WASP Software Engineering Assignment

Leo Dahl

2025

1 Introduction

My research focuses on the visualisation and analysis of proteomics data. Proteins are biological molecules present throughout our bodies, performing life-essential functions in basically all parts of biology. Proteomics is the large-scale study of these molecules, and we typically have data generated from human blood samples. This data usually consists of tables of a number of proteins (features, typically ranging from a few hundred to a few thousand) measured in a number of samples (in our studies typically a few hundred, if many over a thousand). The goal is to process and analyse the data to find links to some biological condition, such as a disease or an infection and visualise these in appropriate ways. To reach this goal I write software for pipelines that take care of the data from data wrangling, normalisation, analysis and visualisation. While I do not write my own artificial intelligence software, I do use AI and ML tools for the analysis of the data. The final product in the software sense tends to be a repository with code for the pipeline, available for anyone to use but with limited direct usefulness for people who are not interested in adapting the code for their own data. To make the results of the anlayses more accessible I have for some projects developed web applications containing interactive visualisations of results for users to freely explore the results of the studies. I have also together with other group members created a software package for the normalisation of the data type that we usually work with, where there is also a focus on how users can easily apply the package to their own data.

2 Lecture principles

1. QA and testing (lecture 2) are of course of great importance for developing a data analysis pipeline to make sure that the pipeline is actually performing the analysis it is supposed to do. Since it is not possible to know exactly what outcome we expect from our own data for dynamic testing, I often use some

small and unrelated test data that are well known so that the output from the pipeline makes sense. In this very narrow case we do not have the problem of an infinite input space as the final input will be our own data. In the case of a package for normalisation the input space is larger, but still limited in the sense that it is only meant for a specific type of data from a specific kind of biological assay, which limits the input space. Still, something like property-based testing could be of interest to find unusual bugs that were not thought of when the code was developed. I also sometimes engage with other group members in a very informal kind of static code review to see if the code makes sense.

2. A topic that is probably relevant for any kind of software is the consideration of the requirements (or requirements engineering, lecture 4). We often work with sensitive personal data, meaning that in many of our projects ethics and safety take are at the top when it comes to requirements. We have to decide what our requirements are in terms of how the data will be processed (limiting what tools we can use to keep the data safe) and how the results should be presented in the end (to not reveal any sensitive information). Other requirements that are often important are of course robustness, accuracy, and reproducibility, which then influence the analyses we perform and the programming and project management tools that are used. Not all of the projects have the same kinds of end product (some will be a pipeline to accompany a paper, some an application for interactive use, some a software package for programming), so the requirements are often different, but we do not usually follow any model for designing the requirements (sometimes they are not even written down), which is why it could be good to look into for our projects.

3 Guest-lecture principles

1. Building on the previous point about requirements engineering, the concepts covered in the Requirements Engineering guest lecture could be relevant for my research. Stakeholder elicitation is one such concept as we rarely explicitly consider the stakeholders beyond having a vague idea (such as making figures understandable by a general audience). Making it clear who will be using the final products would make it easier to come up with the requirements, such as (using an example from the lecture) choosing what options to include for colours when making an interactive visualisation.

2. Another point is the requirement elicitation. As mentioned earlier we do not

always write down the requirements, so having a template for doing so would be helpful in keeping track of what needs have to be met for each project.

4 Data Scientists versus Software Engineers

- From the perspective of an outsider of both fields (coming from biology/bioinformatics), the differences sound reasonable, such as their educational backgrounds being different and their approaches to tackle their tasks. I see many similarities between how the data scientists are described and how I work in bioinformatics, but I think some of the similarities may stem from them both being more like academia while software engineering is often tied more to industry-like environments.

- While the roles definitely have some overlapping qualities, I don't think they will merge. They can learn basics from each other to better take each other into consideration, but they both need in-depth domain-specific knowledge that make the roles distinct and makes it better for them to cooperate, much like how research teams in biology/biotechnology benefit from having different people specialising in biology/experiments and statistics/data analysis (even if the roles are less similar there).

5 Paper analysis

5.1 Paper 1

E. J. Husom, S. Sen, and A. Goknil, 'Engineering Carbon Emission-aware Machine Learning Pipelines', in 2024 IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN), Apr. 2024, pp. 118–128.

1. The paper deals with the topic of the environmental impact of ML pipelines. While ML such as deep learning is revolutionising data analysis in many fields, it comes with considerable environmental impact due to requiring large amounts of energy. The authors of the paper made a ML pipeline, called CEMAI, for more environment-aware ML development throughout the whole process. The two main features are the use of CodeCarbon, a tool that tracks the energy consumption and estimates the carbon emissions, at regular intervals at all stages of the pipeline, and the use of data version control, to skip pipeline steps that would be unnecessary and reduce redundant reruns of the pipeline. The authors tested the pipeline on three different data sets and evaluated the carbon emissions of different pipeline steps, as well

as an evaluation of the impact of hardware (which felt a bit lacking in comparison). By using CEMAI, ML pipelines can be made while considering the carbon emissions of different choices such as how the data is processed or what model is used, so that both model performance and environmental impact can be optimised for.

2. The environmental impact of the research is something all researchers should consider at some point (as well as other sustainability aspects). As I use ML in some, if not most, of my data analysis projects, the question that the paper deals with is of relevance to my research.

3. A fictional large project could be a project with samples from e.g. 10 000 individuals from different cohorts (with different diseases), where each sample has been assayed for 5 000 proteins using large-scale multiplexed affinity proteomics methods. The goal would be to find protein panels that could accurately predict which disease, if any, each individual has, using ML methods, while also optimising for lower carbon emissions. This would require a whole analysis pipeline from extensive data processing (to get the different cohorts to work together), training of different ML models, to performance evaluation. The pipeline from the paper could help find the best analysis setup for performance and emissions. While the numbers are fictional and larger than what I typically have, the setup is similar to my projects and the CEMAI pipeline could definitely be used in a similar way in my research.

4. While I already use workflow management software in my daily research (which uses version control to skip execution of redundant steps), the environmental impact is not something I measure currently. Given the importance of the carbon footprint of ML pipelines, the pipeline that the authors propose or something similar could be considered for future projects to make my research more environmentally aware.

5.2 Paper 2

K. Khadka, J. Chandrasekaran, Y. Lei, R. N. Kacker, and D. Richard Kuhn, 'A Combinatorial Approach to Hyperparameter Optimization', in 2024 IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI (CAIN), Apr. 2024, pp. 140–149.

1. The second paper I selected deals with hyperparameter optimisation (HPO) and is not entirely unrelated to the first paper as they both try to reduce the resource usage of ML pipelines. Traditional approaches to perform HPO can

be either resource-intensive or unreliable, which is why the authors propose the use t-way testing (which is used in software testing) for HPO to potentially reduce the number of required hyperparameter combinations to evaluate and save resources. The approach is tested on four datasets (classification and regression) and compared to grid search, random search, and Bayesian optimisation. The t (number of hyperparameters to check combinations between) is also evaluated. By doing this, the authors aim to evaluate whether t-way testing is more time efficient and results in better model performance compared to other HPO approaches.

2. In my research I usually use the grid search approach for HPO which can take time. t-way testing could serve as a way to reduce that time without sacrificing performance.

3. Taking the previously described fictional project of large-scale proteomic profiling of multiple disease cohorts, t-way testing could be employed together with the green ML pipeline. This could potentially save time (especially as the project would try different ML methods) and reduce the environmental burden of HPO, which also fits with the ideas of the previous paper.

4. t-way testing sounds like a fairly simple change to implement going from grid search. The paper also seems like a good advertisement for Bayesian optimisation as it was usually the fastest with performance not that far behind the other methods. t-way testing could however be a good middleground where some time can be saved without having the extra things that need consideration for Bayesian optimisation, such as priors. In the long temr I would then also look into Bayesian optimization to see if it would be a viable replacement.

6 Research Ethics & Synthesis Reflection

The search was very primitive and basic as I had no idea what papers I could expect and how they could be related to my research. I simply went to the accepted papers section of a CAIN conference page, first for 2025 and then 2024, and read through the titles of the long papers to see if there was anything that seemed like it could be relevant and interesting (luckily the number of papers was not extremely high per year).

I did not have much trouble finding articles that seemed interesting, although not all articles I looked at were selected. There were some that dealt with ML in different fields like recognition of features in images that I skipped. In the end I

went with articles that were very general and whose ideas could be applied to pretty much any field of research using ML.

Originality was ensured by writing everything without the use of LLMs.