generated a new data set and trained an equivariant graph neural network to solve the main issues of the previous model. Nevertheless, two milestones still need to be accomplished before the release of the new AI-accelerated CpHMD implementation, pKAI-MD (Chapter 3.2). First, the integration between the ML model and PypKa's MC module needs to be finished. Finally, it will be necessary to validate and benchmark pKAI-MD by comparing its performance against PypKa-MD.

Appendix A

Supporting Information for Chapter 2.1

Additional Methods

Structure preparation

In order for PypKa to be able to accept as input most protein PDB files, an input file preprocessing module has been included. This module addresses three main issues: correction of extra/missing atoms in PDB files; addition of proton tautomeric positions; conversion of atom and residue names to the PypKa internal naming scheme. This routine allows the use of GRO and PDB file formats with GROMOS, CHARMM and AMBER naming schemes featuring multiple chains of canonical amino acids, DNA bases, ions (chloride and sodium) and membrane (DMPC, POPC, and cholesterol) molecules. At moment, PypKa can only titrate amino acid residues, while the remaining molecules can be included to shape the electrostatic environment of the protein. To generate proton tautomeric positions, we adapted the addHtaut script included in meadTools[170, 389].

PDB2PQR [390] has been developed at Baker's Lab to automate many common tasks of preparing structures for continuum electrostatics calculations. Therefore, we have incorporated PDB2PQR to help PypKa handle the addition of missing atoms and removal solvent and cofactor atoms. It has also been extended to be able to translate the input file into the correct internal naming scheme and convert all titratable residues to their protonated states. The removal of explicit waters, cofactors and small molecules may be sometimes undesired by the user, however, there is no support for these type of molecules for the time being.

Benchmark Settings

In this work, we have used PypKa default parameters except where otherwise mentioned. PypKa uses validated partial charges and atomic radii derived from the GROMOS 54A7 force field[198] as well as proton tautomerism for all titratable groups[170]. The proteins used in this benchmark were taken from the PKAD database[95]. The subset of the full data set was created according to the following criteria: only residues in chain "A", experimental temperature between 298 K and 300 K and experimental pK_a errors below 0.1. In the benchmark against popular pK_a predictors (PropKA[149], H++[169], DelPhiPKA[304], MCCE[213]) the data set is comprised of 5 crystal structures (2LZT, 3LZT, 4LZT, 2VB1, 6RT3) of hen egg-white lysozyme (HEWL). The ionic strength was always set to match the experimental conditions and it was set to 0.1 M when this information was not disclosed. A probe with a 1.4 Å radius was used to characterize the molecular surface of proteins with an ion exclusion layer of 2.0 Å. The dielectric constant of the solvent was set to 80. To speed up the calculations the convergence threshold was set to 0.01 kT/e [99]. To solve the PB equation a two step focusing procedure is used on a grid with 91 points where the coarser grid exhibits a spacing of $\sim$ 1 Å while the distance between nodes in the finer grid is $\sim$ 0.25 Å. The relaxation parameters for the linear and nonlinear iterations were set to 0.20 and 0.75, respectively. MC runs ranged from pH 0 to

14 in 0.25 increments. For each pH value, 10^5 MC steps were performed in all titratable sites and pairs of sites with an interaction energy larger than 2 kT.

```
>>> # Iterate all sites
>>> pH = 7.0
>>> for site in tit:
>>>     resname = site.getName()
>>>     resnumb = site.getResNumber()
>>>     pK = round(site.pK, 1)
>>>     state = site.getProtState(pH)[0]
>>>     print(f'{resname:5} {resnumb:10} {pK:5.1f} {state}')
NTR            1   7.0 undefined
LYS            1  10.4 protonated
GLU            7   3.6 deprotonated
CTR          129   2.3 deprotonated

>>> import matplotlib.pyplot as plt
>>> tit_curve = tit.getTitrationCurve()
>>> n_sites = len([site for site in tit])
>>> x = sorted(list(tit_curve.keys()))
>>> y = [tit_curve[pH] * n_sites for pH in x]
>>> plt.title('Total Titration Curve')
>>> plt.xlabel('pH')
>>> plt.ylabel('Protonation')
>>> plt.plot(x, y)
>>> plt.show()
>>> plt.savefig('titration_curve.png')
```

Listing A.1: Usage example of a PypKa simulation output. The output file titration_curve.png is shown in Supplementary Figure A.1.

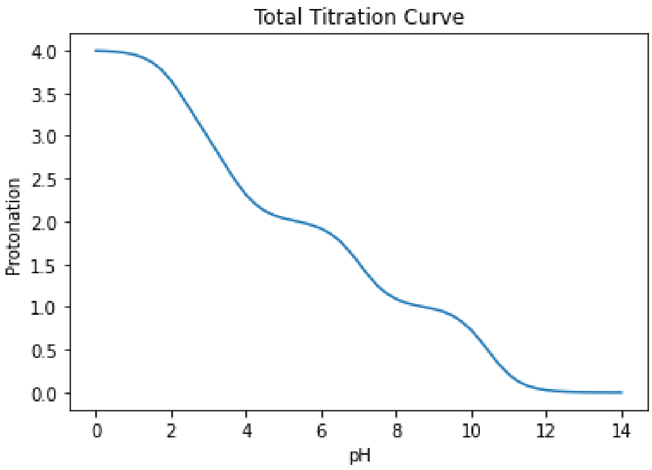


Figure A.1: Output titration curve generated with the code shown in Listing A.1

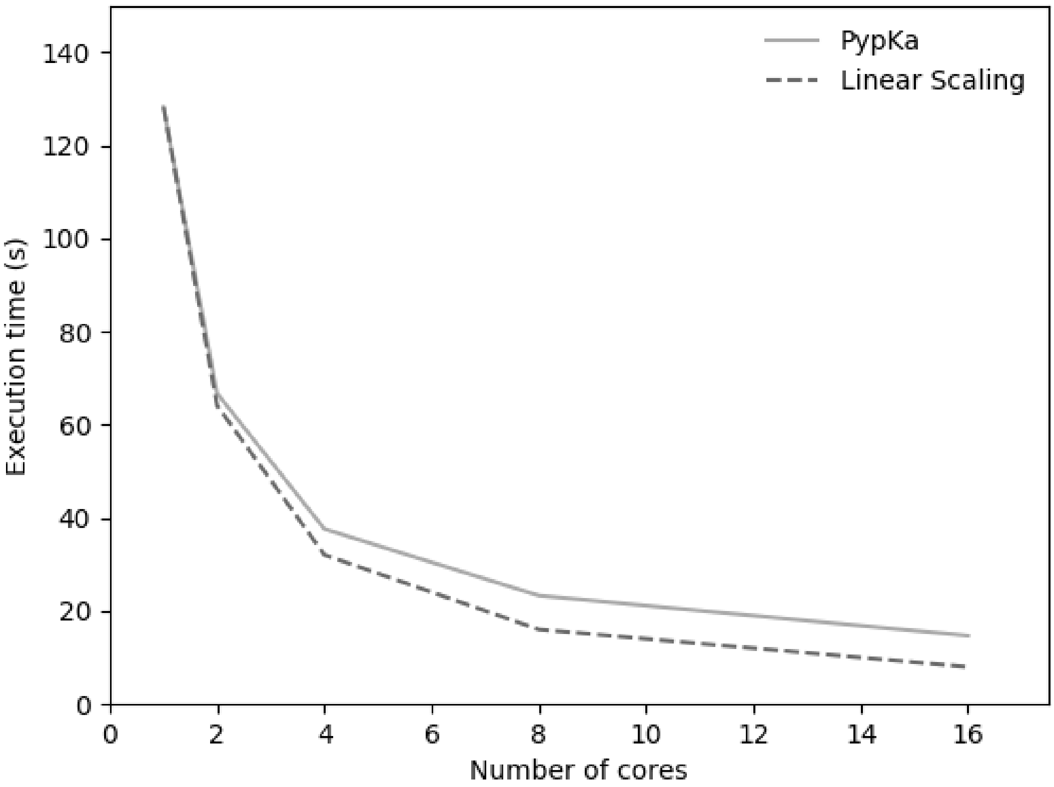


Figure A.2: PypKa runs performed on a machine with 2x octa-core Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz. The calculations were performed using 38 titrable sites (158 tautomers) of a 4LZT HEWL structure with a pH range from 0 to 15 and a pH interval of 0.25.

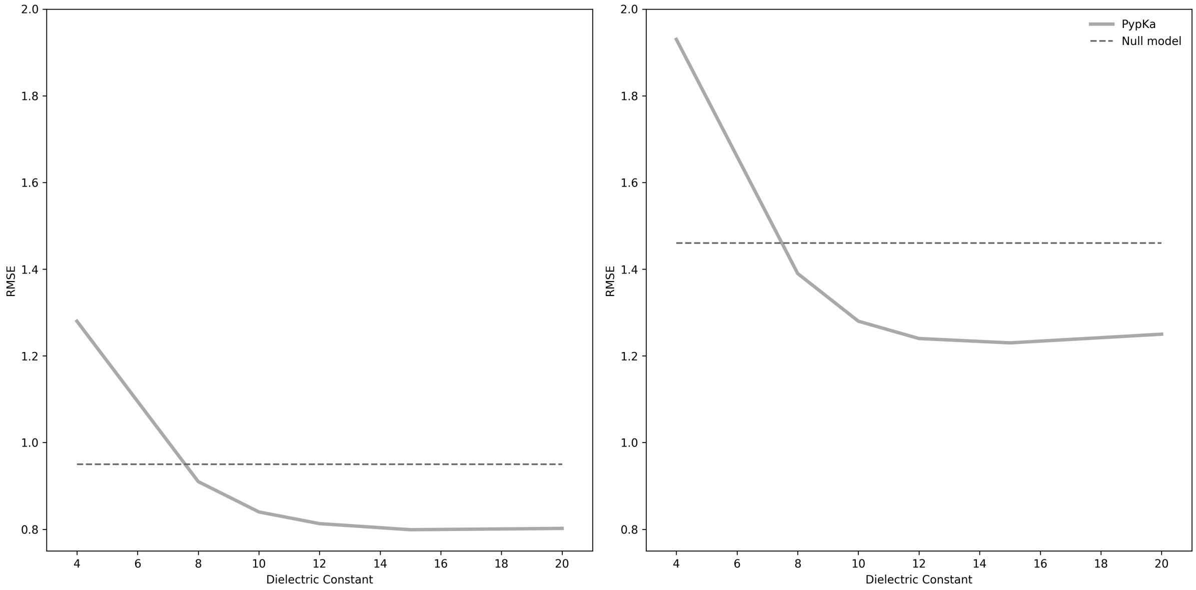


Figure A.3: RMSE of predicted pK_a values using different ε_{prot}. These values were computed using either the full set (521 residues) (A) or a subset (149 residues) with a relative SASA smaller than 50% (B). In these calculations, only the first chain of the PDB files was used for simplicity.

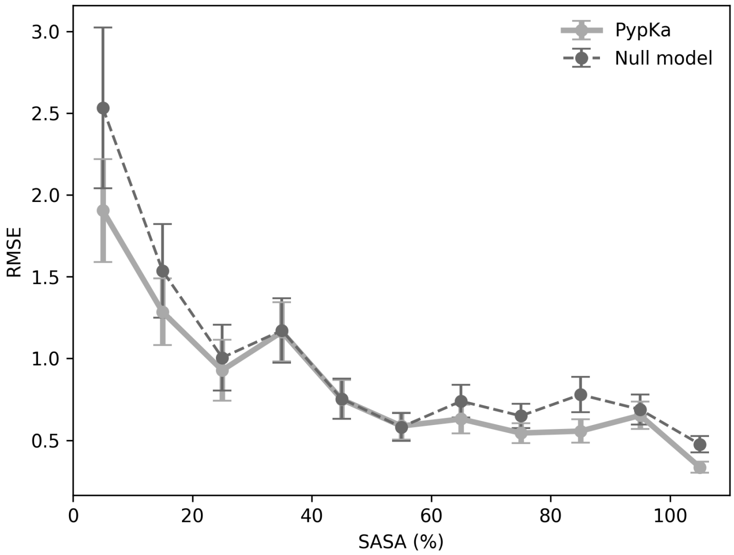


Figure A.4: RMSE values for the pK_a calculations on all 521 residues using PypKa (ε=15) grouped by SASA. The SASA values used were the ones provided by the PKAD database.[95] All values with SASA higher than 100% were collapsed into the last bin (105). The error bars show the SE of the mean of all pK_a shifts to the experimental values.

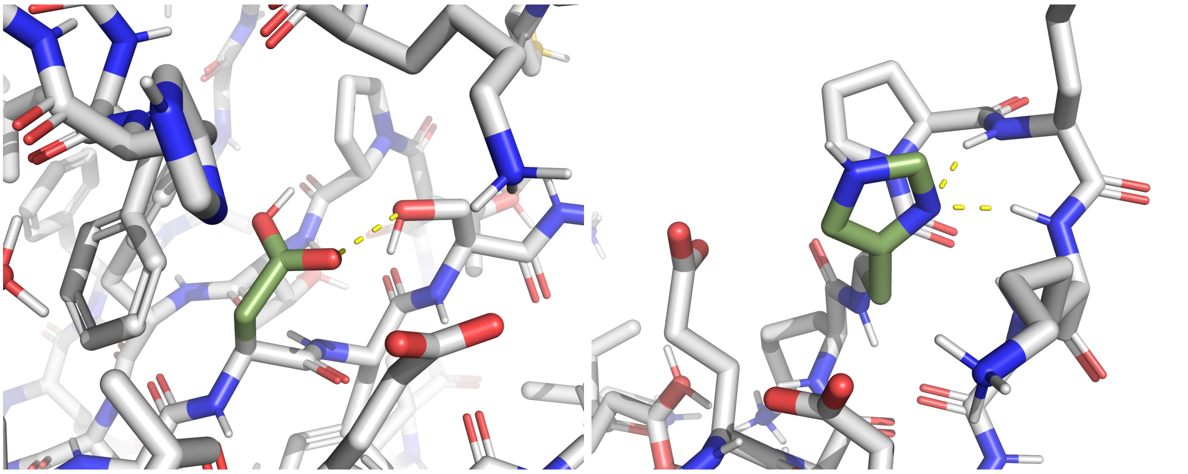


Figure A.5: Experimental structures illustrating two distinct cases identified in Figure 2.1. The electrostatic interaction maps of ASP26 in PDBID:1TRW (A) and of HIS46 of PDBID:1STN (B) are depicted. The two key residues are colored in camo green sticks.

Table A.1: Protein PDB ID codes of the full data set.

1B2X	1XNB	2BCA	1FW7	4ICB	1EX3
1BNJ	2GB1	1EPI	1GB1	4LZT	1DUK
1BNR	2IGD	1STN	1IG5	1QLP	1HV1
1DE3	2IGH	1ERU	1IGC	1PPO	1HV0
1DG9	2LZT	1PNT	1IGD	2OVO	1BCX
1DWR	2QMT	1MEK	1IGV	1YMB	1A93
1EGF	3EGF	1HNG	1EPH	3ICB	4MBN
1EPG	3GB1	2ZTA			

Table A.2: RMSE of the PypKa calculated pK_a values separated by residue type. The full data set (463 residues) was used with a dielectric constant of 15. The Null Model values were obtained from Ref. [19]

Residue	# residues	RMSE	RMSE null	MAE	MAE null	null Model pK_a
ASP	175	0.89	1.20	0.57	0.71	3.67
GLU	172	0.79	0.77	0.59	0.53	4.25
LYS	94	0.56	0.70	0.46	0.58	10.40
HIS	68	0.86	0.92	0.63	0.70	6.54
TYR	8	0.86	1.10	0.59	0.83	9.84

Appendix B

Supporting Information for Chapter 2.2

Table B.1: Fraction of Snase residues titrating at a given pH value.

pH	Titrating Residues	Fraction
1.0	1.6	0.03
2.0	7.1	0.13
3.0	7.8	0.14
4.0	12.6	0.23
5.0	11.3	0.20
6.0	8.4	0.15
7.0	3.6	0.06
8.0	2.4	0.04
9.0	19.3	0.34
10.0	22.0	0.39
11.0	22.2	0.40
12.0	17.2	0.31

Table B.2: HEWL experimental, and theoretical (L-CpHMD and PypKa-MD) pK_a values.

Residue Type	Residue Number	Experimental	L-CpHMD	PypKa-MD
N-ter	1	—	6.51 ± 0.54	6.61 ± 0.41
LYS	1	10.9	10.52 ± 0.05	10.52 ± 0.13
GLU	7	2.85	3.56 ± 0.08	3.58 ± 0.25
LYS	13	10.6	10.31 ± 0.48	10.26 ± 0.55
HIS	15	5.36	5.57 ± 0.14	5.46 ± 0.44
ASP	18	2.66	3.72 ± 0.14	3.67 ± 0.04
TYR	20	—	10.37 ± 0.18	10.51 ± 0.43
TYR	23	—	11.21 ± 0.33	11.21 ± 0.21
LYS	33	10.6	9.26 ± 0.24	8.79 ± 0.46
GLU	35	6.2	5.28 ± 0.69	5.61 ± 0.41
ASP	48	1.6	2.01 ± 0.30	1.82 ± 0.40
ASP	52	3.68	4.52 ± 0.36	4.36 ± 0.50
TYR	53	—	11.61 ± 0.08	11.70 ± 0.85
ASP	66	0.9	3.68 ± 0.24	3.53 ± 0.37
ASP	87	2.07	2.68 ± 0.45	2.42 ± 0.53
LYS	96	10.8	10.40 ± 0.50	10.36 ± 0.54
LYS	97	10.3	10.13 ± 0.13	10.13 ± 0.15
ASP	101	4.09	4.02 ± 0.51	3.95 ± 0.48
LYS	116	10.4	10.35 ± 0.30	10.36 ± 0.08
ASP	119	3.2	3.05 ± 0.30	2.78 ± 0.08
C-ter	129	2.75	3.39 ± 0.30	3.42 ± 0.64

Table B.3: Snase experimental, and theoretical (L-CpHMD and PypKa-MD) pK_a values.

Residue Type	Residue Number	Experimental	L-CpHMD	PypKa-MD
N-ter	1	—	7.06 ± 0.23	6.91 ± 0.46
LYS	5	—	10.42 ± 0.32	10.16 ± 0.15
LYS	6	—	10.30 ± 0.32	10.54 ± 0.32
HIS	8	6.52	5.57 ± 0.74	5.37 ± 0.57
LYS	9	—	11.41 ± 0.89	10.97 ± 0.79
GLU	10	2.82	4.29 ± 0.07	3.99 ± 0.32
LYS	16	—	10.59 ± 0.22	10.47 ± 0.15
ASP	19	—	2.68 ± 0.52	2.46 ± 1.16
ASP	21	—	4.00 ± 0.72	3.78 ± 0.74
LYS	24	—	9.76 ± 0.19	9.71 ± 0.16
LYS	28	—	10.79 ± 0.20	10.72 ± 0.12
ASP	40	3.87	4.10 ± 0.34	3.66 ± 0.46
GLU	43	4.32	4.02 ± 1.04	3.48 ± 1.12
LYS	45	—	10.91 ± 0.39	10.77 ± 0.34
HIS	46	5.86	3.61 ± 3.30	4.63 ± 2.14
LYS	48	—	10.66 ± 0.29	10.64 ± 0.24
LYS	49	—	10.63 ± 0.22	10.71 ± 0.06
GLU	52	3.93	4.98 ± 0.12	4.61 ± 1.20
LYS	53	—	10.95 ± 0.46	10.94 ± 0.25
TYR	54	—	11.03 ± 0.85	10.81 ± 0.36
GLU	57	3.49	4.27 ± 0.20	4.28 ± 0.90
LYS	63	—	10.27 ± 0.85	10.52 ± 0.20
LYS	64	—	10.76 ± 0.19	10.72 ± 0.27
GLU	67	3.76	4.66 ± 0.26	4.60 ± 0.10
LYS	70	—	10.78 ± 0.04	10.84 ± 0.07
LYS	71	—	10.82 ± 0.17	10.77 ± 0.12
GLU	73	3.31	4.81 ± 0.78	4.87 ± 0.52
GLU	75	3.26	5.49 ± 1.13	6.14 ± 0.98
LYS	78	—	10.26 ± 0.23	10.26 ± 0.05
ASP	83	—	2.88 ± 0.71	2.68 ± 0.36
LYS	84	—	10.63 ± 0.37	10.62 ± 0.20
TYR	85	—	9.60 ± 0.14	9.50 ± 0.21
ASP	95	2.16	3.38 ± 0.40	3.02 ± 0.53
LYS	97	—	10.21 ± 0.46	10.01 ± 0.35
GLU	101	3.81	3.53 ± 0.88	3.71 ± 0.08

LYS	110	—	11.07 ± 0.15	10.84 ± 0.37
TYR	113	—	9.90 ± 0.21	10.02 ± 0.34
TYR	115	—	10.71 ± 0.85	10.54 ± 0.43
LYS	116	—	10.45 ± 0.06	10.43 ± 0.28
HIS	121	5.3	3.43 ± 0.50	2.70 ± 0.83
GLU	122	3.89	4.73 ± 0.67	4.83 ± 0.70
HIS	124	5.73	5.16 ± 0.12	4.94 ± 0.33
LYS	127	—	10.38 ± 0.05	10.30 ± 0.22
GLU	129	3.75	4.65 ± 0.08	4.96 ± 1.08
LYS	133	—	10.81 ± 0.50	11.01 ± 0.51
LYS	134	—	10.62 ± 0.07	10.51 ± 0.16
GLU	135	3.76	4.64 ± 0.13	4.55 ± 0.10
LYS	136	—	10.91 ± 0.25	11.00 ± 0.07
GLU	142	4.49	4.50 ± 0.17	4.44 ± 0.57
ASP	143	3.8	3.22 ± 0.44	3.43 ± 0.20
ASP	146	3.86	3.20 ± 0.74	3.61 ± 0.31
C-ter	149	—	3.45 ± 0.35	3.52 ± 0.43

Table B.4: ^{h}Trx experimental, and theoretical (L-CpHMD and PypKa-MD) pK_a values.

Residue Type	Residue Number	Experimental	L-CpHMD	PypKa-MD
NTR	1	—	5.28 ± 0.14	5.98 ± 0.64
LYS	3	—	10.85 ± 0.03	11.02 ± 0.13
GLU	6	4.8	4.46 ± 0.05	4.49 ± 0.16
LYS	8	—	11.09 ± 0.15	11.03 ± 0.18
GLU	13	4.4	4.45 ± 0.23	4.42 ± 0.07
ASP	16	4	4.22 ± 0.05	4.11 ± 0.03
ASP	20	3.8	3.34 ± 0.10	3.34 ± 0.16
LYS	21	—	10.88 ± 0.01	11.01 ± 0.07
ASP	26	9.9	9.52 ± 0.64	7.94 ± 1.02
LYS	36	—	10.84 ± 0.06	10.87 ± 0.19
LYS	39	—	11.40 ± 0.67	11.63 ± 0.19
HIS	43	—	5.76 ± 0.47	5.74 ± 0.07
GLU	47	4.1	4.38 ± 0.09	4.28 ± 0.06
LYS	48	—	10.91 ± 0.07	10.99 ± 0.04
TYR	49	—	10.79 ± 0.43	10.72 ± 0.14
GLU	56	3.1	5.10 ± 0.64	4.21 ± 0.12
ASP	58	2.8	3.99 ± 0.52	5.18 ± 0.42
ASP	60	4.2	3.05 ± 0.36	2.58 ± 0.07
ASP	61	5.3	4.60 ± 0.14	4.41 ± 0.29
ASP	64	3.2	3.61 ± 0.16	3.43 ± 0.10
GLU	68	4.9	5.10 ± 0.17	5.20 ± 0.05
GLU	70	4.6	4.12 ± 0.13	4.09 ± 0.07
LYS	72	—	10.83 ± 0.13	10.66 ± 0.09
LYS	81	—	11.06 ± 0.12	11.02 ± 0.00
LYS	82	—	11.00 ± 0.11	10.87 ± 0.08
LYS	85	—	11.17 ± 0.11	11.34 ± 0.27
GLU	88	3.7	4.22 ± 0.28	3.88 ± 0.09
LYS	94	—	10.67 ± 0.10	10.55 ± 0.06
GLU	95	4.1	3.40 ± 0.07	3.18 ± 0.13
LYS	96	—	10.65 ± 0.03	10.57 ± 0.12
GLU	98	3.9	4.75 ± 0.05	4.68 ± 0.05
GLU	103	4.4	4.69 ± 0.03	4.64 ± 0.03
CTR	105	—	4.43 ± 0.08	4.24 ± 0.03

Table B.5: ^{Ec}Trx experimental, and theoretical (L-CpHMD and PypKa-MD) pK_a values.

Residue Type	Residue Number	Experimental	L-CpHMD	PypKa-MD
NTR	1	7.4	6.23 ± 0.19	6.45 ± 0.08
ASP	2	—	2.99 ± 0.27	2.95 ± 0.47
LYS	3	—	11.07 ± 0.04	11.27 ± 0.12
HIS	6	6.2	6.20 ± 0.34	5.78 ± 0.25
ASP	9	—	3.63 ± 0.67	3.58 ± 1.03
ASP	10	—	2.91 ± 0.11	2.74 ± 0.11
ASP	13	—	3.97 ± 0.37	4.32 ± 0.07
ASP	15	—	5.15 ± 0.12	5.15 ± 0.14
LYS	18	—	11.05 ± 0.13	11.02 ± 0.08
ASP	20	3.8	3.97 ± 0.02	3.93 ± 0.06
ASP	26	7.5	9.26 ± 0.27	8.68 ± 0.65
GLU	30	—	3.39 ± 0.24	3.51 ± 0.17
CYS	32	7.1	9.66 ± 0.24	10.01 ± 0.44
CYS	35	9.9	10.64 ± 0.83	10.41 ± 0.51
LYS	36	—	11.18 ± 0.20	11.02 ± 0.17
ASP	43	—	4.41 ± 0.58	4.23 ± 0.97
GLU	44	—	4.35 ± 0.06	4.28 ± 0.09
ASP	47	—	4.30 ± 0.08	4.42 ± 0.25
GLU	48	—	4.39 ± 0.03	4.31 ± 0.03
TYR	49	—	11.44 ± 0.47	11.92 ± 0.93
LYS	52	—	11.18 ± 0.17	10.95 ± 0.06
LYS	57	—	10.77 ± 0.03	10.00 ± 0.98
ASP	61	—	3.52 ± 0.17	3.80 ± 0.28
LYS	69	—	10.89 ± 0.10	10.76 ± 0.26
TYR	70	—	11.25 ± 0.58	10.93 ± 0.73
LYS	82	—	10.91 ± 0.05	10.83 ± 0.04
GLU	85	—	4.44 ± 0.03	4.39 ± 0.03
LYS	90	—	10.20 ± 0.16	10.20 ± 0.08
LYS	96	—	11.27 ± 0.20	11.37 ± 0.29
LYS	100	—	11.14 ± 0.09	11.05 ± 0.09
GLU	101	—	4.33 ± 0.02	4.26 ± 0.02
ASP	104	—	4.46 ± 0.06	4.26 ± 0.06
C-ter	108	—	3.62 ± 0.03	3.58 ± 0.13

Table B.6: Coefficient of determination (R^2), mean deviation (MD), mean average deviation (MAD), and root mean squared deviation (RMSD) between L-CpHMD and PypKa-MD pK_a predictions for each residue type.

Residue Type	# Residues	R^2	MD	MAD	RMSD
C-ter	4	0.99	0.03	0.08	0.10
ASP	32	0.94	0.12	0.28	0.42
GLU	29	0.81	0.05	0.18	0.28
HIS	7	0.76	0.10	0.39	0.51
N-ter	4	1.00	-0.22	0.29	0.38
LYS	51	0.85	0.05	0.13	0.19
CYS	2	1.00	-0.06	0.29	0.29
TYR	10	0.90	0.01	0.17	0.22
HIS*	6	0.98	0.28	0.28	0.36
ASP†	30	0.97	0.11	0.21	0.25

* Excluding Snase His-46.

† Excluding ^{h}Trx Asp-26 and Asp-58.

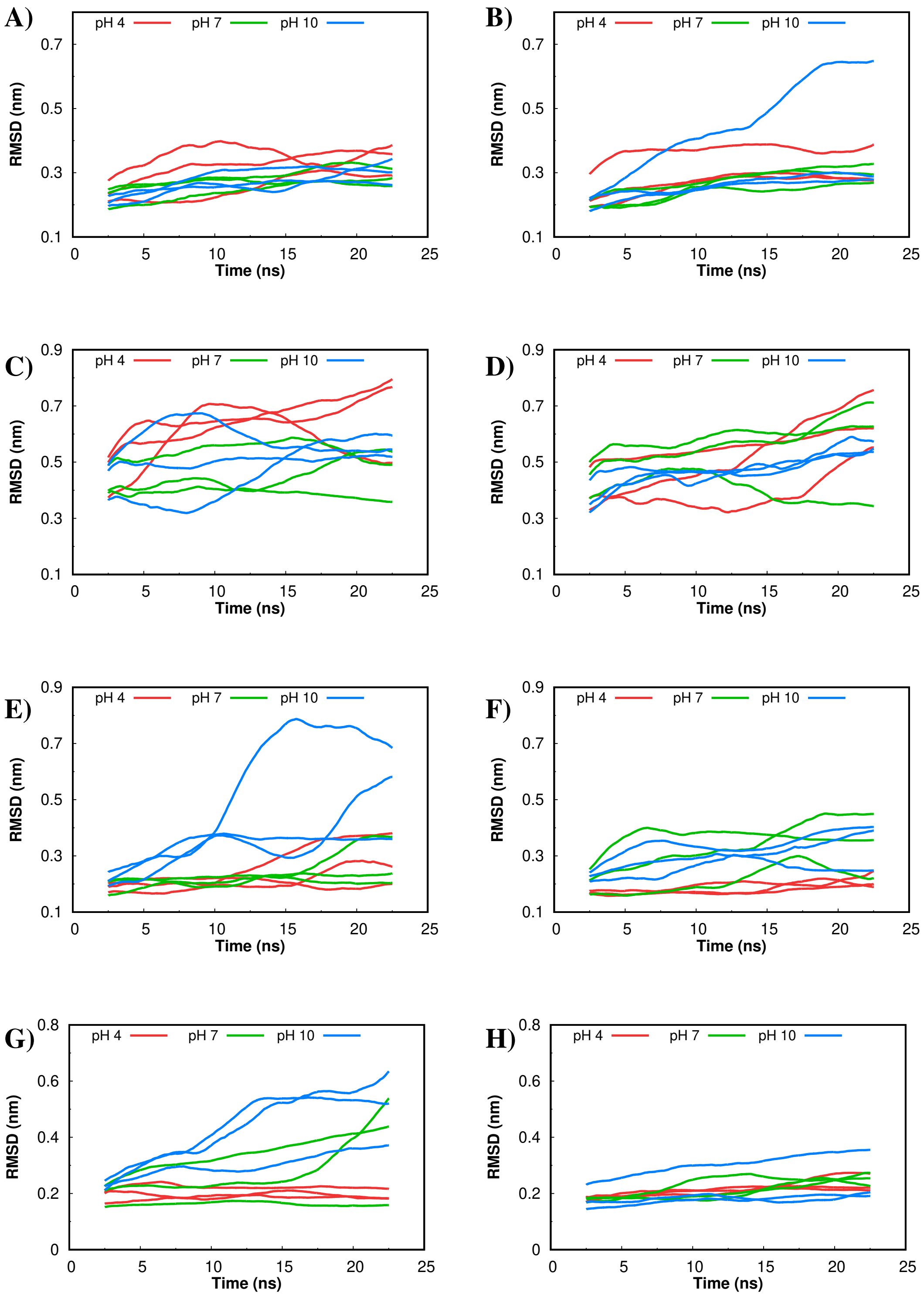


Figure B.1: Cα RMSD values over time of HEWL (A, B), Snase (C, D), ^{h}Trx (E, F), and ^{Ec}Trx (G, H) using L-CpHMD (left) and PypKa-MD (right). Triplicates of acidic (4), neutral and alkaline pH (10) simulations are represented. A sliding window average (5 ns) was applied to remove the undesired fast fluctuations.

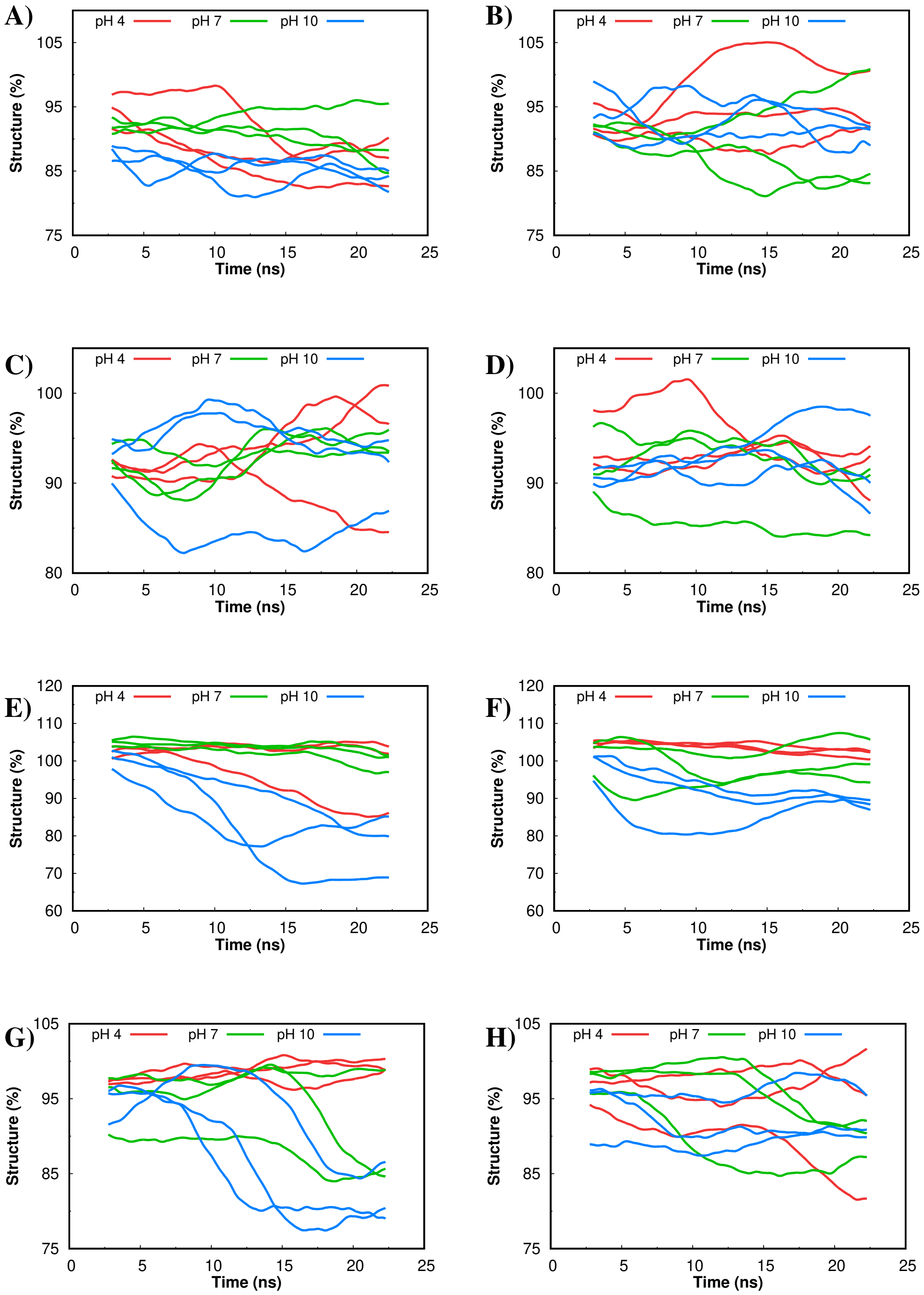


Figure B.2: Secondary structure percentage over time of HEWL (A, B), Snase (C, D), ^{h}Trx (E, F), and ^{Ec}Trx (G, H) using L-CpHMD (left) and PypKa-MD (right). Triplicates of acidic (4), neutral and alkaline pH (10) simulations are represented. A sliding window average (5 ns) was applied to remove the undesired fast fluctuations.

Appendix C

Supporting Information for Chapter 3.1

Table C.1: PDB identification codes of the proteins in experimental test set.

135l	1a2p	1a6m	1a91	1a93	1ans	1b2v	1bcx
1beo	1bhc	1bi6	1bni	1bpi	1bus	1bvi	1cdc
1d0d	1div	1duk	1dwr	1eh6	1epg	1era	1ert
1eru	1ex3	1ey0	1fks	1fna	1goa	1gs9	1h4g
1hho	1hng	1hpx	1hrc	1i0v	1igd	1igv	1jbb
1kxi	1l54	1lni	1lys	1lz1	1mbc	1nfn	1nzp
1p5f	1pnt	1poh	1ppf	1ppo	1ptd	1qh7	1qlp
1rga	1rgg	1sap	1stg	1stn	1trs	1trw	1ubq
1wla	1xnb	1ymb	1yph	1ypi	1ypt	2bca	2bus
2cpl	2hnp	2igd	2lzm	2lzt	2ovo	2sni	2tga
2trx	2zta	3ebx	3egf	3fx5	3icb	3nbs	3rn3
3srn	3ssi	4icb	4lzt	4ma9	4mbn	4pti	6gst
6lyz							

Table C.2: Performance comparison between the Null model (RMSE) and pKAI (RMSE; percentage of errors below 0.5 pH units). Information about the distribution of residue pK_a shifts (ΔpK_a) and relative solvent accessible surface area (SASA$_r$) in the test data is also shown. The Null model was calculated with ΔpK_a equal to zero.

Residue	Abundance	Null	pKAI	Error	ΔpK_a		SASA$_r$	
	(%)	RMSE	RMSE	< 0.5 (%)	Avg	Stdev	Avg	Stdev
GLU	24.9	1.42	0.44	84.7	-0.7	1.2	0.43	0.24
LYS	22.5	1.04	0.32	92.1	0.6	0.9	0.47	0.23
ASP	21.9	1.74	0.50	80.5	-1.0	1.4	0.40	0.26
TYR	13.9	3.14	0.69	67.5	2.4	2.1	0.19	0.20
HIS	9.4	1.92	0.67	73.1	-1.0	1.6	0.29	0.25
CYS	3.9	3.30	0.82	56.6	2.8	1.8	0.11	0.17
NTR	1.7	0.74	0.28	94.2	-0.3	0.7	0.75	0.27
CTR	1.8	0.88	0.35	92.5	-0.2	0.9	0.74	0.27
All	100.0	1.89 (1.24[a])	0.52 (0.31[a])	81.2	0.0	1.9	0.38	0.27

[a] Mean Absolute Error (MAE)

Table C.3: Execution time comparison between PypKa and pKAI. This benchmark was executed on a machine with a single Intel Xeon E5-2620 processor.

Protein	Number of residues / titratable	Execution Time (s)		Speedup Factor	Time per residue / titratable (s)	
		PypKa	pKAI		PypKa	pKAI
4LZT	129/21	26.5	0.8	33×	0.21/ 1.26	0.006/0.038
4K5C	341/100	92.0	1.2	76×	0.27/ 0.92	0.004/0.012
7C8J	902/249	2898.2	2.3	1260×	3.21/11.64	0.003/0.009

Table C.4: Experimental pK_a benchmark of several methods on a data set of 736 residues from 97 proteins. For each method, we report their RMSE, the mean absolute error (MAE), the 0.9 quantile, the error percentage below 0.5 pK units, and the coefficient of determination (R^2). The null model values have been taken from [19, 94].

	RMSE	MAE	Quantile 0.9	Error < 0.5 (%)	R^2
Null	1.09	0.72	1.51	52.3	0.84
PypKA	1.07	0.71	1.48	52.6	0.85
PROPKA	1.11	0.73	1.58	51.1	0.84
pKAI	1.15	0.75	1.66	49.3	0.82
pKAI+	0.98	0.64	1.37	55.0	0.87

Table C.5: Comparison between Null model and pKAI+ RMSE values. The Null model is defined as the pK_a values of the residues in water taken from Reference [19].

Residue	Abundance (%)	Null RMSE	pKAI+ RMSE	Error < 0.5 (%)	ΔpK_a		SASA$_r$	
					Avg	Stdev	Avg	Stdev
GLU	29.6	0.77	0.81	58.3	-0.5	0.9	0.45	0.24
LYS	14.4	0.74	0.68	60.4	0.3	0.6	0.55	0.21
ASP	29.2	1.30	1.08	59.5	-0.6	0.9	0.45	0.25
TYR	2.4	1.23	0.95	38.9	0.5	0.7	0.33	0.25
HIS	19.4	1.14	0.97	42.0	-0.5	1.1	0.39	0.22
CYS	1.2	3.39	3.43	0.0	-0.1	1.5	0.11	0.09
NTR	1.5	0.59	0.47	63.6	-0.3	0.8	0.74	0.20
CTR	2.2	0.41	0.56	75.0	-0.1	0.7	0.77	0.23
TOTAL	100.0	1.09	0.98	55.0	-0.4	1.0	0.46	0.25

[a] Mean Absolute Error (MAE)

Table C.6: One hot encoding classes of all atoms used.

Atom Name	Residue	Atom Classes
N	Main Chain	N
O	Main Chain	O
NE2	GLN	N_AMIDE
ND2	ASN	N_AMIDE
OE1	GLN	O_AMIDE
OD1	ASN	O_AMIDE
NE	ARG	NE_ARG
NH1/NH2	ARG	NH_ARG
NZ	LYS	NZ_LYS
N	NTR	NZ_LYS
OXT	CTR	O_COOH
OD1/OD2	ASP	O_COOH
OE1/OE2	GLU	O_COOH
OG	SER	OG_SER
OG1	THR	OG1_THR
ND1	HIS	ND1_HIS
NE2	HIS	NE2_HIS
NE1	TRP	NE1_TRP
OH	TYR	OH_TYR
SG	CYS	SG_CYS
SD	Methionine	SD_MET

Table C.7: RMSE improvement by adding a solvent exposure-related extra feature to the input layer. Different ways of measuring solvent exposure were tested: Half-sphere exposure (HSE), Coordination Number, Residue Depth, and relative solvent accessible surface area (SASA$_r$). HSE is a 2D measure and be subdivided into an upper (side chain facing, HSE^{up}) and lower sphere (backbone facing, HSE^{down}) half-spheres. Furthermore, two residues can be used as reference C$_\alpha$ (HSE$_\alpha$) and C$_\beta$ (HSE$_\beta$). Residue depth is the average distance of all residue's atoms to the molecular surface, and Residue Depth$_{C\alpha}$ is the atom depth of C$_\alpha$.

Feature	RMSE Improvement
HSE$_\alpha^{up}$	0.01
HSE$_\alpha^{down}$	0.01
HSE$_\beta^{up}$	0.01
HSE$_\beta^{down}$	0.02
Coordination Number	0.02
Residue Depth	0.01
Residue Depth$_{C\alpha}$	0.01
SASA$_r$	0.02

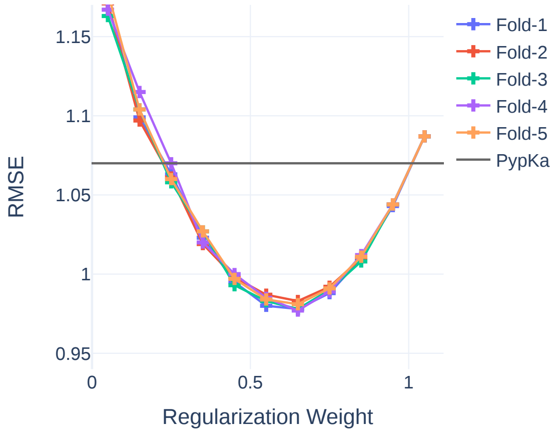


Figure C.1: Performance of pKAI+ with different regularization weights on 5 folds of the experimental test set.

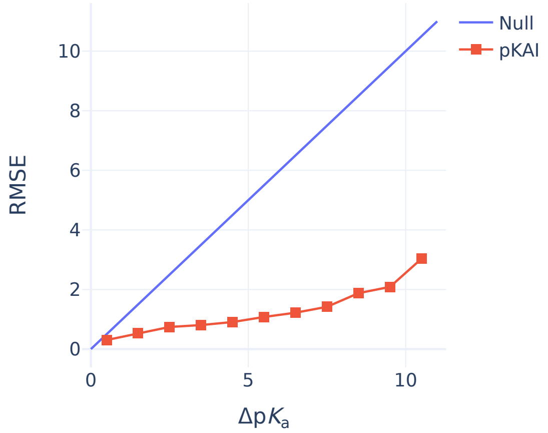


Figure C.2: RMSE variation versus the magnitude of the pK_a shift (ΔpK_a). The calculations were performed for pKAI and Null model using the PypKa predictions as reference.

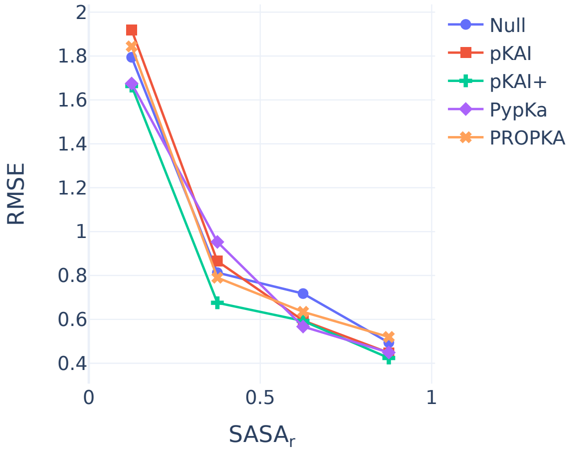


Figure C.3: pKAI+ performance at predicting experimental pK_a values dependency on the magnitude of solvent exposure (SASA) of the residues.

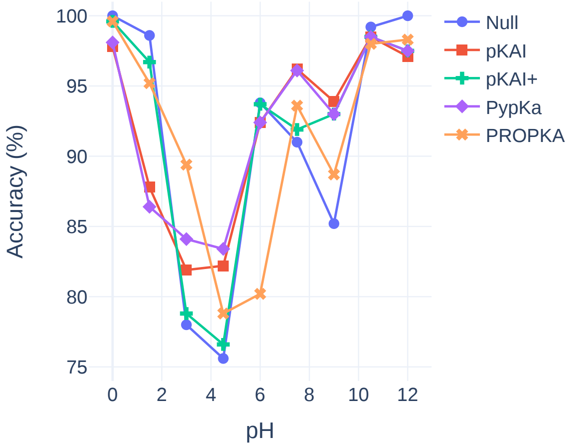


Figure C.4: Accuracy of several methods at predicting the most representative protonation states derived from experimental pK_a values.

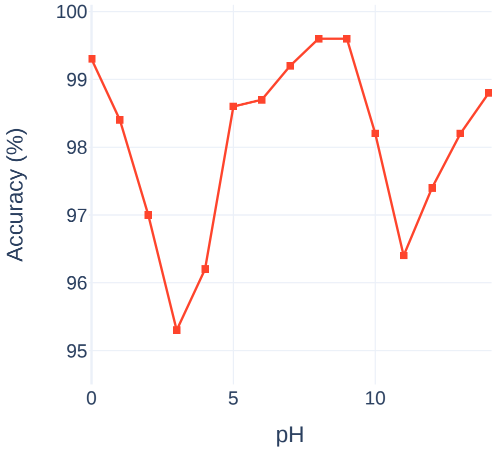


Figure C.5: pKAI accuracy at predicting PypKa-derived protonation states.

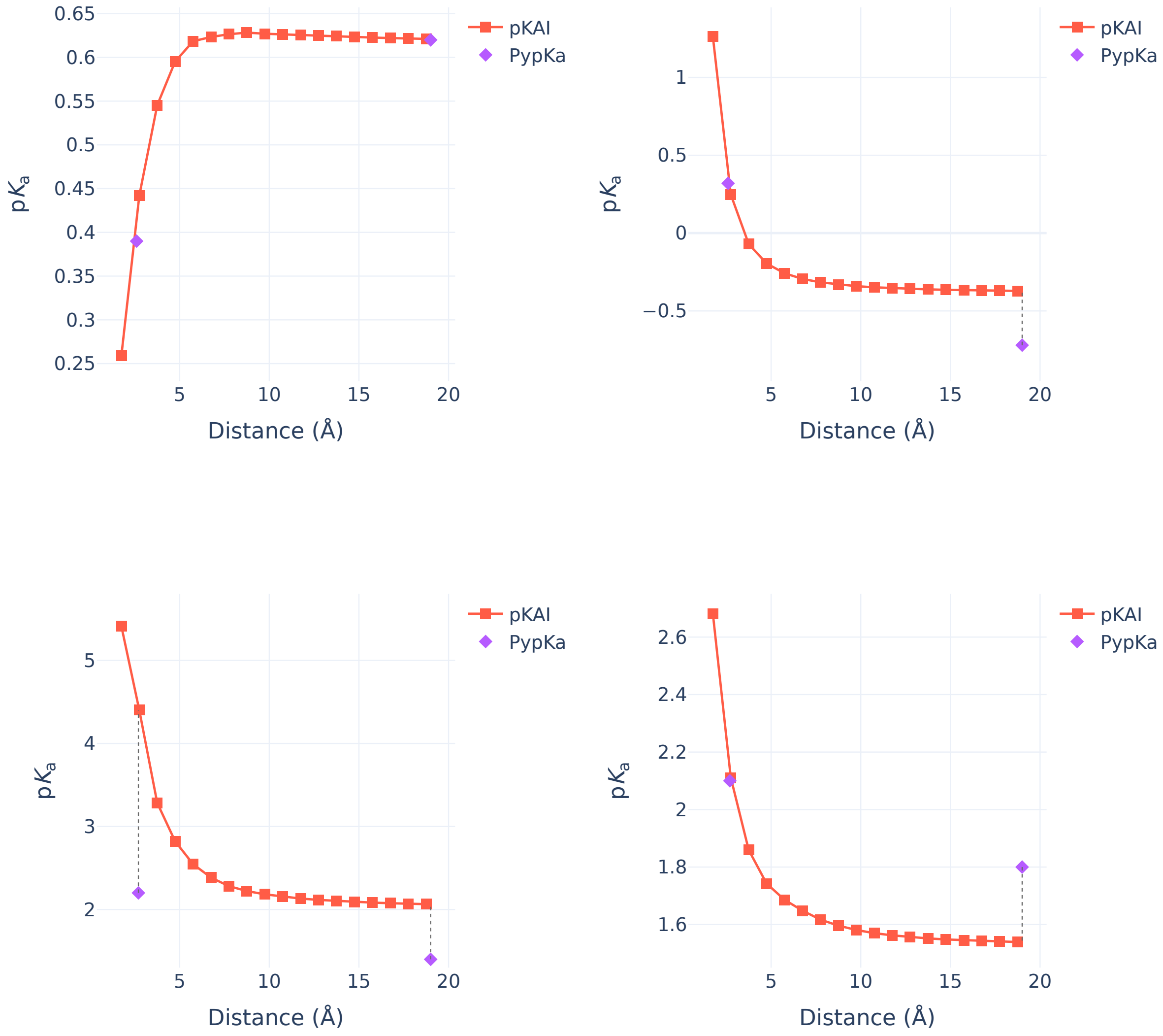


Figure C.6: Impact of changing the distance of the closest atom on pKAI's predictions for: residue GLU-154 from structure 6FT4 (A); residue LYS-118 from structure 2HRK (C); residue TYR-98 from structure 6FT4 (C); residue LYS-55 from structure 2BJU (D). For reference, we have included PypKa's predictions of the same residue in the state presented in the experimental structure and in an modified structure in which the closest atom is absent.

Appendix D

Supporting Information for Chapter 3.2

Table D.1: Summary of the individual tasks, respective resolution level and abbreviation used to train the multi-task models. The dimensionality of the output vector is also shown.

Task	Abbrev.	Level	Dim
Hydrogen probabilities	P(H)	Atom	53
ΔpK_a values	pK_a	Residue	1
Isolectric point	pI	Protein	1

Table D.2: Number of proteins and pK_a values used to train and test the GNN-based pK_a predictor.

Split	Proteins	pK_a
Train	56.8k	6.3M
Test	3.0k	322.4k

Table D.3: Number of examples of the different tasks used to train and test the multi-task GNN model.

Split	P(H)	pK_a	pI
Train	6.0M	1.8M	24.2k
Test	329.8k	101.7k	1.3k

Table D.4: Number of examples used to train and test the GNNs to predict pK_{intr} and titratable site interactions.

Split	Proteins	pK_{intr}	Interactions
Train	23.9k	5.9M	1.6B
Test	1.2k	317.9k	87.3M

Table D.5: Performance of different node features for the prediction of multiple electrostatics-related tasks. Multi-task cost and individual RMSE values shown, as well as the number of distinct node features.

	Dims	Avg	P(H)	pK_a	pI
Custom Classes	18	0.364	8.16×10^{-2}	0.481	0.400
Atom Name	25	0.374	8.34×10^{-2}	0.498	0.407
Element	5	0.434	8.98×10^{-2}	0.569	0.482

Table D.6: Impact of the inclusion of different atoms on the performance of multiple electrostatics-related tasks.

	Avg	P(H)	pK_a	pI
w/ C_α	0.364	8.16×10^{-2}	0.481	0.400
w/ C_β	0.354	8.12×10^{-2}	0.457	0.401
w/ C_α & C_β	0.367	8.48×10^{-2}	0.487	0.401
w/ C_α w/o backbone	0.414	8.83×10^{-2}	0.566	0.430

Table D.7: Impact of the inclusion of different C_β atoms on the performance of multiple electrostatics-related tasks.

	Avg	P(H)	pK_a	pI
All C_β	0.354	8.12×10^{-2}	0.457	0.401
Alifatic	0.363	8.42×10^{-2}	0.458	0.422
Titrating	0.377	8.62×10^{-2}	0.478	0.437
Other	0.376	8.38×10^{-2}	0.480	0.431

Table D.8: Performance comparison between pKAI 1.0 and pKAI 2.0 at predicting pK_a values.

	RMSE	MAE	R^2
Null	1.89	1.24	0.00
pKAI 1.0	0.52	0.31	0.93
pKAI 2.0	0.35	0.19	0.97

Table D.9: Residue RMSE comparison between pKAI 2.0, and pKAI 1.0.

	Null	pKAI 2.0	pKAI 1.0
GLU	1.42	0.25	0.44
LYS	1.04	0.22	0.32
ASP	1.74	0.28	0.50
TYR	3.14	0.52	0.69
HIS	1.92	0.47	0.67
CYS	3.30	0.68	0.82
NTR	0.74	0.46	0.28
CTR	0.88	0.26	0.35

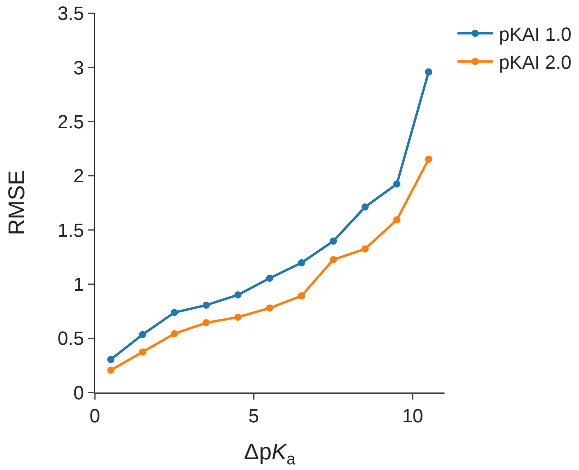


Figure D.1: pKAI 1.0 and pKAI 2.0 performance dependence on the ΔpK_a.

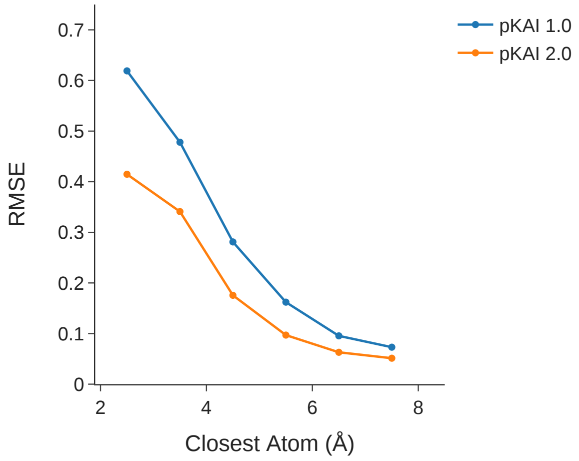


Figure D.2: pKAI 1.0 and pKAI 2.0 performance dependence on the closest interacting atom.

Appendix E

Supporting Information for Chapter 4.1

Table E.1: Summary table of the pKPDB database containing 120k structures of ∼33k distinct proteins. SASA_r stands for relative solvent accessible surface area.

Proteins	Structures	Isoelectric Points	pK_a values	SASA_r
32.8 k	120.0 k	119.0 k	12.6 M	12.5 M

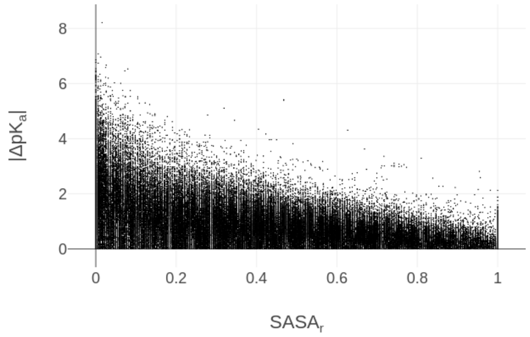


Figure E.1: Scatter plot of the relative SASA (SASA_r) versus absolute pK_a shifts (ΔpK_a). For visual clarity, only 100k points were randomly selected from the pKPDB database.

Bibliography

[1] P. B. P. S. Reis et al. “PypKa: A Flexible Python Module for Poisson–Boltzmann-Based pKa Calculations”. In: *Journal of Chemical Information and Modeling* 60.10 (2020). PMID: 32857502, pp. 4442–4448. DOI: `10.1021/acs.jcim.0c00718`.

[2] P. B. Reis, D.-A. Clevert, and M. Machuqueiro. “pKPDB: a protein data bank extension database of p Ka and pI theoretical values”. In: *Bioinformatics* 38.1 (2022), pp. 297–298.

[3] P. B. Reis et al. “A Fast and Interpretable Deep Learning Approach for Accurate Electrostatics-Driven pKa Predictions in Proteins”. In: *Journal of Chemical Theory and Computation* 0.0 (0000), null. DOI: `10.1021/acs.jctc.2c00308`.

[4] D. Vila-Viçosa et al. “A pH Replica Exchange Scheme in the Stochastic Titration Constant-pH MD Method”. In: *Journal of Chemical Theory and Computation* 15.5 (2019), pp. 3108–3116. DOI: `10.1021/acs.jctc.9b00030`.

[5] T. F. Silva et al. “The impact of using single atomistic long-range cutoff schemes with the GROMOS 54A7 force field”. In: *Journal of chemical theory and computation* 14.11 (2018), pp. 5823–5833.

[6] J. Sequeira et al. “Extending the stochastic titration CpHMD to CHARMM36m”. In: (2022). submitted.

[7] N. F. Oliveira et al. “pKa Calculations in Membrane Proteins from Molecular Dynamics Simulations”. In: *Computational Design of Membrane Proteins*. Springer, 2021, pp. 185–195.

[8] P. R. Magalhães et al. “Identification of Pan-Assay INterference compoundS (PAINS) Using an MD-Based Protocol”. In: *Computational Design of Membrane Proteins*. Springer, 2021, pp. 263–271.

[9] P. R. Magalhães et al. “Optimization of an in Silico Protocol Using Probe Permeabilities to Identify Membrane Pan-Assay Interference Compounds”. In: *Journal of Chemical Information and Modeling* ().

[10] L. Guedes et al. “Bioactivities of Centaurium erythraea (Gentianaceae) decoctions: Antioxidant activity, enzyme inhibition and docking studies”. In: *Molecules* 24.20 (2019), p. 3795.

[11] P. B. Reis et al. “Antibody-Antigen Binding Interface Analysis in the Big Data Era”. In: *Frontiers in Molecular Biosciences* (2022), p. 753.

[12] S. Sörensen. “The measurement of the hydrogen ion concentration and its importance for enzymatic process”. In: *Biochemische Zeitschrift* 21 (1909), pp. 131–304.

[13] D. L. Purich and R. D. Allison. *Handbook of biochemical kinetics: a guide to dynamic processes in the molecular life sciences*. Elsevier, 1999.

[14] K. L. Raphael et al. “Bicarbonate concentration, acid-base status, and mortality in the health, aging, and body composition study”. In: *Clinical Journal of the American Society of Nephrology* 11.2 (2016), pp. 308–316.

[15] K. Gilmour. “TRANSPORT AND EXCHANGE OF RESPIRATORY GASES IN THE BLOOD | Carbonic Anhydrase in Gas Transport and Exchange”. In: *Encyclopedia of Fish Physiology*. Ed. by A. P. Farrell. San Diego: Academic Press, 2011, pp. 899–908. ISBN: 978-0-08-092323-9. DOI: https://doi.org/10.1016/B978-0-12-374553-8.00113-1. URL: https://www.sciencedirect.com/science/article/pii/B9780123745538001131.

[16] D. L. Nelson and M. M. Cox. *Lehninger Principles of Biochemistry*. New York, New York: WH Freeman and Company, 2005, pp. 343–420.

[17] R. N. Pittman. “Regulation of Tissue Oxygenation”. In: *Colloquium Series on Integrated Systems Physiology: From Molecule to Function* 3.3 (2011), pp. 1–100. DOI: 10.4199/C00029ED1V01Y201103ISP017.

[18] M. F. Perutz et al. “The pKa values of two histidine residues in human haemoglobin, the Bohr effect, and the dipole moments of α-helices”. In: *Journal of Molecular Biology* 183.3 (1985), pp. 491–498. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(85)90016-6. URL: https://www.sciencedirect.com/science/article/pii/0022283685900166.

[19] R. L. Thurlkill et al. “*pK* values of the ionizable groups of proteins”. In: *Protein Sci.* 15.5 (2006), pp. 1214–1218.

[20] P. Bheemasenachar and J. Yeung. “Acid–base abnormalities”. In: *Core Topics in Critical Care Medicine*. Ed. by F. Gao Smith. Cambridge University Press, 2010, pp. 148–158. DOI: 10.1017/CBO9780511712289.022.

[21] C. Bohr, K. Hasselbalch, and A. Krogh. “Ueber einen in biologischer Beziehung wichtigen Einfluss, den die Kohlensäurespannung des Blutes auf dessen Sauerstoffbindung übt 1”. In: *Skandinavisches Archiv Für Physiologie* 16.2 (1904), pp. 402–412.

[22] Y. D. Nechipurenko et al. “The Role of Acidosis in the Pathogenesis of Severe Forms of COVID-19”. In: *Biology* 10.9 (2021), p. 852.

[23] D. Böning, W. M. Kuebler, and W. Bloch. “The Oxygen Dissociation Curve of Blood in COVID-19”. In: *American Journal of Physiology-Lung Cellular and Molecular Physiology* (2021).

[24] P. Born and R. Castro. “A combination of Bohr and Haldane effects provide a physiologic explanation for the increase in arterial oxygen saturation when a face mask is added to a high-flow nasal cannula in severely hypoxemic COVID-19 patients”. In: *Critical Care* 25.1 (2021), pp. 1–3.

[25] W. J. Waddell and R. G. Bates. “Intracellular pH.” In: *Physiological Reviews* 49.2 (1969), pp. 285–329.

[26] H. Hagberg. “Intracellular pH during ischemia in skeletal muscle: relationship to membrane potential, extracellular pH, tissue lactic acid and ATP”. In: *Pflügers Archiv* 404.4 (1985), pp. 342–347.

[27] J. R. Parratt, M. J. Taggart, and S. Wray. “Functional effects of intracellular pH alteration in the human uterus: simultaneous measurements of pH and force”. In: *Reproduction* 105.1 (1995), pp. 71–75. DOI: 10.1530/jrf.0.1050071. URL: https://rep.bioscientifica.com/view/journals/rep/105/1/jrf_105_1_010.xml.

[28] G. Hao, Z. P. Xu, and L. Li. “Manipulating extracellular tumour pH: an effective target for cancer therapy”. In: *RSC advances* 8.39 (2018), pp. 22182–22192.

[29] D. Coman et al. “Extracellular pH mapping of liver cancer on a clinical 3T MRI scanner”. In: *Magnetic Resonance in Medicine* 83.5 (2020), pp. 1553–1564. DOI: https://doi.org/10.1002/mrm.28035. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.28035. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.28035.

[30] M. A. Will, N. A. Clark, and J. E. Swain. “Biological pH buffers in IVF: help or hindrance to success”. In: *Journal of assisted reproduction and genetics* 28.8 (2011), pp. 711–724.

[31] M. Lehmann and F. Mimouni. “Serum Phosphate Concentration: Effect on Serum Ionized Calcium Concentration In Vitro”. In: *American Journal of Diseases of Children* 143.11 (1989), pp. 1340–1341. ISSN: 0002-922X. DOI: 10.1001/archpedi.1989.02150230098031. eprint: https://jamanetwork.com/journals/jamapediatrics/articlepdf/514859/archpedi_143_11_031.pdf. URL: https://doi.org/10.1001/archpedi.1989.02150230098031.

[32] H. Xu et al. “Phosphate Assay Kit in One Cell for Electrochemical Detection of Intracellular Phosphate Ions at Single Cells”. In: *Frontiers in Chemistry* 7 (2019), p. 360. ISSN: 2296-2646. DOI: 10.3389/fchem.2019.00360. URL: https://www.frontiersin.org/article/10.3389/fchem.2019.00360.

[33] A. Noorwali et al. “Measurement of Intracellular Inorganic Phosphate in Human Blood Red Cells, Leucocytes and Platelets”. In: *Regulation of Phosphate and Mineral Metabolism*. Ed. by S. G. Massry, J. M. Letteri, and E. Ritz. Boston, MA: Springer US, 1982, pp. 137–146. ISBN: 978-1-4684-4259-5. DOI: 10.1007/978-1-4684-4259-5_19. URL: https://doi.org/10.1007/978-1-4684-4259-5_19.

[34] D. J. Taylor et al. “Effect of insulin on intracellular pH and phosphate metabolism in human skeletal muscle in vivo”. In: *Clinical Science* 81.1 (1991), pp. 123–128. ISSN: 0143-5221. DOI: 10.1042/cs0810123. eprint: https://portlandpress.com/clinsci/article-pdf/81/1/123/463044/cs0810123.pdf. URL: https://doi.org/10.1042/cs0810123.

[35] J. J. Zimmerman, A. von Saint André-von Arnim, and J. McLaughlin. “Chapter 74 - Cellular Respiration”. In: *Pediatric Critical Care (Fourth Edition)*. Ed. by B. P. Fuhrman and J. J. Zimmerman. Fourth Edition. Saint Louis: Mosby, 2011, pp. 1058–1072. ISBN: 978-0-323-07307-3. DOI: https://doi.org/10.1016/B978-0-323-07307-3.10074-6. URL: https://www.sciencedirect.com/science/article/pii/B9780323073073100746.

[36] K. Sahlin et al. “Intracellular pH and bicarbonate concentration in human muscle during recovery from exercise”. In: *Journal of Applied Physiology* 45.3 (1978), pp. 474–480.

[37] T. Yoshida and H. Watari. “Changes in intracellular pH during repeated exercise”. In: *European journal of applied physiology and occupational physiology* 67.3 (1993), pp. 274–278.

[38] A. Melkikh and M. Sutormina. “Model of active transport of ions in cardiac cell”. In: *Journal of Theoretical Biology* 252.2 (2008), pp. 247–254. ISSN: 0022-5193. DOI: https://doi.org/10.1016/j.jtbi.2008.02.006. URL: https://www.sciencedirect.com/science/article/pii/S0022519308000568.

[39] N. Sperelakis. *Physiology and Pathophysiology of the Heart*. Vol. 90. Springer Science & Business Media, 2012.

[40] K. Izutsu. “Intracellular pH, H ion flux and H ion permeability coefficient in bullfrog toe muscle”. In: *The Journal of physiology* 221.1 (1972), pp. 15–27.

[41] P. Costa et al. “Determination of ionic permeability coefficients of the plasma membrane of Xenopus laevis oocytes under voltage clamp.” In: *The Journal of physiology* 413.1 (1989), pp. 199–211.

[42] R. W. Putnam. “Intracellular pH regulation”. In: *Cell Physiology Source Book*. Elsevier, 2001, pp. 357–372.

[43] U. Holtbäck and A. C. Aperia. “CHAPTER 18 - Paracrine Regulation of Renal Function by Dopamine”. In: *Seldin and Giebisch’s The Kidney (Fourth Edition)*. Ed. by R. J. ALPERN and S. C. HEBERT. Fourth Edition. San Diego: Academic Press, 2008, pp. 443–461. ISBN: 978-0-12-088488-9. DOI: https://doi.org/10.1016/B978-012088488-9.50021-8. URL: https://www.sciencedirect.com/science/article/pii/B9780120884889500218.

[44] J. R. Casey, S. Grinstein, and J. Orlowski. “Sensors and regulators of intracellular pH”. In: *Nature reviews Molecular cell biology* 11.1 (2010), pp. 50–61.

[45] P. S. Aronson. “Kinetic Properties of the Plasma Membrane Na+ -H+ Exchanger”. In: *Annual Review of Physiology* 47.1 (1985). PMID: 2581505, pp. 545–560. DOI: 10.

1146/annurev.ph.47.030185.002553. eprint: https://doi.org/10.1146/annurev.ph.47.030185.002553. URL: https://doi.org/10.1146/annurev.ph.47.030185.002553.

[46] J. I. Vandenberg, J. C. Metcalfe, and A. A. Grace. “Mechanisms of pHi recovery after global ischemia in the perfused heart.” In: *Circulation Research* 72.5 (1993), pp. 993–1003. DOI: 10.1161/01.RES.72.5.993. eprint: https://www.ahajournals.org/doi/pdf/10.1161/01.RES.72.5.993. URL: https://www.ahajournals.org/doi/abs/10.1161/01.RES.72.5.993.

[47] U. K. Mondal and M. A. Ilies. “Chapter 7 - Efflux pumps, NHE1, monocarboxylate transporters, and ABC transporter subfamily inhibitors”. In: *pH-Interfering Agents as Chemosensitizers in Cancer Therapy*. Ed. by C. T. Supuran and S. Carradori. Vol. 10. Cancer Sensitizing Agents for Chemotherapy. Academic Press, 2021, pp. 95–120. DOI: https://doi.org/10.1016/B978-0-12-820701-7.00017-8. URL: https://www.sciencedirect.com/science/article/pii/B9780128207017000178.

[48] W. F. Boron, L. Chen, and M. D. Parker. “Modular structure of sodium-coupled bicarbonate transporters”. In: *Journal of Experimental Biology* 212.11 (2009), pp. 1697–1706. ISSN: 0022-0949. DOI: 10.1242/jeb.028563.

[49] E. Gross and I. Kurtz. “Structural determinants and significance of regulation of electrogenic Na+-HCO 3 - cotransporter stoichiometry”. In: *American Journal of Physiology-Renal Physiology* 283.5 (2002), F876–F887. DOI: 10.1152/ajprenal.00148.2002.

[50] L. R. Gawenis et al. “Mice with a targeted disruption of the AE2 Cl-/HCO 3- exchanger are achlorhydric”. In: *Journal of Biological Chemistry* 279.29 (2004), pp. 30531–30539.

[51] L. Jiang et al. “pHi and serum regulate AE2-mediated Cl-/HCO3- exchange in CHOP cells of defined transient transfection status”. In: *American Journal of Physiology-Cell Physiology* 267.3 (1994), pp. C845–C856. DOI: 10.1152/ajpcell.1994.267.3.C845.

[52] B. Humphreys et al. “Functional characterization and regulation by pH of murine AE2 anion exchanger expressed in Xenopus oocytes”. In: *American Journal of Physiology-Cell Physiology* 267.5 (1994), pp. C1295–C1307.

[53] D. STERLING and J. R. CASEY. “Transport activity of AE3 chloride/bicarbonate anion-exchange proteins and their regulation by intracellular pH”. In: *Biochemical Journal* 344.1 (1999), pp. 221–229.

[54] S. B. Athauda et al. “A comparative study on the NH2-terminal amino acid sequences and some other properties of six isozymic forms of human pepsinogens and pepsins”. In: *The Journal of Biochemistry* 106.5 (1989), pp. 920–927.

[55] K.-U. Petersen. “Pepsin and its importance for functional dyspepsia: relic, regulator or remedy?” In: *Digestive Diseases* 36.2 (2018), pp. 98–105.

[56] N. Demaurex. “pH Homeostasis of Cellular Organelles”. In: *Physiology* 17.1 (2002), pp. 1–5. DOI: 10.1152/physiologyonline.2002.17.1.1. eprint: https://doi.org/10.1152/physiologyonline.2002.17.1.1. URL: https://doi.org/10.1152/physiologyonline.2002.17.1.1.

[57] A. M. Porcelli et al. “pH difference across the outer mitochondrial membrane measured with a green fluorescent protein mutant”. In: *Biochemical and Biophysical Research Communications* 326.4 (2005), pp. 799–804. ISSN: 0006-291X. DOI: https://doi.org/10.1016/j.bbrc.2004.11.105. URL: https://www.sciencedirect.com/science/article/pii/S0006291X04026890.

[58] M. Taniguchi and H. Yoshida. “1.37 - Unfolded Protein Response”. In: *Comprehensive Biotechnology (Third Edition)*. Ed. by M. Moo-Young. Third Edition. Oxford: Pergamon, 2011, pp. 508–520. ISBN: 978-0-444-64047-5. DOI: https://doi.org/10.1016/B978-0-444-64046-8.00033-1. URL: https://www.sciencedirect.com/science/article/pii/B9780444640468000331.

[59] O. A. Weisz. “Organelle Acidification and Disease”. In: *Traffic* 4.2 (2003), pp. 57–64. DOI: https://doi.org/10.1034/j.1600-0854.2003.40201.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1034/j.1600-0854.2003.40201.x.

[60] N. Piwon et al. “ClC-5 Cl–channel disruption impairs endocytosis in a mouse model for Dent’s disease”. In: *Nature* 408.6810 (2000), pp. 369–373.

[61] M. Hara-Chikuma et al. “Impaired acidification in early endosomes of ClC-5 deficient proximal tubule”. In: *Biochemical and biophysical research communications* 329.3 (2005), pp. 941–946.

[62] U. Kornak et al. “Loss of the ClC-7 chloride channel leads to osteopetrosis in mice and man”. In: *Cell* 104.2 (2001), pp. 205–215.

[63] D. Kasper et al. “Loss of the chloride channel ClC-7 leads to lysosomal storage disease and neurodegeneration”. In: *The EMBO journal* 24.5 (2005), pp. 1079–1091.

[64] M. Poët et al. “Lysosomal storage disease upon disruption of the neuronal chloride transport protein ClC-6”. In: *Proceedings of the National Academy of Sciences* 103.37 (2006), pp. 13854–13859.

[65] A. C. Fry and F. E. Karet. “Inherited renal acidoses”. In: *Physiology* 22.3 (2007), pp. 202–211.

[66] D. J. Colacurcio and R. A. Nixon. “Disorders of lysosomal acidification—The emerging role of v-ATPase in aging and neurodegenerative disease”. In: *Ageing research reviews* 32 (2016), pp. 75–88.

[67] D. M. Wolfe et al. “Autophagy failure in A lzheimer’s disease and the role of defective lysosomal acidification”. In: *European journal of neuroscience* 37.12 (2013), pp. 1949–1961.

[68] M. McBrayer and R. A. Nixon. “Lysosome and calcium dysregulation in Alzheimer’s disease: partners in crime”. In: *Biochemical Society Transactions* 41.6 (2013), pp. 1495–1502.

[69] Z. Breijyeh and R. Karaman. “Comprehensive review on Alzheimer’s disease: Causes and treatment”. In: *Molecules* 25.24 (2020), p. 5789.

[70] A. Sinning and C. A. Hübner. “Minireview: pH and synaptic transmission”. In: *FEBS letters* 587.13 (2013), pp. 1923–1928.

[71] B. Fang et al. “Hypothesis on the relationship between the change in intracellular pH and incidence of sporadic Alzheimer’s disease or vascular dementia”. In: *International Journal of Neuroscience* 120.9 (2010), pp. 591–595.

[72] A. M. Vincent, M. TenBroeke, and K. Maiese. “Neuronal intracellular pH directly mediates nitric oxide-induced programmed cell death”. In: *Journal of neurobiology* 40.2 (1999), pp. 171–184.

[73] G. Basurto-Islas et al. “Activation of asparaginyl endopeptidase leads to Tau hyperphosphorylation in Alzheimer disease”. In: *Journal of Biological Chemistry* 288.24 (2013), pp. 17495–17507.

[74] S. Kozlov et al. “Alzheimer’s disease: as it was in the beginning”. In: *Reviews in the Neurosciences* 28.8 (2017), pp. 825–843.

[75] L. A. Demetrius and D. K. Simon. “The inverse association of cancer and Alzheimer’s: a bioenergetic mechanism”. In: *Journal of the Royal Society Interface* 10.82 (2013), p. 20130006.

[76] R. Tabarés-Seisdedos and J. L. Rubenstein. “Inverse cancer comorbidity: a serendipitous opportunity to gain insight into CNS disorders”. In: *Nature Reviews Neuroscience* 14.4 (2013), pp. 293–304.

[77] J. Sánchez-Valle et al. “A molecular hypothesis to explain direct and inverse co-morbidities between Alzheimer’s Disease, Glioblastoma and Lung cancer”. In: *Scientific reports* 7.1 (2017), pp. 1–12.

[78] L. Schwartz et al. “Cancer and Alzheimer’s disease: intracellular pH scales the metabolic disorders”. In: *Biogerontology* 21.6 (2020), pp. 683–694.

[79] J. A. Driver et al. “Inverse association between cancer and Alzheimer’s disease: results from the Framingham Heart Study”. In: *Bmj* 344 (2012).

[80] J. D. V. Moreira et al. “The redox status of cancer cells supports mechanisms behind the Warburg effect”. In: *Metabolites* 6.4 (2016), p. 33.

[81] J. P. Blass, R. K.-F. SHEU, and G. E. Gibson. “Inherent abnormalities in energy metabolism in Alzheimer disease: interaction with cerebrovascular compromise”. In: *Annals of the New York Academy of Sciences* 903.1 (2000), pp. 204–221.

[82] A. Isidoro et al. “Breast carcinomas fulfill the Warburg hypothesis and provide metabolic markers of cancer prognosis”. In: *Carcinogenesis* 26.12 (2005), pp. 2095–2104.

[83] M. V. Liberti and J. W. Locasale. “The Warburg effect: how does it benefit cancer cells?” In: *Trends in biochemical sciences* 41.3 (2016), pp. 211–218.

[84] E. Persi et al. “Systems analysis of intracellular pH vulnerabilities for cancer therapy”. In: *Nature communications* 9.1 (2018), pp. 1–11.

[85] K. Cotter et al. “Recent Insights into the Structure, Regulation, and Function of the V-ATPases”. In: *Trends in Biochemical Sciences* 40.10 (2015), pp. 611–622. ISSN: 0968-0004. DOI: `https://doi.org/10.1016/j.tibs.2015.08.005`. URL: `https://www.sciencedirect.com/science/article/pii/S0968000415001450`.

[86] P. Swietach et al. “The chemistry, physiology and pathology of pH in cancer”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 369.1638 (2014), p. 20130099.

[87] K. Von Schwarzenberg et al. “Mode of cell death induction by pharmacological vacuolar H+-ATPase (V-ATPase) inhibition”. In: *Journal of Biological Chemistry* 288.2 (2013), pp. 1385–1396.

[88] T. Torigoe et al. “Vacuolar H+-ATPase: functional mechanisms and potential as a target for cancer chemotherapy”. In: *Anti-cancer drugs* 13.3 (2002), pp. 237–243.

[89] T. Nishisho et al. “The a3 isoform vacuolar type H+-ATPase promotes distant metastasis in the mouse B16 melanoma cells”. In: *Molecular Cancer Research* 9.7 (2011), pp. 845–855.

[90] K. Cotter et al. “Activity of plasma membrane V-ATPases is critical for the invasion of MDA-MB231 breast cancer cells”. In: *Journal of Biological Chemistry* 290.6 (2015), pp. 3680–3692.

[91] J. Capecci and M. Forgac. “The function of vacuolar ATPase (V-ATPase) a subunit isoforms in invasiveness of MCF10a and MCF10CA1a human breast cancer cells”. In: *Journal of Biological Chemistry* 288.45 (2013), pp. 32731–32741.

[92] D. R. Lide. *CRC handbook of chemistry and physics*. Vol. 85. CRC press, 2004.

[93] C. A. Fitch et al. “Arginine: Its pKa value revisited”. In: *Protein science* 24.5 (2015), pp. 752–761.

[94] G. R. Grimsley, J. M. Scholtz, and C. N. Pace. “A summary of the measured *pK* values of the ionizable groups in folded proteins”. In: *Protein Sci.* 18.1 (2009), pp. 247–251.

[95] S. Pahari, L. Sun, and E. Alexov. “PKAD: a database of experimentally measured pKa values of ionizable groups in proteins”. In: *Database* 2019 (2019).

[96] R. K. Cannan and B. C. J. G. Knight. “Dissociation Constants of Cystine, Cysteine, Thioglycollic Acid and α-Thiolactic Acid”. In: *Biochemical Journal* 21.6 (1927), p. 1384.

[97] W. Carr and W. Shutt. “Dielectric properties and ionisation constants of amino acids”. In: *Transactions of the Faraday Society* 35 (1939), pp. 579–587.

[98] T. W. Birch and L. J. Harris. “A redetermination of the titration dissociation constants of arginine and histidine with a demonstration of the zwitterion constitution of these molecules”. In: *Biochemical Journal* 24.2 (1930), p. 564.

[99] V. H. Teixeira et al. “pKa Values of Titrable Amino Acids at the Water/Membrane Interface”. In: *J. Chem. Theory Comput.* 12.3 (2016), pp. 930–934.

[100] A. Schönichen et al. “Considering protonation as a posttranslational modification regulating protein structure and function”. In: *Annual review of biophysics* 42 (2013), pp. 289–314.

[101] A. J. M. Ribeiro et al. “Mechanism and Catalytic Site Atlas (M-CSA): a database of enzyme reaction mechanisms and active sites”. In: *Nucleic acids research* 46.D1 (2018), pp. D618–D623.

[102] A. J. Ribeiro et al. “A global analysis of function and conservation of catalytic residues in enzymes”. In: *Journal of Biological Chemistry* 295.2 (2020), pp. 314–324.

[103] B. J. Pope et al. “Solution structure of human cofilin: actin binding, pH sensitivity, and relationship to actin-depolymerizing factor”. In: *Journal of Biological Chemistry* 279.6 (2004), pp. 4840–4848.

[104] J. He et al. “Molecular mechanism of membrane targeting by the GRP1 PH domain”. In: *Journal of lipid research* 49.8 (2008), pp. 1807–1815.

[105] R. Akter et al. “Islet amyloid polypeptide: structure, function, and pathophysiology”. In: *Journal of diabetes research* 2016 (2016).

[106] M. W. Van der Kamp and V. Daggett. “Influence of pH on the human prion protein: insights into the early steps of misfolding”. In: *Biophysical journal* 99.7 (2010), pp. 2289–2298.

[107] S. D. Cady et al. “Structure and function of the influenza A M2 proton channel”. In: *Biochemistry* 48.31 (2009), pp. 7356–7364.

[108] E. L. Fledderman et al. “Myristate exposure in the human immunodeficiency virus type 1 matrix protein is modulated by pH”. In: *Biochemistry* 49.44 (2010), pp. 9551–9562.

[109] V. Tsuber et al. “Mutations in cancer cause gain of cysteine, histidine, and tryptophan at the expense of a net loss of arginine on the proteome level”. In: *Biomolecules* 7.3 (2017), p. 49.

[110] S. M. Marino and V. N. Gladyshev. “Analysis and functional prediction of reactive cysteine residues”. In: *Journal of Biological Chemistry* 287.7 (2012), pp. 4419–4425.

[111] R. Huber and R. Criddle. “Comparison of the chemical properties of selenocysteine and selenocystine with their sulfur analogs”. In: *Archives of Biochemistry and Biophysics* 122.1 (1967), pp. 164–173. ISSN: 0003-9861. DOI: https://doi.org/10.1016/0003-9861(67)90136-1. URL: https://www.sciencedirect.com/science/article/pii/0003986167901361.

[112] J. Lu et al. “Penultimate selenocysteine residue replaced by cysteine in thioredoxin reductase from selenium-deficient rat liver”. In: *The FASEB Journal* 23.8 (2009), pp. 2394–2402.

[113] X.-M. Xu et al. “Targeted insertion of cysteine by decoding UGA codons with mammalian selenocysteine machinery”. In: *Proceedings of the National Academy of Sciences* 107.50 (2010), pp. 21430–21434.

[114] S. M. Marino and V. N. Gladyshev. “Cysteine Function Governs Its Conservation and Degeneration and Restricts Its Utilization on Protein Surfaces”. In: *Journal of Molecular Biology* 404.5 (2010), pp. 902–916. ISSN: 0022-2836. DOI: `https://doi.org/10.1016/j.jmb.2010.09.027`. URL: `https://www.sciencedirect.com/science/article/pii/S0022283610010156`.

[115] F. R. Blattner et al. “The complete genome sequence of Escherichia coli K-12”. In: *science* 277.5331 (1997), pp. 1453–1462.

[116] B. L. Urquhart et al. ““Proteomic contigs” of Mycobacterium tuberculosis and Mycobacterium bovis (BCG) using novel immobilised pH gradients”. In: *Electrophoresis* 18.8 (1997), pp. 1384–1392.

[117] R. A. VanBogelen et al. “Diagnosis of cellular states of microbial organisms using proteomics”. In: *ELECTROPHORESIS: An International Journal* 20.11 (1999), pp. 2149–2159.

[118] R. Schwartz, C. S. Ting, and J. King. “Whole proteome pI values correlate with subcellular localizations of proteins for organisms within the three domains of life”. In: *Genome research* 11.5 (2001), pp. 703–709.

[119] A. A. Tokmakov, A. Kurotani, and K.-I. Sato. “Protein pI and Intracellular Localization”. In: *Frontiers in Molecular Biosciences* 8 (2021).

[120] C. G. Knight et al. “Global analysis of predicted proteomes: functional adaptation of physical properties”. In: *Proceedings of the National Academy of Sciences* 101.22 (2004), pp. 8390–8395.

[121] T. Kawashima et al. “Archaeal adaptation to higher temperatures revealed by genomic sequence of Thermoplasma volcanium”. In: *Proceedings of the National Academy of Sciences* 97.26 (2000), pp. 14257–14262.

[122] J. Kiraga et al. “The relationships between the isoelectric point and: length of proteins, taxonomy and ecology of organisms”. In: *BMC genomics* 8.1 (2007), pp. 1–16.

[123] T. Arakawa and S. N. Timasheff. “Theory of protein solubility”. In: *Methods in enzymology*. Vol. 114. Elsevier, 1985, pp. 49–77.

[124] M. Pirmoradian, D. Aarsland, and R. A. Zubarev. “Isoelectric point region pI$\approx$7.4 as a treasure island of abnormal proteoforms in blood”. In: *Discoveries* 4.4 (2016).

[125] A. Kurotani et al. “Localization-specific distributions of protein pI in human proteome are governed by local pH and membrane charge”. In: *BMC Molecular and Cell Biology* 20.1 (2019), pp. 1–10.

[126] A. Hunter and H. Borsook. "The dissociation constants of arginine". In: *Biochemical Journal* 18.5 (1924), p. 883.

[127] C. L. Schmidt, P. L. Kirk, and W. Appleman. "The apparent dissociation constants of arginine and of lysine and the apparent heats of ionization of certain amino acids". In: *Journal of Biological Chemistry* 88.1 (1930), pp. 285–293.

[128] M. J. Harms et al. "Arginine residues at internal positions in a protein are always charged". In: *Proceedings of the National Academy of Sciences* 108.47 (2011), pp. 18954–18959.

[129] J. Reijenga et al. "Development of methods for the determination of pKa values". In: *Analytical chemistry insights* 8 (2013), ACI–S12304.

[130] H. G. Denham. "The electrometric determination of the hydrolysis of salts". In: *Journal of the Chemical Society, Transactions* 93 (1908), pp. 41–63.

[131] L. Di and E. H. Kerns. "Chapter 24 - pKa Methods". In: *Drug-Like Properties (Second Edition)*. Ed. by L. Di and E. H. Kerns. Second Edition. Boston: Academic Press, 2016, pp. 307–312. ISBN: 978-0-12-801076-1. DOI: https://doi.org/10.1016/B978-0-12-801076-1.00024-1. URL: https://www.sciencedirect.com/science/article/pii/B9780128010761000241.

[132] S. G. Tajc et al. "Direct determination of thiol p K a by isothermal titration microcalorimetry". In: *Journal of the American Chemical Society* 126.34 (2004), pp. 10508–10509.

[133] M. Schubert et al. "Probing electrostatic interactions along the reaction pathway of a glycoside hydrolase: histidine characterization by NMR spectroscopy". In: *Biochemistry* 46.25 (2007), pp. 7383–7395.

[134] D. K. Poon et al. "Unambiguous determination of the ionization state of a glycoside hydrolase active site lysine by ^{1}H-^{1}5N heteronuclear correlation spectroscopy". In: *Journal of the American Chemical Society* 128.48 (2006), pp. 15388–15389.

[135] M. A. Hass and F. A. Mulder. "Contemporary NMR studies of protein electrostatics". In: *Annual Review of Biophysics* 44 (2015), pp. 53–75.

[136] M. A. Hass et al. "Conformational exchange in pseudoazurin: different kinds of microsecond to millisecond dynamics characterized by their pH and buffer dependence using 15N NMR relaxation". In: *Biochemistry* 48.1 (2009), pp. 50–58.

[137] D. Farrell et al. "Titration_DB: Storage and analysis of NMR-monitored protein pH titration curves". In: *Proteins: Structure, Function, and Bioinformatics* 78.4 (2010), pp. 843–857.

[138] H. Webb et al. "Remeasuring HEWL pKa values by NMR spectroscopy: Methods, analysis, accuracy, and implications for theoretical pKa calculations". In: *Proteins Struct. Funct. Bioinf.* 79.3 (2011), pp. 685–702.

[139] T. A. van Beek. "Low-field benchtop NMR spectroscopy: status and prospects in natural product analysis†". In: *Phytochemical Analysis* 32.1 (2021), pp. 24–37. DOI: https://doi.org/10.1002/pca.2921.

[140] D. S. Wishart. “NMR metabolomics: A look ahead”. In: *Journal of Magnetic Resonance* 306 (2019), pp. 155–161. ISSN: 1090-7807. DOI: https://doi.org/10.1016/j.jmr.2019.07.013. URL: https://www.sciencedirect.com/science/article/pii/S109078071930134X.

[141] G. M. Ullmann. “Relations between protonation constants and titration curves in polyprotic acids: a critical view”. In: *The Journal of Physical Chemistry B* 107.5 (2003), pp. 1263–1271.

[142] G. M. Ullmann and E. Bombarda. “pKa values and redox potentials of proteins. What do they mean?” In: *Biological Chemistry* 394.5 (2013), pp. 611–619.

[143] A. R. Klingen, E. Bombarda, and G. M. Ullmann. “Theoretical investigation of the behavior of titratable groups in proteins”. In: *Photochemical & Photobiological Sciences* 5.6 (2006), pp. 588–596.

[144] A. Onufriev, D. A. Case, and G. M. Ullmann. “A novel view of pH titration in biomolecules”. In: *Biochemistry* 40.12 (2001), pp. 3413–3419.

[145] H. Li, A. D. Robertson, and J. H. Jensen. “The determinants of carboxyl pKa values in turkey ovomucoid third domain”. In: *Proteins: Structure, Function, and Bioinformatics* 55.3 (2004), pp. 689–704.

[146] H. Li et al. “The prediction of protein p K a’s using QM/MM: the p K a of lysine 55 in turkey ovomucoid third domain”. In: *The Journal of Physical Chemistry B* 106.13 (2002), pp. 3486–3494.

[147] N. Ghosh and Q. Cui. “p K a of Residue 66 in Staphylococal nuclease. I. Insights from QM/MM Simulations with Conventional Sampling”. In: *The Journal of Physical Chemistry B* 112.28 (2008), pp. 8387–8397.

[148] M. G. Ullmann, L. Noodleman, and D. A. Case. “Density functional calculation of pK a values and redox potentials in the bovine Rieske iron-sulfur protein”. In: *JBIC Journal of Biological Inorganic Chemistry* 7.6 (2002), pp. 632–639.

[149] M. H. Olsson et al. “PROPKA3: consistent treatment of internal and surface residues in empirical pKa predictions”. In: *J. Chem. Theory Comput.* 7.2 (2011), pp. 525–537.

[150] J. Shan and E. L. Mehler. “Calculation of pKa in proteins with the microenvironment modulated-screened coulomb potential”. In: *Proteins: Structure, Function, and Bioinformatics* 79.12 (2011), pp. 3346–3355.

[151] K. P. Tan et al. “Depth: a web server to compute depth, cavity sizes, detect potential small-molecule ligand-binding cavities and predict the pKa of ionizable residues in proteins”. In: *Nucleic Acids Research* 41.W1 (2013), W314–W321. ISSN: 0305-1048. DOI: 10.1093/nar/gkt503. eprint: https://academic.oup.com/nar/article-pdf/41/W1/W314/16944536/gkt503.pdf. URL: https://doi.org/10.1093/nar/gkt503.

[152] Y. He, J. Xu, and X.-M. Pan. “A statistical approach to the prediction of pKa values in proteins”. In: *Proteins: Structure, Function, and Bioinformatics* 69.1 (2007), pp. 75–82.

[153] J. P. Cvitkovic, C. D. Pauplis, and G. A. Kaminski. “PKA17—A Coarse-Grain Grid-Based Methodology and Web-Based Software for Predicting Protein pK a Shifts”. In: *Journal of computational chemistry* 40.18 (2019), pp. 1718–1726.

[154] G. Kieseritzky and E.-W. Knapp. “Improved pKa prediction: combining empirical and semimicroscopic methods”. In: *Journal of computational chemistry* 29.15 (2008), pp. 2575–2581.

[155] A. Y. Chen et al. “Protein pKa Prediction by Tree-Based Machine Learning”. In: *Journal of Chemical Theory and Computation* 18.4 (2022), pp. 2673–2686. DOI: `10.1021/acs.jctc.1c01257`.

[156] H. Gokcan and O. Isayev. “Prediction of protein pKa with representation learning”. In: *Chem. Sci.* 13 (8 2022), pp. 2462–2474. DOI: `10.1039/D1SC05610G`.

[157] Z. Cai et al. “Protein pKa Prediction with Machine Learning”. In: *ACS omega* 6.50 (2021), pp. 34823–34831.

[158] G. D. Hawkins, C. J. Cramer, and D. G. Truhlar. “Parametrized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium”. In: *The Journal of Physical Chemistry* 100.51 (1996), pp. 19824–19839.

[159] A. Onufriev, D. Bashford, and D. A. Case. “Modification of the generalized Born model suitable for macromolecules”. In: *The Journal of Physical Chemistry B* 104.15 (2000), pp. 3712–3720.

[160] H. Nguyen, D. R. Roe, and C. Simmerling. “Improved generalized born solvent model parameters for protein simulations”. In: *Journal of chemical theory and computation* 9.4 (2013), pp. 2020–2034.

[161] B. Aguilar, R. Shadrach, and A. V. Onufriev. “Reducing the secondary structure bias in the generalized Born model via R6 effective radii”. In: *Journal of Chemical Theory and Computation* 6.12 (2010), pp. 3613–3630.

[162] M. S. Lee, F. R. Salsbury Jr, and C. L. Brooks III. “Novel generalized Born methods”. In: *The Journal of chemical physics* 116.24 (2002), pp. 10606–10614.

[163] W. Im, M. S. Lee, and C. L. Brooks III. “Generalized born model with a simple smoothing function”. In: *Journal of computational chemistry* 24.14 (2003), pp. 1691–1702.

[164] W. Rocchia, E. Alexov, and B. Honig. “Extending the applicability of the nonlinear Poisson–Boltzmann equation: multiple dielectric constants and multivalent ions”. In: *J. Phys. Chem. B* 105.28 (2001), pp. 6507–6514.

[165] E. Jurrus et al. “Improvements to the APBS biomolecular solvation software suite”. In: *Protein Science* 27.1 (2018), pp. 112–128.

[166] D. Bashford. “An object-oriented programming suite for electrostatic effects in biological molecules An experience report on the MEAD project”. In: *International Conference on Computing in Object-Oriented Parallel Environments*. Springer. 1997, pp. 233–240.

[167] P. Beroza et al. “Protonation of interacting residues in a protein by a Monte Carlo method: application to lysozyme and the photosynthetic reaction center of Rhodobacter sphaeroides”. In: *Proc. Natl. Acad. Sci. USA* 88.13 (1991), pp. 5804–5808.

[168] S. Pahari et al. “DelPhiPKa: Including salt in the calculations and enabling polar residues to titrate”. In: *Proteins: Structure, Function, and Bioinformatics* 86.12 (2018), pp. 1277–1283.

[169] R. Anandakrishnan, B. Aguilar, and A. V. Onufriev. “H++ 3.0: automating pK prediction and the preparation of biomolecular structures for atomistic molecular modeling and simulations”. In: *Nucleic Acids Res.* 40.W1 (2012), W537–W541.

[170] A. M. Baptista and C. M. Soares. “Some Theoretical and Computational Aspects of the Inclusion of Proton Isomerism in the Protonation Equilibrium of Proteins”. In: *J. Phys. Chem. B* 105 (2001), pp. 293–309.

[171] R. E. Georgescu, E. G. Alexov, and M. R. Gunner. “Combining Conformational Flexibility and Continuum Electrostatics for Calculating pKas in Proteins”. In: *Biophys. J.* 83.4 (2002), pp. 1731–1748. ISSN: 0006-3495. DOI: `https://doi.org/10.1016/S0006-3495(02)73940-4`. URL: `http://www.sciencedirect.com/science/article/pii/S0006349502739404`.

[172] U. Khaniya et al. “Characterizing protein protonation microstates using Monte Carlo sampling”. In: *bioRxiv* (2022).

[173] B. Kuhn, P. A. Kollman, and M. Stahl. “Prediction of pKa shifts in proteins using a combination of molecular mechanical and continuum solvent calculations”. In: *Journal of computational chemistry* 25.15 (2004), pp. 1865–1872.

[174] A. Warshel and S. T. Russell. “Calculations of electrostatic interactions in biological systems and in solutions”. In: *Q. Rev. Biophys.* 17.03 (1984), pp. 283–422.

[175] I. Eberini et al. “Reorganization in apo-and holo-β-lactoglobulin upon protonation of Glu89: Molecular dynamics and pK_a calculations”. In: *Proteins Struct. Funct. Bioinf.* 54 (2004), pp. 744–758.

[176] Y. Huang, R. C. Harris, and J. Shen. “Generalized Born based continuous constant pH molecular dynamics in Amber: Implementation, benchmarking and analysis”. In: *Journal of chemical information and modeling* 58.7 (2018), pp. 1372–1383.

[177] J. Mongan, D. A. Case, and J. A. McCammon. “Constant pH molecular dynamics in generalized Born implicit solvent”. In: *J. Comput. Chem.* 25 (2004), pp. 2038–2048.

[178] A. M. Baptista, V. H. Teixeira, and C. M. Soares. “Constant-pH molecular dynamics using stochastic titration”. In: *J. Chem. Phys.* 117.9 (2002), pp. 4184–4200.

[179] J. E. Nielsen, M. Gunner, and B. Garcia-Moreno E. “The pKa Cooperative: A collaborative effort to advance structure-based calculations of pKa values and electrostatic effects in proteins”. In: *Proteins: Structure, Function, and Bioinformatics* 79.12 (2011), pp. 3249–3259.

[180] E. Alexov et al. "Progress in the prediction of pKa values in proteins". In: *Proteins: structure, function, and bioinformatics* 79.12 (2011), pp. 3260–3275.

[181] C. N. Schutz and A. Warshel. "What are the dielectric "constants" of proteins and how to validate electrostatic models?" In: *Proteins: Struct. Funct. Genet.* 44.4 (2001), pp. 400–417.

[182] M. K. Gilson. "Introduction to continuum electrostatics, with molecular applications". In: *Biophysics Textbooks online* (2000).

[183] M. Amin and J. Küpper. "Variations in Proteins Dielectric Constants". In: *ChemistryOpen* 9.6 (2020), pp. 691–694.

[184] B. N. Dominy, H. Minoux, and C. L. Brooks III. "An electrostatic basis for the stability of thermophilic proteins". In: *Proteins: Structure, Function, and Bioinformatics* 57.1 (2004), pp. 128–141.

[185] L. Li et al. "On the Dielectric "Constant" of Proteins: Smooth Dielectric Function for Macromolecular Modeling and Its Implementation in DelPhi". In: *J. Chem. Theory Comput.* 9.4 (2013). PMID: 23585741, pp. 2126–2136. DOI: `10.1021/ct400065j`. eprint: `https://doi.org/10.1021/ct400065j`. URL: `https://doi.org/10.1021/ct400065j`.

[186] L. Wang, M. Zhang, and E. Alexov. "DelPhiPKa web server: predicting pKa of proteins, RNAs and DNAs". In: *Bioinformatics* 32.4 (2015), pp. 614–615.

[187] P. Beroza and D. A. Case. "Including side chain flexibility in continuum electrostatic calculations of protein titration". In: *J. Phys. Chem.* 100.51 (1996), pp. 20156–20163.

[188] C. J. Stein, J. M. Herbert, and M. Head-Gordon. "The Poisson–Boltzmann model for implicit solvation of electrolyte solutions: Quantum chemical implementation and assessment via Sechenov coefficients". In: *The Journal of chemical physics* 151.22 (2019), p. 224111.

[189] N. Imai and T. Onishi. "Analytical Solution of Poisson-Boltzmann Equation for Two-Dimensional Many-Center Problem". In: *The Journal of Chemical Physics* 30.4 (1959), pp. 1115–1116.

[190] B. Lu et al. "Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications". In: *Commun Comput Phys* 3.5 (2008), pp. 973–1009.

[191] G. M. Ullmann and E. Bombarda. "Continuum electrostatic analysis of proteins". In: *Protein modelling* (2014), pp. 135–163.

[192] J. Kielland. "Individual activity coefficients of ions in aqueous solutions". In: *Journal of the American Chemical Society* 59.9 (1937), pp. 1675–1678.

[193] M. Schaefer, M. Sommer, and M. Karplus. "pH-dependence of protein stability: absolute electrostatic free energy differences between conformations". In: *The Journal of Physical Chemistry B* 101.9 (1997), pp. 1663–1683.

[194] E. Demchuk and R. C. Wade. "Improving the Continuum Dielectric Approach to Calculating pKas of Ionizable Groups in Proteins". In: *J. Phys. Chem.* 100.43 (1996),

pp. 17373–17387. DOI: 10.1021/jp960111d. eprint: https://doi.org/10.1021/jp960111d. URL: https://doi.org/10.1021/jp960111d.

[195] A. Jean-Charles et al. “Electrostatic contributions to solvation energies: Comparison of free energy perturbation and continuum calculations”. In: *Journal of the American Chemical Society* 113.4 (1991), pp. 1454–1455.

[196] V. Mohan et al. “Continuum model calculations of solvation free energies: accurate evaluation of electrostatic contributions”. In: *The Journal of Physical Chemistry* 96.15 (1992), pp. 6428–6431.

[197] J. P. Postma, H. J. Berendsen, and J. R. Haak. “Thermodynamics of cavity formation in water. A molecular dynamics study”. In: *Faraday Symposia of the Chemical Society*. Vol. 17. Royal Society of Chemistry. 1982, pp. 55–67.

[198] V. H. Teixeira et al. “On the use of different dielectric constants for computing individual and pairwise terms in Poisson–Boltzmann studies of protein ionization equilibrium”. In: *J. Phys. Chem. B* 109.30 (2005), pp. 14691–14706.

[199] J. M. Swanson, S. A. Adcock, and J. A. McCammon. “Optimized radii for Poisson-Boltzmann calculations with the AMBER force field”. In: *Journal of Chemical Theory and Computation* 1.3 (2005), pp. 484–493.

[200] D. Sitkoff, K. A. Sharp, and B. Honig. “Accurate calculation of hydration free energies using macroscopic solvent models”. In: *The Journal of Physical Chemistry* 98.7 (1994), pp. 1978–1988.

[201] F. Fogolari, A. Brigo, and H. Molinari. “The Poisson–Boltzmann equation for biomolecular electrostatics: a tool for structural biology”. In: *J. Mol. Recognit.* 15.6 (2002), pp. 377–392.

[202] P. Grochowski and J. Trylska. “Continuum molecular electrostatics, salt effects, and counterion binding—a review of the Poisson–Boltzmann theory and its modifications”. In: *Biopolymers: Original Research on Biomolecules* 89.2 (2008), pp. 93–113.

[203] M. Born. “Volumes and heats of hydration of ions”. In: *Z. Phys* 1.1 (1920), pp. 45–48.

[204] D. Bashford and D. A. Case. “Generalized born models of macromolecular solvation effects”. In: *Annual review of physical chemistry* 51.1 (2000), pp. 129–152.

[205] W. C. Still et al. “Semianalytical treatment of solvation for molecular mechanics and dynamics”. In: *Journal of the American Chemical Society* 112.16 (1990), pp. 6127–6129.

[206] A. V. Onufriev and D. A. Case. “Generalized Born implicit solvent models for biomolecules”. In: *Annual review of biophysics* 48 (2019), pp. 275–296.

[207] M. Schaefer and C. Froemmel. “A precise analytical method for calculating the electrostatic energy of macromolecules in aqueous solution”. In: *Journal of molecular biology* 216.4 (1990), pp. 1045–1066.

[208] A. Onufriev, D. A. Case, and D. Bashford. “Effective Born radii in the generalized Born approximation: the importance of being perfect”. In: *Journal of computational chemistry* 23.14 (2002), pp. 1297–1304.

[209] D. Bashford and M. Karplus. “pKa’s of ionizable groups in proteins: atomic detail from a continuum electrostatic model”. In: *Biochemistry* 29.44 (1990), pp. 10219–10225.

[210] A.-S. Yang et al. “On the calculation of pKas in proteins”. In: *Proteins Struct. Funct. Bioinf.* 15.3 (1993), pp. 252–265. DOI: 10.1002/prot.340150304. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.340150304. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.340150304.

[211] A. M. Baptista. “Theoretical methods for the simulation of proteins at constant pH”. PhD thesis. Lisboa: Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, 1998.

[212] N. Metropolis et al. “Equation of state calculations by fast computing machines”. In: *J. Chem. Phys.* 21.6 (1953), pp. 1087–1092.

[213] Y. Song, J. Mao, and M. Gunner. “MCCE2: improving protein pKa calculations with extensive side chain rotamer sampling”. In: *J. Comput. Chem.* 30.14 (2009), pp. 2231–2247.

[214] J. Mongan and D. A. Case. “Biomolecular simulations at constant pH”. In: *Current opinion in structural biology* 15.2 (2005), pp. 157–163.

[215] M. Machuqueiro and A. M. Baptista. “Constant-pH Molecular Dynamics with Ionic Strength Effects: Protonation–Conformation Coupling in Decalysine”. In: *J. Phys. Chem. B* 110 (2006), pp. 2927–2933.

[216] W. Chen et al. “Recent development and application of constant pH molecular dynamics”. In: *Molecular simulation* 40.10-11 (2014), pp. 830–838.

[217] F. L. Barroso daSilva and L. G. Dias. “Development of constant-pH simulation methods in implicit solvent and applications in biomolecular systems”. In: *Biophysical reviews* 9.5 (2017), pp. 699–728.

[218] A. M. Baptista, P. J. Martel, and S. B. Petersen. “Simulation of protein conformational freedom as a function of pH: constant-pH molecular dynamics using implicit titration”. In: *Proteins Struct. Funct. Bioinf.* 27.4 (1997), pp. 523–544.

[219] M. S. Lee, F. R. Salsbury, and C. L. Brooks III. “Constant-pH molecular dynamics using continuous titration coordinates”. In: *Proteins Struct. Funct. Bioinf.* 56 (2004), pp. 738–752.

[220] J. Khandogin and C. L. Brooks III. “Constant pH molecular dynamics with proton tautomerism”. In: *Biophys. J.* 89.1 (2005), pp. 141–157.

[221] X. Kong and C. L. Brooks III. “λ-dynamics: A new approach to free energy calculations”. In: *J. Chem. Phys.* 105.6 (1996), pp. 2414–2423.

[222] S. Donnini et al. “Charge-neutral constant pH molecular dynamics simulations using a parsimonious proton buffer”. In: *Journal of chemical theory and computation* 12.3 (2016), pp. 1040–1051.

[223] B. R. Brooks et al. “CHARMM: the biomolecular simulation program”. In: *Journal of computational chemistry* 30.10 (2009), pp. 1545–1614.

[224] R. C. Harris and J. Shen. “GPU-accelerated implementation of continuous constant pH molecular dynamics in amber: p K a predictions with single-pH simulations”. In: *Journal of chemical information and modeling* 59.11 (2019), pp. 4821–4832.

[225] N. Aho et al. “Scalable Constant pH Molecular Dynamics in GROMACS”. In: (2022).

[226] Y. Huang et al. *Mechanism of pH-dependent activation of the sodium-proton antiporter NhaA. Nat Commun 7: 12940*. 2016.

[227] G. B. Goh, J. L. Knight, and C. L. Brooks. “Constant pH molecular dynamics simulations of nucleic acids in explicit solvent”. In: *Journal of chemical theory and computation* 8.1 (2012), pp. 36–46.

[228] J. A. Wallace and J. K. Shen. “Charge-leveling and proper treatment of long-range electrostatics in all-atom molecular dynamics at constant pH”. In: *J. Chem. Phys.* 137.18 (2012), p. 184105.

[229] Y. Huang et al. “All-atom continuous constant pH molecular dynamics with particle mesh Ewald and titratable water”. In: *Journal of chemical theory and computation* 12.11 (2016), pp. 5411–5421.

[230] W. Chen et al. “Introducing titratable water to all-atom molecular dynamics at constant pH”. In: *Biophysical journal* 105.4 (2013), pp. L15–L17.

[231] J. Khandogin and C. L. Brooks III. “Toward the accurate first-principles prediction of ionization equilibria in proteins”. In: *Biochemistry-US* 45.31 (2006), pp. 9363–9373.

[232] J. A. Wallace and J. K. Shen. “Continuous constant pH molecular dynamics in explicit solvent with pH-based replica exchange”. In: *Journal of chemical theory and computation* 7.8 (2011), pp. 2617–2629.

[233] F. Grünewald et al. “Titratable Martini model for constant pH simulations”. In: *The Journal of chemical physics* 153.2 (2020), p. 024118.

[234] F. L. Barroso da Silva, F. Sterpone, and P. Derreumaux. “OPEP6: A new constant-pH molecular dynamics simulation scheme with OPEP coarse-grained force field”. In: *Journal of Chemical Theory and Computation* 15.6 (2019), pp. 3875–3888.

[235] D. J. Reilley et al. “Titr-DMD—A Rapid, Coarse-Grained Quasi-All-Atom Constant pH Molecular Dynamics Framework”. In: *Journal of Chemical Theory and Computation* 17.7 (2021), pp. 4538–4549.

[236] L. C. Filipe et al. “Structuring peptide dendrimers through pH modulation and substrate binding”. In: *The Journal of Physical Chemistry B* 120.38 (2016), pp. 10138–10152.

[237] P. B. Reis et al. “Role of Counterions in Constant-pH Molecular Dynamics Simulations of PAMAM Dendrimers”. In: *ACS Omega* 3.2 (2018), pp. 2001–2009.

[238] C. A. Carvalheda, S. R. R. Campos, and A. M. Baptista. “The Effect of Membrane Environment on Surfactant Protein C Stability Studied by Constant-pH Molecular Dynamics”. In: *J. Chem. Inf. Model.* (2015). DOI: 10.1021/acs.jcim.5b00076. eprint: `http://dx.doi.org/10.1021/acs.jcim.5b00076`. URL: `http://dx.doi.org/10.1021/acs.jcim.5b00076`.

[239] P. R. Magalhães et al. “Effect of a pH gradient on the protonation states of cytochrome c oxidase: A continuum electrostatics study”. In: *Journal of Chemical Information and Modeling* 57.2 (2017), pp. 256–266.

[240] H. A. Santos et al. “Constant-pH MD simulations of DMPA/DMPC lipid bilayers”. In: *J. Chem. Theory Comput.* 11.12 (2015), pp. 5973–5979.

[241] D. Vila-Viçosa et al. “Constant-pH MD simulations of an oleic acid bilayer”. In: *J. Chem. Theory Comput.* 11.5 (2015), pp. 2367–2376.

[242] N. F. Oliveira and M. Machuqueiro. “Novel US-CpHMD Protocol to Study the Protonation-Dependent Mechanism of the ATP/ADP Carrier”. In: *Journal of Chemical Information and Modeling* (2022).

[243] M. Dlugosz and J. M. Antosiewicz. “Constant-pH molecular dynamics simulations: a test case of succinic acid”. In: *Chem. Phys.* 302.1-3 (2004), 161–170. ISSN: 0301-0104. DOI: `{10.1016/j.chemphys.2004.03.031}`.

[244] M. Dlugosz, J. M. Antosiewicz, and A. D. Robertson. “Constant-pH molecular dynamics study of protonation-structure relationship in a heptapeptide derived from ovomucoid third domain”. In: *Phys. Rev. E* 69.2, Part 1 (2004). ISSN: 1539-3755. DOI: `{10.1103/PhysRevE.69.021915}`.

[245] Y. Meng and A. E. Roitberg. “Constant pH replica exchange molecular dynamics in biomolecules using a discrete protonation model”. In: *Journal of chemical theory and computation* 6.4 (2010), pp. 1401–1412.

[246] J. M. Swails, D. M. York, and A. E. Roitberg. “Constant pH replica exchange molecular dynamics in explicit solvent using discrete protonation states: implementation, testing, and validation”. In: *J. Chem. Theory Comput.* 10.3 (2014), pp. 1341–1352.

[247] S. G. Itoh, A. Damjanović, and B. R. Brooks. “pH replica-exchange method based on discrete protonation states”. In: *Proteins Struct. Funct. Bioinf.* 79.12 (2011), pp. 3420–3436.

[248] B. K. Radak et al. “Constant-pH Molecular Dynamics Simulations for Large Biomolecular Systems”. In: *J. Chem. Theory Comput.* 13.12 (2017), pp. 5933–5934.

[249] J. C. Phillips et al. “Scalable molecular dynamics with NAMD”. In: *Journal of computational chemistry* 26.16 (2005), pp. 1781–1802.

[250] G. Carleo et al. “Machine learning and the physical sciences”. In: *Reviews of Modern Physics* 91.4 (2019), p. 045002.

[251] A. L. Tarca et al. “Machine learning and its applications to biology”. In: *PLoS computational biology* 3.6 (2007), e116.

[252] J. Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), pp. 583–589. DOI: `10.1038/s41586-021-03819-2`.

[253] M. Molina and F. Garip. “Machine learning for sociology”. In: *Annual Review of Sociology* 45 (2019), pp. 27–45.

[254] M. W. Libbrecht and W. S. Noble. “Machine learning applications in genetics and genomics”. In: *Nature Reviews Genetics* 16.6 (2015), pp. 321–332.

[255] K. Chowdhary. “Natural language processing”. In: *Fundamentals of artificial intelligence* (2020), pp. 603–649.

[256] A. Voulodimos et al. “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience* 2018 (2018).

[257] A. B. Nassif et al. “Speech recognition using deep neural networks: A systematic review”. In: *IEEE access* 7 (2019), pp. 19143–19165.

[258] J. Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[259] Y. Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[260] H. Tan et al. “Vimpac: Video pre-training via masked token prediction and contrastive learning”. In: *arXiv preprint arXiv:2106.11250* (2021).

[261] Z. Tong et al. “VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training”. In: *arXiv preprint arXiv:2203.12602* (2022).

[262] A. Rives et al. “Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences”. In: *bioRxiv* (2019). DOI: `10.1101/622803`. URL: `https://www.biorxiv.org/content/10.1101/622803v4`.

[263] F. AlMahamid and K. Grolinger. “Reinforcement Learning Algorithms: An Overview and Classification”. In: *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. 2021, pp. 1–7.

[264] P. G. Polishchuk, T. I. Madzhidov, and A. Varnek. “Estimation of the size of drug-like chemical space based on GDB-17 data”. In: *Journal of computer-aided molecular design* 27.8 (2013), pp. 675–679.

[265] M. Popova, O. Isayev, and A. Tropsha. “Deep reinforcement learning for de novo drug design”. In: *Science advances* 4.7 (2018), eaap7885.

[266] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[267] Q. Wang et al. “A comprehensive survey of loss functions in machine learning”. In: *Annals of Data Science* (2020), pp. 1–26.

[268] J. Kiefer and J. Wolfowitz. “Stochastic estimation of the maximum of a regression function”. In: *The Annals of Mathematical Statistics* (1952), pp. 462–466.

[269] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[270] M. M. Bejani and M. Ghatee. “A systematic review on overfitting control in shallow and deep neural networks”. In: *Artificial Intelligence Review* 54.8 (2021), pp. 6391–6438.

[271] N. Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[272] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[273] J. Bruna et al. “Spectral networks and locally connected networks on graphs”. In: *arXiv preprint arXiv:1312.6203* (2013).

[274] V. G. Satorras, E. Hoogeboom, and M. Welling. “E (n) equivariant graph neural networks”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9323–9332.

[275] S. Kim et al. “PubChem 2019 update: improved access to chemical data”. In: *Nucleic acids research* 47.D1 (2019), pp. D1102–D1109.

[276] D. A. Benson et al. “GenBank”. In: *Nucleic acids research* 41.D1 (2012), pp. D36–D42.

[277] “UniProt: the universal protein knowledgebase in 2021”. In: *Nucleic acids research* 49.D1 (2021), pp. D480–D489.

[278] S. K. Burley et al. “RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences”. In: *Nucleic acids research* 49.D1 (2021), pp. D437–D451.

[279] J. Dunbar et al. “SAbDab: the structural antibody database”. In: *Nucleic acids research* 42.D1 (2014), pp. D1140–D1146.

[280] J. Pereira et al. “High-accuracy protein structure prediction in CASP14”. In: *Proteins: Structure, Function, and Bioinformatics* 89.12 (2021), pp. 1687–1699.

[281] K. Tunyasuvunakool et al. “Highly accurate protein structure prediction for the human proteome”. In: *Nature* 596.7873 (2021), pp. 590–596. DOI: `10.1038/s41586-021-03828-1`.

[282] M. Varadi et al. “AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models”. In: *Nucleic Acids Research* 50.D1 (2021), pp. D439–D444.

[283] A. Warshel and J. Åqvist. “Electrostatic energy and macromolecular function”. In: *Annu. Rev. Biophys. Biophys. Chem.* 20.1 (1991), pp. 267–298.

[284] P. J. Martel, A. Baptista, and S. B. Petersen. “Protein electrostatics”. In: *Biotechnol. Annu. Rev.* 2 (1996), pp. 315–372.

[285] D. Voges and A. Karshikoff. “A model of a local dielectric constant in proteins”. In: *J. Chem. Phys.* 108.5 (1998), pp. 2219–2227. DOI: `10.1063/1.475602`. eprint: `https://doi.org/10.1063/1.475602`. URL: `https://doi.org/10.1063/1.475602`.

[286] L. Wang et al. “Using DelPhi capabilities to mimic protein’s conformational reorganization with amino acid specific dielectric constants”. In: *Commun. Comput. Phys.* 13.1 (2013), pp. 13–30. ISSN: 1815-2406. DOI: 10.4208/cicp.300611.120911s. URL: https://europepmc.org/articles/PMC3966310.

[287] J. E. Nielsen and G. Vriend. “Optimizing the hydrogen-bond network in Poisson–Boltzmann equation-based pKa calculations”. In: *Proteins Struct. Funct. Bioinf.* 43.4 (2001), pp. 403–412. DOI: 10.1002/prot.1053. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.1053. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.1053.

[288] E. Alexov and M. Gunner. “Incorporating protein conformational flexibility into the calculation of pH-dependent protein properties”. In: *Biophys. J.* 72.5 (1997), pp. 2075–2093.

[289] M. Machuqueiro et al. “Membrane-induced conformational changes of kyotorphin revealed by molecular dynamics simulations”. In: *J. Phys. Chem. B* 114.35 (2010), pp. 11659–11667.

[290] V. H. Teixeira, A. S. C. Capacho, and M. Machuqueiro. “The role of electrostatics in TrxR electron transfer mechanism: A computational approach”. In: *Proteins Struct. Funct. Bioinf.* 84.12 (2016), pp. 1836–1843.

[291] R. Burgi, P. A. Kollman, and W. F. van Gunsteren. “Simulating proteins at constant pH: An approach combining molecular dynamics and Monte Carlo simulation”. In: *Proteins Struct. Funct. Bioinf.* 47.4 (2002), pp. 469–480.

[292] M. Machuqueiro and A. M. Baptista. “Molecular Dynamics Constant-pH and Reduction Potential: Application to Cytochrome c_3”. In: *J. Am. Chem. Soc.* 131 (2009), pp. 12586–12594.

[293] M. Machuqueiro and A. M. Baptista. “Is the prediction of pKa values by constant-pH molecular dynamics being hindered by inherited problems?” In: *Proteins Struct. Funct. Bioinf.* 79 (2011), pp. 3437–3447.

[294] Y. N. Vorobjev. “Potential of mean force of water-proton bath and molecular dynamic simulation of proteins at constant pH”. In: *J. Comput. Chem.* 33.8 (2012), 832–842. ISSN: 0192-8651. DOI: {10.1002/jcc.22909}.

[295] J. M. Swails and A. E. Roitberg. “Enhancing conformation and protonation state sampling of hen egg white lysozyme using pH replica exchange molecular dynamics”. In: *J. Chem. Theory Comput.* 8.11 (2012), pp. 4393–4404.

[296] J. Lee et al. “Constant pH molecular dynamics in explicit solvent with enveloping distribution sampling and Hamiltonian exchange”. In: *J. Chem. Theory Comput.* 10.7 (2014), pp. 2738–2750.

[297] P. R. Magalhães, M. Machuqueiro, and A. M. Baptista. “Constant-pH Molecular Dynamics Study of Kyotorphin in an Explicit Bilayer”. In: *Biophys. J.* 108.9 (2015), pp. 2282–2290.

[298] W. F. D. Bennett et al. "Constant pH simulations with the coarse-grained MARTINI model - Application to oleic acid aggregates". In: *Can. J. Chemistry* 91.9 (2013), pp. 839–846.

[299] G. B. Goh et al. "Constant pH molecular dynamics of proteins in explicit solvent with proton tautomerism". In: *Proteins Struct. Funct. Bioinf.* 82 (2014), pp. 1319–1331.

[300] H. A. Stern. "Molecular simulation with variable protonation states at constant pH". In: *J. Chem. Phys.* 126 (2007), p. 164112.

[301] M. Machuqueiro and A. M. Baptista. "Acidic range titration of HEWL using a constant-pH molecular dynamics method". In: *Proteins Struct. Funct. Bioinf.* 72 (2008), pp. 289–298.

[302] D. Bashford and K. Gerwert. "Electrostatic Calculations of the *pKa* Values of Ionizable Groups in Bacteriorhodopsin". In: *J. Mol. Biol.* 224 (1992), pp. 473–486.

[303] Y. Y. Sham, Z. T. Chu, and A. Warshel. "Consistent calculations of p K a's of ionizable residues in proteins: semi-microscopic and microscopic approaches". In: *J. Phys. Chem. B* 101.22 (1997), pp. 4458–4472.

[304] L. Wang, M. Zhang, and E. Alexov. "DelPhiPKa web server: predicting pKa of proteins, RNAs and DNAs". In: *Bioinformatics* 32.4 (2016), pp. 614–615.

[305] B. Kuhlman et al. "pKa values and the pH dependent stability of the N-terminal domain of L9 as probes of electrostatic interactions in the denatured state. Differentiation between local and nonlocal interactions". In: *Biochemistry* 38.15 (1999), pp. 4896–4903.

[306] C. J. Gibas and S. Subramaniam. "Explicit solvent models in protein pKa calculations". In: *Biophys. J.* 71.1 (1996), pp. 138–147.

[307] J. Antosiewicz, J. A. McCammon, and M. K. Gilson. "Prediction of pH-dependent properties of proteins". In: *J. Mol. Biol.* 238.3 (1994), pp. 415–436.

[308] H. W. van Vlijmen, M. Schaefer, and M. Karplus. "Improving the accuracy of protein pKa calculations: conformational averaging versus the average structure". In: *Proteins Struct. Funct. Bioinf.* 33.2 (1998), pp. 145–158.

[309] J. E. Mertz and B. M. Pettitt. "Molecular Dynamics at a Constant pH". In: *The International Journal of Supercomputer Applications and High Performance Computing* 8.1 (1994), pp. 47–53. DOI: 10.1177/109434209400800106. eprint: https://doi.org/10.1177/109434209400800106. URL: https://doi.org/10.1177/109434209400800106.

[310] U. Börjesson and P. H. Hünenberger. "Explicit-solvent molecular dynamics simulation at constant pH: methodology and application to small amines". In: *J. Chem. Phys.* 114 (2001), p. 9706.

[311] S. Donnini et al. "Constant pH molecular dynamics in explicit solvent with λ-dynamics". In: *J. Chem. Theory Comput.* 7.6 (2011), pp. 1962–1978.

[312] J. Henriques et al. “Charge Parametrization of the DvH-c_3 Heme Group: Validation Using Constant-(pH,*E*) Molecular Dynamics Simulations”. In: *J. Phys. Chem. B* 117 (2013), pp. 70–82.

[313] C. A. Carvalheda et al. “Structural effects of pH and deacylation on surfactant protein C in an organic solvent mixture: a constant-pH MD study”. In: *J. Chem. Inf. Model.* 53.11 (2013), pp. 2979–2989.

[314] T. Dissanayake et al. “Interpretation of pH–activity profiles for acid–base catalysis from molecular simulations”. In: *Biochemistry* 54.6 (2015), pp. 1307–1313.

[315] P. Dobrev et al. “Accurate Three States Model for Amino Acids with Two Chemically Coupled Titrating Sites in Explicit Solvent Atomistic Constant pH Simulations and pKa Calculations”. In: *J. Chem. Theory Comput.* 13.1 (2017), pp. 147–160.

[316] N. F. B. Oliveira, I. D. S. Pires, and M. Machuqueiro. “Improved GROMOS 54A7 Charge Sets for Phosphorylated Tyr, Ser, and Thr to Deal with pH-Dependent Binding Phenomena”. In: *J. Chem. Theory Comput.* 16.10 (2020), pp. 6368–6376.

[317] J. M. Antosiewicz and M. Długosz. “Constant-pH Brownian dynamics simulations of a protein near a charged surface”. In: *ACS Omega* 5.46 (2020), pp. 30282–30298.

[318] L. da Rocha, A. M. Baptista, and S. R. Campos. “Approach to Study pH-Dependent Protein Association Using Constant-pH Molecular Dynamics: Application to the Dimerization of β-Lactoglobulin”. In: *J. Chem. Theory Comput.* 18.3 (2022), pp. 1982–2001.

[319] P. Buslaev et al. “Best practices in constant pH MD simulations: accuracy and sampling”. In: *ChemRxiv* (2022). DOI: 10.26434/chemrxiv-2022-c6lg2. DOI: `10.26434/chemrxiv-2022-c6lg2`. URL: `https://doi.org/10.26434/chemrxiv-2022-c6lg2`.

[320] S. R. R. Campos, M. Machuqueiro, and A. M. Baptista. “Constant-pH molecular dynamics simulations reveal a β-rich form of the human prion protein at low pH”. In: *J. Phys. Chem. B* 114 (2010), pp. 12692–12700.

[321] M. Machuqueiro and A. M. Baptista. “The pH-Dependent Conformational States of Kyotorphin: A Constant-pH Molecular Dynamics Study”. In: *Biophys. J.* 92 (2007), pp. 1836–1845.

[322] D. Vila-Viçosa et al. “Conformational study of GSH and GSSG using constant-pH molecular dynamics simulations”. In: *J. Phys. Chem. B* 117.25 (2013), pp. 7507–7517.

[323] S. R. Campos, O. Iranzo, and A. M. Baptista. “Constant-pH MD simulations portray the protonation and structural behavior of four decapeptides designed to coordinate Cu2+”. In: *The Journal of Physical Chemistry B* 120.6 (2016), pp. 1080–1091.

[324] D. Vila-Viçosa et al. “Membrane-Induced p K a Shifts in wt-pHLIP and Its L16H Variant”. In: *Journal of chemical theory and computation* 14.6 (2018), pp. 3289–3297.

[325] T. F. Silva, D. Vila-Viçosa, and M. Machuqueiro. “Improved Protocol to Tackle the pH Effects on Membrane-Inserting Peptides”. In: *Journal of Chemical Theory and Computation* 17.7 (2021), pp. 3830–3840.

[326] D. Lousa et al. “Effect of pH on the influenza fusion peptide properties unveiled by constant-pH molecular dynamics simulations combined with experiment”. In: *Scientific reports* 10.1 (2020), pp. 1–18.

[327] N. Schmid et al. “Definition and testing of the GROMOS force-field versions 54A7 and 54B7”. In: *Eur. Biophys. J.* 40.7 (2011), p. 843.

[328] W. Huang, Z. Lin, and W. F. van Gunsteren. “Validation of the GROMOS 54A7 Force Field with Respect to β-Peptide Folding”. In: *J. Chem. Theory Comput.* 7.5 (2011), pp. 1237–1243.

[329] W. Rocchia et al. “Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects”. In: *J. Comput. Chem.* 23 (2002), pp. 128–137.

[330] V. H. Teixeira et al. “Protonation of DMPC in a Bilayer Environment Using a Linear Response Approximation”. In: *J. Chem. Theory Comput.* 10 (2014), pp. 2176–2184.

[331] M. J. Abraham et al. “GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers”. In: *SoftwareX* 1 (2015), pp. 19–25.

[332] P. Bauer, B. Hess, and E. Lindahl. *GROMACS 2022.2 Manual*. Version 2022.2. 2022. DOI: 10.5281/zenodo.6637572. URL: https://doi.org/10.5281/zenodo.6637572.

[333] S. L. Williams, C. A. F. De Oliveira, and J. A. McCammon. “Coupling constant pH molecular dynamics with accelerated molecular dynamics”. In: *J. Chem. Theory Comput.* 6.2 (2010), pp. 560–568.

[334] S. Kuramitsu and K. Hamaguchi. “Analysis of the acid-base titration curve of hen lysozyme”. In: *J. Biochem.* 87.4 (1980), pp. 1215–1219.

[335] K. Bartik, C. Redfield, and C. M. Dobson. “Measurement of the individual pKa values of acidic residues of hen and turkey lysozymes by two-dimensional 1H NMR”. In: *Biophys. J.* 66.4 (1994), pp. 1180–1184.

[336] C. A. Castaneda et al. “Molecular determinants of the *pKa* values of Asp and Glu residues in staphylococcal nuclease”. In: *Proteins Struct. Funct. Bioinf.* 77.3 (2009), pp. 570–588.

[337] I. G. Tironi et al. “A generalized reaction field method for molecular dynamics simulations”. In: *J. Chem. Phys.* 102 (1995), pp. 5451–5459.

[338] B. Hess. “P-LINCS: A Parallel Linear Constraint Solver for Molecular Simulation”. In: *J. Chem. Theory Comput.* 4 (2008), pp. 116–122.

[339] S. Miyamoto and P. A. Kollman. “SETTLE: An analytical version of the SHAKE and RATTLE algorithm for rigid water models”. In: *J. Comput. Chem.* 13 (1992), pp. 952–962.

[340] G. Bussi, D. Donadio, and M. Parrinello. “Canonical sampling through velocity rescaling”. In: *J. Chem. Phys.* 126 (2007), p. 014101.

[341] M. Parrinello and A. Rahman. “Polymorphic transitions in single crystals: A new molecular dynamics method”. In: *J. Appl. Phys.* 52.12 (1981), pp. 7182–7190. ISSN: 0021-8979. DOI: 10.1063/1.328693.

[342] M. K. Gilson, K. A. Sharp, and B. Honig. “Calculating the Eletrostatic Potential of Molecules in Solution: Method and Error Assessment”. In: *J. Comput. Chem.* 9 (1987), pp. 327–335.

[343] M. A. Walsh et al. “Refinement of triclinic hen egg-white lysozyme at atomic resolution”. In: *Acta Crystallographica Section D: Biological Crystallography* 54.4 (1998), pp. 522–546.

[344] T. R. Hynes and R. O. Fox. “The crystal structure of staphylococcal nuclease refined at 1.7 Å resolution”. In: *Proteins: Structure, Function, and Bioinformatics* 10.2 (1991), pp. 92–105.

[345] J. Qin, G. M. Clore, and A. M. Gronenborn. “The high-resolution three-dimensional solution structures of the oxidized and reduced states of human thioredoxin”. In: *Structure* 2.6 (1994), pp. 503–522.

[346] S. K. Katti, D. M. LeMaster, and H. Eklund. “Crystal structure of thioredoxin from Escherichia coli at 1.68 Å resolution”. In: *Journal of molecular biology* 212.1 (1990), pp. 167–184.

[347] J. Kim, J. Mao, and M. Gunner. “Are acidic and basic groups in buried proteins predicted to be ionized?” In: *J. Mol. Biol.* 348.5 (2005), pp. 1283–1298.

[348] D. Vila-Viçosa et al. “Reversibility of prion misfolding: insights from constant-pH molecular dynamics simulations”. In: *J. Phys. Chem. B* 116.30 (2012), pp. 8812–8821.

[349] B. H. Morrow, P. H. Koenig, and J. K. Shen. “Atomistic simulations of pH-dependent self-assembly of micelle and bilayer from fatty acids”. In: *J. Chem. Phys.* 137 (2012), pp. 194902–194902.

[350] J. M. Swails et al. “pH-Dependent Mechanism of Nitric Oxide Release in Nitrophorins 2 and 4”. In: *J. Phys. Chem. B* 113.4 (2009), pp. 1192–1201. DOI: 10.1021/jp806906x.

[351] C. L. Stanton and K. N. Houk. “Benchmarking pKa Prediction Methods for Residues in Proteins”. In: *Journal of Chemical Theory and Computation* 4.6 (2008). PMID: 26621236, pp. 951–966. DOI: 10.1021/ct8000014.

[352] A. C. Lee and G. M. Crippen. “Predicting pKa”. In: *J. Chem. Inf. Model.* 49.9 (2009). PMID: 19702243, pp. 2013–2033. DOI: 10.1021/ci900209w.

[353] M. Mirdita, M. Steinegger, and J. Söding. “MMseqs2 desktop and local web server app for fast, interactive sequence searches”. In: *Bioinformatics* 35.16 (2019), pp. 2856–2858. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty1057.

[354] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.

neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[355] W. Falcon. "PyTorch Lightning". In: *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning* 3 (2019).

[356] J. Chen, W. Geng, and G.-W. Wei. "MLIMC: Machine learning-based implicit-solvent Monte Carlo". In: *Chinese Journal of Chemical Physics* 34.6 (2021), pp. 683–694. DOI: 10.1063/1674-0068/cjcp2109150.

[357] T. Hamelryck. "An amino acid has two sides: a new 2D measure provides a different view of solvent exposure". In: *Proteins: Structure, Function, and Bioinformatics* 59.1 (2005), pp. 38–48.

[358] D. P. Kingma and J. L. Ba. "Adam: A method for stochastic gradient descent". In: *ICLR: International Conference on Learning Representations*. 2015, pp. 1–15.

[359] T. Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[360] M. Sundararajan, A. Taly, and Q. Yan. *Axiomatic Attribution for Deep Networks*. 2017. arXiv: 1703.01365 [cs.LG].

[361] S. M. Lundberg and S.-I. Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[362] A. V. Onufriev and E. Alexov. "Protonation and pK changes in protein–ligand binding". In: *Quarterly Reviews of Biophysics* 46.2 (2013), pp. 181–209. DOI: 10.1017/S0033583513000024.

[363] C. R. Ellis et al. "Constant pH molecular dynamics reveals pH-modulated binding of two small-molecule BACE1 inhibitors". In: *The journal of physical chemistry letters* 7.6 (2016), pp. 944–949.

[364] W. Chen, Y. Huang, and J. Shen. "Conformational activation of a transmembrane proton channel from constant pH molecular dynamics". In: *The journal of physical chemistry letters* 7.19 (2016), pp. 3961–3966.

[365] S.-T. Hong et al. "Anti-tnf alpha antibody humira with ph-dependent binding characteristics: a constant-ph molecular dynamics, gaussian accelerated molecular dynamics, and in vitro study". In: *Biomolecules* 11.2 (2021), p. 334.

[366] K. Jha, S. Saha, and H. Singh. "Prediction of protein–protein interaction using graph neural networks". In: *Scientific Reports* 12.1 (2022), pp. 1–12.

[367] A. Strokach et al. "Fast and flexible protein design using deep graph neural networks". In: *Cell systems* 11.4 (2020), pp. 402–411.

[368] Y. Xia et al. "GraphBind: protein structural context embedded rules learned by hierarchical graph neural networks for recognizing nucleic-acid-binding residues". In: *Nucleic acids research* 49.9 (2021), e51–e51.

[369] M. Réau et al. "DeepRank-GNN: A Graph Neural Network Framework to Learn Patterns in Protein-Protein Interfaces". In: *bioRxiv* (2021).

[370] S. Pittala and C. Bailey-Kellogg. "Learning context-aware structural representations to predict antigen and antibody binding interfaces". In: *Bioinformatics* 36.13 (2020), pp. 3996–4003.

[371] M. Fey and J. E. Lenssen. "Fast Graph Representation Learning with PyTorch Geometric". In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*. 2019.

[372] D. Hendrycks and K. Gimpel. *Gaussian Error Linear Units (GELUs)*. 2020. arXiv: 1606.08415 [cs.LG].

[373] E. Bunkute et al. "PIP-DB: the Protein Isoelectric Point database". In: *Bioinformatics* 31.2 (2014), pp. 295–296. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu637. eprint: https://academic.oup.com/bioinformatics/article-pdf/31/2/295/6999997/btu637.pdf. URL: https://doi.org/10.1093/bioinformatics/btu637.

[374] H. M. Berman et al. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (2000), pp. 235–242. ISSN: 0305-1048. DOI: 10.1093/nar/28.1.235. eprint: https://academic.oup.com/nar/article-pdf/28/1/235/9895144/280235.pdf. URL: https://doi.org/10.1093/nar/28.1.235.

[375] J. Jumper et al. "High Accuracy Protein Structure Prediction Using Deep Learning". In: *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction*. 2020.

[376] L. P. Kozlowski. "Proteome-pI: proteome isoelectric point database". In: *Nucleic Acids Research* 45.D1 (2016), pp. D1112–D1116. ISSN: 0305-1048. DOI: 10.1093/nar/gkw978. eprint: https://academic.oup.com/nar/article-pdf/45/D1/D1112/28861634/gkw978.pdf. URL: https://doi.org/10.1093/nar/gkw978.

[377] Y. Rose et al. "RCSB Protein Data Bank: Architectural Advances Towards Integrated Searching and Efficient Access to Macromolecular Structure Data from the PDB Archive". In: *Journal of Molecular Biology* (2020), p. 166704. ISSN: 0022-2836. DOI: https://doi.org/10.1016/j.jmb.2020.11.003. URL: https://www.sciencedirect.com/science/article/pii/S0022283620306227.

[378] M. Steinegger and J. Söding. "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets". In: *Nature Biotechnology* 35.11 (2017), pp. 1026–1028. DOI: 10.1038/nbt.3988.

[379] P. J. A. Cock et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (2009), pp. 1422–1423. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp163. eprint: https://

academic.oup.com/bioinformatics/article-pdf/25/11/1422/944180/btp163.pdf. URL: https://doi.org/10.1093/bioinformatics/btp163.

[380] M. F. Sanner, A. J. Olson, and J.-C. Spehner. "Reduced surface: An efficient way to compute molecular surfaces". In: *Biopolymers* 38.3 (1996), pp. 305–320.

[381] W. Kabsch and C. Sander. "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features". In: *Biopolymers: Original Research on Biomolecules* 22.12 (1983), pp. 2577–2637.

[382] G. Van Zundert et al. "The HADDOCK2. 2 web server: user-friendly integrative modeling of biomolecular complexes". In: *Journal of molecular biology* 428.4 (2016), pp. 720–725.

[383] D. S. Goodsell, G. M. Morris, and A. J. Olson. "Automated docking of flexible ligands: applications of AutoDock". In: *Journal of molecular recognition* 9.1 (1996), pp. 1–5.

[384] T. J. Dolinsky et al. "PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations". In: *Nucleic acids research* 35.suppl_2 (2007), W522–W525.

[385] L. P. Kozlowski. "Proteome-pI: proteome isoelectric point database". In: *Nucleic acids research* 45.D1 (2017), pp. D1112–D1116.

[386] L. P. Kozlowski. "IPC 2.0: prediction of isoelectric point and pKa dissociation constants". In: *Nucleic Acids Research* 49.W1 (2021), W285–W292.

[387] "Protein Data Bank: the single global archive for 3D macromolecular structure data". In: *Nucleic acids research* 47.D1 (2019), pp. D520–D528.

[388] "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic acids research* 49.D1 (2021), pp. D480–D489.

[389] A. Baptista. *meadTools software package*. 2001. URL: https://www.itqb.unl.pt/labs/molecular-simulation/in-house-software (visited on 05/11/2020).

[390] T. J. Dolinsky et al. "PDB2PQR: an automated pipeline for the setup of Poisson–Boltzmann electrostatics calculations". In: *Nucleic Acids Res.* 32.suppl_2 (2004), W665–W667.