Collaborative Coding and Reproducible Research: A Three-Year Course Retrospective

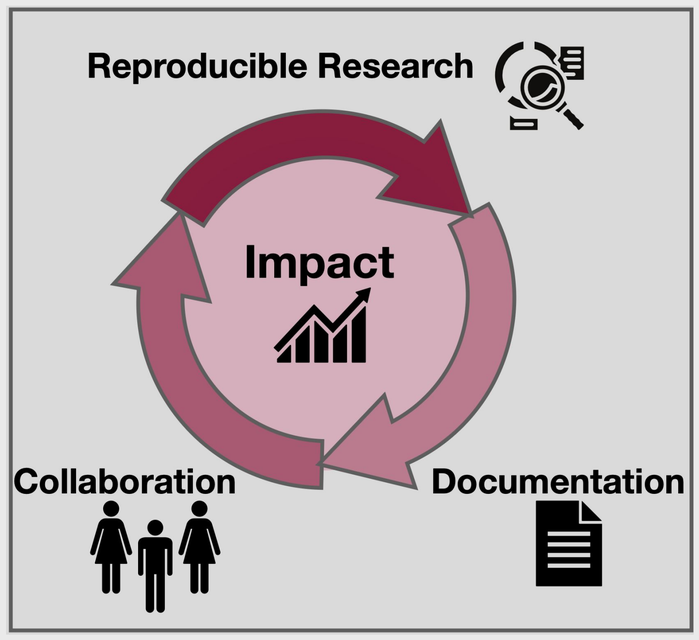
Elisabeth Wetzer[1], Gregor Decristoforo[2], Kristoffer Wickstrøm[1]
[1]Machine Learning Group, Dept. of Physics and Technology, UiT
[2]Research Software Engineering Group, Seksjon for digitale forskningsressurser og infrastruktur, UiT

Motivation

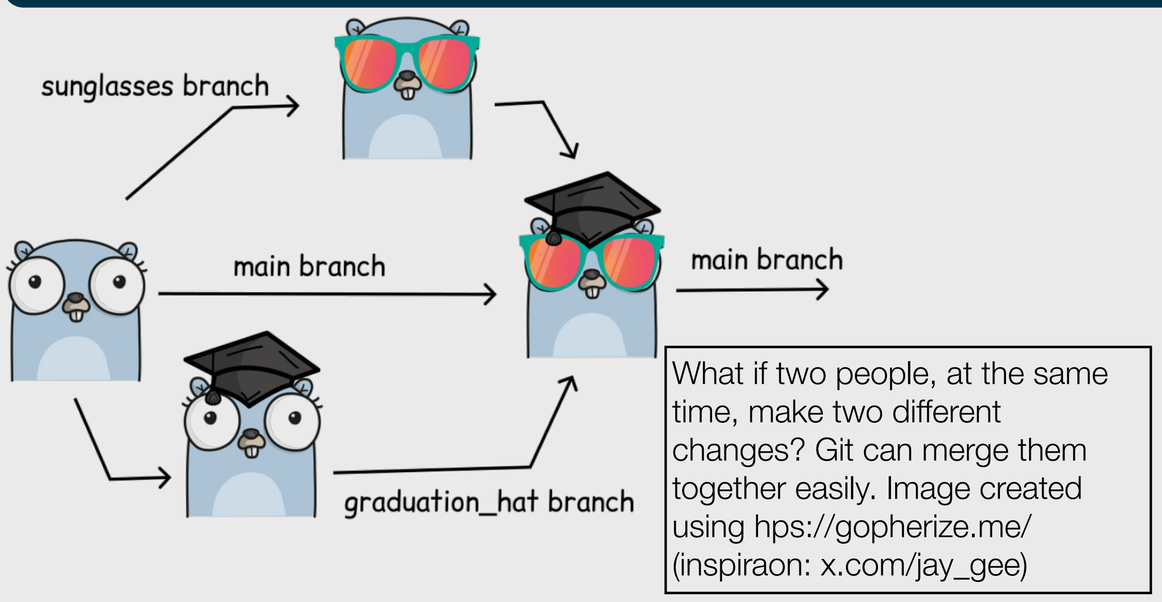
- **Need for Reproducible Code:** Without reproducible code, computational experiments and findings of a scientific paper cannot be independently verified, limiting transparency, undermining trust, and hindering the ability of others to build upon the research.
- **Lack of Training in Best Coding Practices and Version Control:** Academia is highly interdisciplinary, with many researchers specializing in computational approaches coming from diverse backgrounds such as physics, signal processing, or mathematics.
- **From Publication to Impact:** To drive innovation and create meaningful impact, research must extend beyond publications and translate into actionable, shareable, and practical outcomes

1.) *How can we introduce good practices to a broader audience, such as master's students, before they develop their own ad-hoc solutions?*

2.) *What types of assessments can effectively measure individual students' proficiency in collaborative coding?*



Course Content (5 credits)

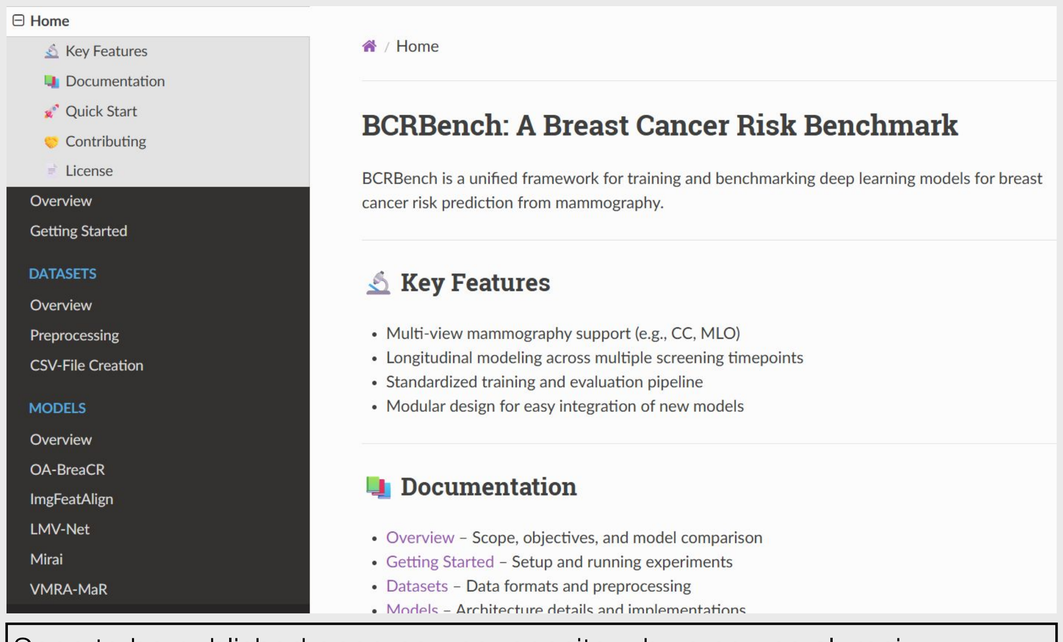


What if two people, at the same time, make two different changes? Git can merge them together easily. Image created using hps://gopherize.me/ (inspiraon: x.com/jay_gee)

- **Version Control and Collaboration:** Learn Git basics, collaborative workflows, code reviews, and best practices for managing projects.
- **Development Tools and Practices:** Explore automated testing, modular code design, profiling, reproducible environments.
- **Documentation, Notebooks, and Publishing:** Create effective documentation, work with notebooks, and release code as Python packages with proper licensing.
- **AI and Practical Application:** Use AI-assisted coding tools, work on real-world projects, and receive feedback to improve coding skills.

Success Stories

- **Three previous editions:** The course was held in spring and fall 2025, and spring 2026 with positive feedback by the attending PhD students.
- **Increased Interest:** As a group we observed increased interest in developing project repositories and prepare them for publications in open software journals



Soon to be published open-source repository by a course alumni: github.com/sot176/BreastCancerRiskBenchmark unifying state-of-the-art models within a single reproducible framework for breast cancer risk modeling with comprehensive documentation and opportunity for community contributions

Course Outcomes

- **Master Reproducible Research Practices:** Develop essential skills to ensure your research is transparent, reproducible, and easily shareable.
- **Hands-On Experience with Key Tools:** Practical expertise in version control (Git/GitHub), automated testing, code quality standards, HPC.
- **Collaborative Learning and Real-World Application:** Work in teams on a collaborative home exam to develop and document a reproducible codebase, applying principles of teamwork.

Resources

- **Course Material:**
  fys-8805-collaborative-coding.github.io/lecture-material
  github.com/SFI-Visual-Intelligence/Collaborative-Coding-Exam
- **Contact:**
  elisabeth.wetzer@uit.no
  gregor.decristoforo@uit.no
  kristoffer.k.wickstrom@uit.no
- **Code Refinery:**
  coderefinery.org