Software Engineering for AI/AS

Daniel Brunnsåker, *Data Science and AI*
Chalmers University of Technology
Gothenburg, Sweden
danbru@chalmers.se

✦

1 INTRODUCTION

My work lies at the intersection of systems biology, automation, and artificial intelligence, aiming to accelerate and improve scientific discovery in complex biological systems. Systems biology is a field that seeks to understand how genes, proteins, and metabolites interact across time and varying conditions. Given the immense complexity of even comparatively simpler organisms, this field increasingly relies on computational tools, high-throughput experimentation, and structured knowledge integration.

A central theme of my research has been the development of automated, scalable workflows that allow for rapid, iterative exploration of biological phenomena. In my early work, we designed high-throughput experimental pipelines combining computational planning with automated wet-lab setups. Using mass spectrometry techniques, we explored gene functions related to temporal biological phenomena [3].

Another major component of my research involves the use of explainable AI (usually through relational learning or logic programming) to extract interpretable rules and genotype-phenotype relationships from large-scale biological datasets. By integrating structured biological data with AI techniques, one can uncover biologically meaningful "rules", which can then be validated on experimental data [2].

To further increase efficiency and reproducibility in biological research, I have also worked on automating data acquisition itself. By integrating mass spectrometry instrumentation with standard lab automation tools, we enabled hands-free, ultra high-throughput metabolomics analysis, hopefully paving the way for more complete automated experimental cycles using intracellular biochemicals as readouts [7].

Currently, I am developing an integrated, end-to-end platform for automated scientific discovery. This system combines hypothesis generation from structured knowledge using inductive logic programming, automated experimental execution via robotic platforms, and real-time data analysis using mass spectrometry [1].

Throughout my work, baker's yeast has served as the organism of choice due to its rich knowledge base and biological tractability. However, the broader goal is to build frameworks and methodologies that generalize beyond yeast.

TLDR: In short, my research is focused on building methods and tools that enable quick, interpretable, and autonomous biological discovery. This involves integrating computational tools, automation, and structured biological knowledge representations to turn the traditionally slow, manual scientific method into a scalable, iterative process more adapted to modern research in the life sciences.

2 LECTURE PRINCIPLES

One principle from software engineering that has been repeatedly used in my research is regarding the verification and validation of machine learning models. In traditional software engineering, verification addresses the question of whether a system has been built correctly, while validation considers whether the right system has been built in the first place. When applied to my research, verification corresponds to ensuring that my computational models behave as intended, i.e., training is reproducible, data is processed correctly, predictive performance is consistent and interpretation is robust. Validation, on the other hand, addresses the scientific side of the problem. Here, the central question is whether the models outputs are scientifically meaningful. For example, rules uncovered through the applied machine learning techniques must not only produce novel hypotheses, but must also align with existing biological knowledge and be testable through biological experiments.

A second concept (albeit very much a subset of the previously mentioned concept) is about testing in real-world contexts. In classical software engineering, this would correspond to concepts like integration and system testing. While unit tests can confirm that individual components of a program function correctly, real-world testing ensures that the entire system operates (somewhat) reliably in the environment it is supposed to work in. Since a lot of my work involves the direct integration with real-life systems, tests need to be much more encompassing. Automated lab workflows bring together computational hypothesis generation, robotic execution, and live biological samples. Even if the underlying machine learning models perform well, their value depends on whether their predictions remain robust when enacted in the lab (noting that even a falsification is a good result!). For instance, an ML-derived experimental plan should be biologically relevant, and its outcomes should reliably support the question that is being asked. This kind of testing resembles the deployment of safety-critical ML systems (although far from as harsh testing conditions), where robustness must be established not only in simulation but also under real-world conditions.

3 GUEST-LECTURE PRINCIPLES

Note that I decided to interpret some of these concepts very liberally for ease of comparison.

A theme brought up during one of the guest lectures (by Per Lenberg from SAAB) that resonates with my research is the importance of human aspects in software and machine learning. As AI systems increasingly take on roles that were once performed by researchers (such as designing experiments or interpreting large-scale biological data in my specific case), human involvement is likely to instead shift towards ensuring that these systems are understandable, aligned with scientific goals and are fulfilling the requirements necessary to satisfyingly answer the scientific issue at hand.

Expanding on the point of requirements (a concept covered by Julian Frattini). In software engineering, requirements could define what a system needs to achieve, preferably capturing both functional and non-functional needs. Translating this to my research, requirements engineering is crucial because these systems operate at the intersection of computation, automation, and biology. Functional requirements might specify that a workflow must generate hypotheses, run robotic experiments, and analyse some biological readout of the experiment. Non-functional requirements, however, are just as critical (and probably more important of the system is to have any larger impact): the system should be reproducible, scalable, and interpretable.

4 DATA SCIENTISTS VS. SOFTWARE ENGINEERS

Roughly summarizing the text from the book [5], it seems like the main differences revolve around two main concepts:

- Data scientists focus on building accurate models and analysing them in more experimental settings, while
- Software (ML) engineers focus on deployment, scaling and maintaining the models in a real-word setting.

While I do (to some extent) agree with the distinctions made between data scientists and software engineers, I also think that both perspectives are essential. Looking forward, I think individuals will increasingly need more "cross-disciplinary literacy", i.e. data scientists will need to learn some engineering practices, and engineers should try to gain more specific ML knowledge. Realistically though, i expect that in larger organizations, specialization will likely increase regardless. Specialization allows experts to focus on their strengths, but this will put an increasingly large strain on the interface between them (an issue also raised in the introductory chapters). Smaller teams may continue to favor "hybrid" roles, but realistically, both trends will coexist.

Anecdotally though, I feel I have achieved more success by embracing both aspects of engineering and "science". It has often served me well to have a more pragmatic approach to design and implementation (for example, keep it as simple as possible, as anything more complicated is far more likely to break down in the end).

5 PAPER ANALYSIS

What About the Data? A Mapping Study on Data Engineering for AI Systems

Core ideas

The core concept presented in the paper [4] is that AI is (at least partly) limited by the engineering of its data infrastructure (or rather, the practices regarding the infrastructure). The paper maps several recent papers (25, between 2019 and 2023) and subsequently show that most of the actual work done in the field focuses mostly on data pipelines rather than robust data infrastructure and maintenance. Consequently, they also denote the importance of including "DataOps" (in contrast to the already established concepts of DevOps and MLOps) to a much higher extent than what is done today.

Connection to my research

My work aims to generate and use scientific knowledge automatically, which involves combining structured knowledge with large-scale experimental data. One of the challenges is that biological data are highly heterogeneous and often missing metadata or contextual details. This directly reflects the issues raised in the paper: without robust data engineering, the discovery pipeline cannot be reliable. For example, the paper's focus on data validation across training and data pipelines mirrors the need in my work to ensure that, for example, mass spectrometry data is correctly captured, labeled, and comparable across experiments. Similarly, their push for more encompassing data ecosystems connects to my efforts to generalize discovery frameworks beyond specific biological organisms and labs. In this context, my use of ontologies provides standardized vocabularies and structured relationships to help fill in missing context, potentially harmonize heterogeneous data sources, and ensure that knowledge extracted by these systems remains interpretable and reusable across experiments and domains.

Integration into a larger project/future outlook

One example of a fictional (or not so fictional, depending on if you're the DoE or not) project could be a multi-site platform for autonomous labs, where scientific hypotheses are generated automatically, tested through robotic execution, analyzed in real time, and then used to refine the systems own models of the world. Such a project requires much more than just a good model. It would need a highly robust data infrastructure. The practices described in the paper such as DataOps processes and knowledge engineering could hopefully help in ensuring quality and interpretability, but also enable interoperability.

My primary research would fit as the scientific reasoning and automation layer within this project, but more generally, I think that proper data practices are especially important in this setting. Something which I have tackled to some extent, but tend to make simplifications (essentially barring myself from scaling it properly). More fully integrating a "DataOps" mindset could helo alleviate these issues.

Bringing Machine Learning Models Beyond the Experimental Stage with Explainable AI

Core ideas

The central idea of the paper [6] is that explainable AI (XAI) is an enabler for bringing machine learning models beyond proof-of-concept and into production systems. In their case study, the authors show that models only gained traction (i.e. actually used) once explanations were integrated into the workflow. Explanations increased user trust, enabled domain experts to validate and monitor models themselves, and created an interface between human judgment and ML-based predictions. Not only did this increase use, but it also enabled significant model improvements, as domain-experts could provide advice on how to better align it with their intuition and expertise.

Connection to my research

This connects directly to my own research, where I work on building systems for automated scientific discovery. In science, the importance of interpretability is quite high. A hypothesis that cannot be transparently explained is unlikely to be tested, let alone accepted as scientific knowledge. My use of explainable machine learning and controlled vocabularies is aimed at ensuring that models generate interpretable rules and explanations, which can then be critically assessed and validated through experiments. Science is for humans, not robots.

Integration into a larger project/future outlook

It is currently very popular to use LLMs for everything, especially in AI4Science. One could imagine a larger AI project where a large language model (LLM) is used to generate scientific hypotheses (that are then down the line potentially experimentally validated). In this setting, transparency is essential. A hypothesis is unlikely to be tested or accepted if its underlying reasoning is opaque. The ideas from the paper fit naturally here, as explainable AI techniques could potentially provide the interpretability layer that makes LLM outputs useful to scientists. Rather than producing free-form text, the LLM would need to supply clear reasoning paths, connections to existing knowledge, and structured explanations that domain experts can analyse (and potentially use to modify the initial approach).

   Note that I have no clue on how to implement this. This topic was also covered in the second lecture (on Quality Assurance and Testing). Explainable models make the system much easier to properly test.

Some caveats

Whilst I do like this approach, and I think explainability is extremely important, the methodology in the paper does also raise some concerns. Very generally, they are outsourcing the explanations to specific post-hoc method (SHAP), which also could have potential underlying issues in explanatory power. I also think it is important to be aware that one does not want to "over-align" to the domain-expert(s). There could be interesting or important patterns that could be ignored in this way.

6 RESEARCH ETHICS & SYNTHESIS REFLECTION

I looked through the catalogue of the latest CAIN papers, sorting for long/full length papers. As my own research is typically based on explainable hypothesis generation and adherence to scientific knowledge, I tried to mainly look for topics based on either xAI or structured data.

While I did find it difficult to exactly relate the topics covered here with my own research, I think it was somewhat easy to find tangential topics. Scientific discovery is not necessarily a high-risk (or safety critical), and is a good test-bed for a lot of these concepts. Additionally, whilst I did not read through all of the accepted papers in the conference, there did not seem to be many situations in where they actually applied their insights in practice. Although, there were instances where some do it indirectly through observational studies (e.g. paper 1), whilst some do directly evaluate it (a reason why i liked the second paper mentioned). I also had some difficulty understanding the lingo, but luckily most of these concepts were mentioned in the lectures in some form or another (e.g., "code smells").

I think this in general was quite an interesting assignment. As somewhat of an outsider (I don't have a very rigourous software background...), it is quite easy to inadvertently ignore aspects of "proper practice" that might be obvious to other people. I especially liked the parts on testing, as I think this is completely fundamental to any process (scientific or not).

REFERENCES

[1] Daniel Brunnsåker. Agentic ai integrated with scientific knowledge: Laboratory validation in systems biology. *bioRxiv*, 2025. Preprint, posted August 18, 2025.

[2] Daniel Brunnsåker, Filip Kronström, Ievgeniia A. Tiukova, and Ross D. King. Interpreting protein abundance in Saccharomyces cerevisiae through relational learning. *Bioinformatics*, 40(2):btae050, 2024.

[3] Daniel Brunnsåker, Gabriel K. Reder, Nikul K. Soni, Otto I. Savolainen, Alexander H. Gower, Ievgeniia A. Tiukova, and Ross D. King. High-throughput metabolomics for the design and validation of a diauxic shift model. *npj Systems Biology and Applications*, 9:11, 2023.

[4] Petra Heck. What about the data? a mapping study on data engineering for ai systems. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, CAIN '24, page 43–52, New York, NY, USA, 2024. Association for Computing Machinery.

[5] Christian Kästner. *Machine Learning in Production: From Models to Products*. MIT Press, Cambridge, MA, 2025.

[6] Niels With Mikkelsen, Lasse Steen Pedersen, Mansoor Hussain, Victor Foged, and Ekkart Kindler. Bringing machine learning models beyond the experimental stage with explainable ai. In *2025 IEEE/ACM 4th International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pages 119–129, 2025.

[7] G. K. Reder et al. Autonoms: Automated ion mobility metabolomic fingerprinting. *Journal of the American Society for Mass Spectrometry*, 35(3):542–550, 2024.