Software Engineering Assignment

Christopher Kolloff

August 31, 2023

1 Introduction

Proteins play a fundamental role in the processes of life, and their dynamics are central to understanding these intricate processes. My research primarily revolves around how the protein transitions between relevant structural states, what the speed of these interchanges are and how structural rearrangements relate to the protein's function. I try to understand their dynamic behavior through a combination of computational and experimental methodologies.

One key area of my study focuses on feature representation learning of protein conformations. In this project, the goal is to effectively capture protein dynamics, which remains a challenge despite the advances made in machine learning. To address this, I employ temporal graph neural networks in conjunction with equivariant representations. This strategy aims to create a comprehensive and precise map of a protein's structural states, enhancing our understanding of the relevant protein conformations.

Another aspect of my research involves the integration of both experimental and computational data. Such integration is essential as it mitigates the biases that are inherent with computational methods. I am developing statistical models that provide a dynamic atomistic perspective of protein behavior. More specifically, I utilize generative diffusion models to dissect protein–ligand interactions, with the constraints set by experimental observations. This knowledge is instrumental for example for the development of drugs and therapeutics, as it offers insights into the mechanism-of-action of proteins.

Moreover, the exploration of protein's conformational space constrained by experimental data is another crucial aspect of my work. Despite the potential of denoising diffusion probabilistic models in sampling various protein structures, there remains a significant gap in integrating experimental constraints for modeling the kinetics and thermodynamics of biomolecules. In my research, I aim to develop methodologies that sample relevant conformational states guided by static and dynamic experimental observables.

In summary, my research approach towards understanding protein dynamics aims to bridge the gaps between experimental data and computational models and to offer insights that could improve our understanding of protein behavior and their therapeutic implications.

2 Robert's lectures

2.1 Test Cases and Boundary Value Analysis

In my research I find the principles of Software Engineering extremely valuable, particularly when it comes to creating test cases. By designing conditions or sets of conditions to evaluate aspects of my software I make sure that the tools I develop align precisely with my intended goals. These test cases allow me to confirm the accuracy and effectiveness of my software tools ensuring that any output they produce is reliable and consistent. For instance I often use toy systems, like 1D energy potentials where I have control over every parameter to ensure that the algorithm performs as expected. Furthermore especially when working with networks reproducibility is a key factor. Through test cases I provide documentation on how my software's expected to behave under various circumstances. It's also important to include tests with scenarios such as overlapping atoms or twisted bonds – this is known as boundary value analysis. Lastly as I strive to optimize the efficiency of my software test cases provide insights, into its performance. They help me identify areas that need improvement and allow me to maximize the potential of the tools I develop.

2.2 Behavioral Software Engineering

The field of Behavioral Software Engineering (BSE) focuses on the elements involved in software development including the behaviors, emotions and cognitive processes of developers and stakeholders. Although BSE currently doesn't play a role in my research I would like to highlight its impact, on projects in the future going beyond just technical aspects. To begin with by understanding the patterns of collaborators I can optimize the design and development process of my tools. This involves identifying biases or behavioral tendencies that may pose challenges during the development cycle. By being aware of these pitfalls in advance proactive measures can be taken to address them effectively. Furthermore BSE plays a role in improving team dynamics. Collaboration is often essential in research endeavors so understanding the behavioral aspects of team members is key to fostering communication, better decision making capabilities and increased productivity. This aspect becomes particularly valuable when collaborating with researchers who have little to no coding experience like experimentalists. Lastly as the project approaches its stages before making the code public BSE can offer insights, from an end users perspective. This ensures that considerations are made not from a standpoint but also take into account how users will interact with and benefit from the software. By examining the aspects of users of my research tools or applications I can customize the software to be more user friendly, intuitive and, in line with users expectations and requirements. Essentially incorporating Behavioral Software Engineering into my research not improves the software development process. Also takes a comprehensive approach by considering the human factors that are crucial, for a successful (research) project.

3 Guest lectures

3.1 DevBots

Productivity and accuracy are essential, in the field of software engineering. There are tools that play a role in achieving these goals. One such tool that has become a part of my research workflow is GitHub Copilot. I rely on it for essentially all of my projects. GitHub Copilot greatly simplifies the coding process by predicting and suggesting lines or blocks of code based on the context. This saves me an amount of time when it comes to repetitive coding tasks. With this efficiency I can dedicate time to algorithm development or data analysis. It also accelerates the documentation process by providing code based references. In addition to its time saving capabilities GitHub Copilot acts as a collaborator by offering suggestions from a repository of code and patterns, which goes beyond what I might consider, suggesting methods or functions. Another advantage is error prevention. GitHub Copilot detects semantic mistakes in time minimizing debugging efforts, which is particularly valuable when dealing with complex software components. Lastly as my research often involves collaboration GitHub Copilot helps maintain code quality and consistency. It ensures that the codebase remains accessible and understandable for collaborators, with backgrounds.

3.2 AutoML

Automated Machine Learning (AutoML) aims to automate the process of applying machine learning to real world problems. Integrating AutoML into my research holds promise. Can offer significant benefits. To begin with feature selection plays a role, in machine learning particularly when dealing with datasets such as high dimensional structural data of proteins. AutoML can systematically and independently identify the features from my data potentially uncovering new insights and relationships in protein dynamics that might be overlooked through manual feature engineering. This is particularly fascinating in relation to the project mentioned in the introduction, where I employ GNNs along with representations to pinpoint dynamically relevant conformations. Moreover model selection and hyperparameter tuning often consume an amount of time in machine learning. AutoML can efficiently address these aspects. With its ability to rapidly test and compare machine learning models available AutoML ensures the optimal model for my dataset thereby enhancing prediction accuracy and reliability. Additionally given the nature of my research which involves refining and rerunning models with data, AutoMLs adaptive learning capabilities can expedite this process while ensuring seamless evolution of models, alongside emerging data. Currently I am not utilizing automated machine learning. However incorporating it into my research has the potential to significantly enhance and streamline the procedures.

4 Role of Software Architecture and Privacy in my research

4.1 Automated Software Testing

Automated software testing is using pre-made tests to automatically check that software works right, performs well, and is reliable. It's not like manual testing where you need people to run tests and look at the results. Automated testing uses scripts and tools to run tests and say if they passed or failed. The main advantage is it's efficient - automated tests can run all the time without extra cost and they're usually faster and more thorough than manual tests and this matters a lot for agile development and continuous deployment where new versions are made really fast. In molecular simulations for biology, its important that machine learning models are accurate. You could make automated tests to check machine learning models for biomolecules are working well and reliably. For both molecular dynamics and machine learning getting good data is crucial. Automated testing could help ensure the data stay good through analysis and comparing to experiments, so there's less difference between the simulation and real results. My own research is about making modeling techniques that take a lot of computing power more efficient. Automated tests could always evaluate if the algorithms are staying efficient, so we make better use of computing resources. Pharma companies, bioinformatics companies and research labs want accurate, efficient biomolecular simulations more and more. Offering automated software testing tailored for this field could therefore be a business opportunity.

Related paper:

Michael C. Gerten, James I. Lathrop, Myra B. Cohen, and Titus H. Klinge. 2021. ChemTest: an automated software testing framework for an emerging paradigm. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20). Association for Computing Machinery, New York, NY, USA, 548–560. `https://doi.org/10.1145/3324884.3416638`

4.2 Human–Computer Interaction

Human Computer Interaction (HCI) focuses on studying how individuals utilize computers and creating user interfaces that are intuitive and responsive, to users needs. It involves a combination of computer science, psychology and engineering to facilitate interaction between humans and computers. In my research HCI has gained importance as it allows researchers with computing experience to comprehend complex data more effectively. By employing designed platforms and data visualizations based on HCI principles the complexities of molecular simulations can be simplified. Furthermore the emergence of AI and machine learning has revolutionized fields like predicting protein folding. HCI plays a role in developing interfaces that enable researchers to understand how AI processes data instead of treating it as a black box. This transparency is expected to encourage researchers and pharmaceutical companies to embrace these tools. From a perspective HCI holds potential in assisting pharmaceutical companies in comprehending the effects of their drug candidates at

the molecular level. With user interfaces powered by AI the entire process from drug design, to clinical trials can be accelerated efficiently.

Related paper:

Hossain, M.M., Roy, B., Roy, C., Schneider, K. (2023). Extensibility Challenges of Scientific Workflow Management Systems. In: Mori, H., Asahi, Y. (eds) Human Interface and the Management of Information. HCII 2023. Lecture Notes in Computer Science, vol 14016. Springer, Cham. `https://doi.org/10.1007/978-3-031-35129-7_4`

5 Future Trends and Directions of Software Engineering in AI4Science

Considering the progress, in AI and ML it's clear that software engineering will have a more significant role in the future. One trend I anticipate based on my research is the increasing demand for advanced, modular and scalable software architectures. As our models and algorithms become complex designed solutions will be crucial to ensure reproducibility, scalability and collaboration. This is particularly relevant for researchers like me who may not have an education in software engineering. The principles of SE will be essential to ensure that the software tools we develop today can adapt to changes.

When it comes to my career whether I choose to remain in academia or venture into industry SE will play a role in shaping my path. In academia funding agencies and journal publishers are starting to emphasize the importance of reproducibility and transparency. On the hand in industry settings, efficient, reliable and maintainable code serves as the foundation for developing products and services. Therefore improving my SE skills will enhance my versatility. Open up career opportunities across both sectors.

Looking ahead over the 5 to 10 years I believe there will be a shift, in how SE's perceived within the AI community. I believe that it won't be perceived as a skill but rather as a core discipline that every AI researcher or practitioner should follow. Moreover as AI systems become a component of decision making processes the need, for software engineering solutions that prioritize ethics, fairness, and transparency will continue to grow.