DY 38: Focus Session: Innovations in Research Software Engineering (joint session BP/DY)

Research software engineering (RSE) is an emerging field in science, with practitioners spanning a continuous spectrum from "researchers who code" to "software engineers developing for science". In Germany, a growing movement supported by deRSE e.V. is gaining recognition, and more institutions are acknowledging the increasing demand across various disciplines. This focus session will provide a platform to highlight recent advances in applications, tooling, and software in the fields of biophysics, dynamics, and statistical physics, as well as developments in the recognition and proliferation of RSE as a profession within our field and academia in general.

Organized by Simon Christ and Sophia Rudorf (Hannover).

Time: Thursday 15:00–18:00 Location: H44

Invited Talk DY 38.1 Thu 15:00 H44

Community-driven software and data training for computational biology — •Toby Hodges — The Carpentries, Oakland, CA, USA

The Carpentries is a global community teaching essential software and data skills for research. Certified Instructors teach hundreds of workshops to thousands of learners all over the world every year, introducing them to essential skills for computational research such as programming, version control, and data organisation. In recent years, the community has also begun to develop and deliver lessons that build on these foundations, teaching more intermediate and advanced Research Software Engineering skills such as HPC, parallel programming, and containerised computing. This talk will explore how open source, collaborative training efforts can build capacity for computational research, discuss what makes this model work and some lessons learned along the way, and finish with a look at what the community plans to do next.

DY 38.2 Thu 15:30 H44

Python-based interface to micromagnetic simulation software: Ubermag — •Hans Fangohr[1,2,3], Martin Lang[1,2], Samuel J.R. Holt[1,2], Swapneel Amit Pathak[1,2], Kauser Zulfiqar[1,2,4], and Marijan Beg[5] — [1]MPSD, Hamburg, Germany — [2]CFEL, Hamburg, Germany — [3]Univ. Southampton, UK — [4]Univ. Hamburg, Germany — [5]Imperial College London, UK

We describe the Python-based user environment "Ubermag" to help scientists use well-established (micromagnetic) simulation packages.

Within Ubermag [1], researchers can express the physics problem they want to simulate in a scientist-friendly but machine readable problem definition based on Python syntax [2]. Ubermag translates this problem into the configuration files needed for micromagnetic simulation packages such as OOMMF or mumax3. On completion of the simulation, the computed data is presented back to the user at the Python level. Ubermag is often used in Jupyter Notebooks, and supports rich media to provide figures and equations within the notebook.

We report on the motivation for Ubermag, the design and implementation process, and our experiences made both from the perspective of science users and from the research software engineers. We touch on a range of topics, including interface design, domain specific languages, testing, packaging, Jupyter, and reproducibility.

This work was supported by EPSRC UK Skyrmion Grant EP/N032128/1, and the European research projects OpenDreamKit (676541) and MaMMoS (101135546).

[1] DOI 10.1109/tmag.2021.3078896; [2] DOI 10.1063/1.4977225

DY 38.3 Thu 15:45 H44

OCTOPOS.jl: A Julia-based tool for synonymous codon optimization — Simon Christ[1], Jan-Hendrik Trösemeier[2], and •Sophia Rudorf[1] — [1]Institute of Cell Biology and Biophysics, Leibniz University Hannover, Germany — [2]independent researcher

OCTOPOS.jl is a research software designed to optimize synonymous mRNA sequences for improved heterologous gene expression in various host organisms. Combining a detailed mechanistic model of in-vivo protein synthesis with machine learning, OCTOPOS.jl predicts protein expression based on codon choice. Originally developed as a Java desktop application, the software has been reimplemented in the Julia programming language to enhance performance, modularity, and scalability. The new implementation serves as the foundation for a graphical user interface and a web application, accessible at https://octopos.cell.uni-hannover.de/. These updates improve accessibility and usability, broadening its appeal to both computational and experimental biologists. OCTOPOS.jl supports organism-specific genetic sequence engineering and detailed analysis of translation dynamics, thus providing a valuable resource for the synthetic biology and biotechnology communities.

DY 38.4 Thu 16:00 H44

Invert pattern forming systems with BayesFlow to bridge the gap from simulation to experimental observation — •Hans Olischläger — Interdisciplinary Center for Scientific Computing (IWR) — Heidelberg University

The description of experimental systems by complex spatial models, be it with (stochastic) partial differential equations, agent-based simulation or otherwise, is often the condensation of all the central scientific hypotheses regarding a particular object of study.

I argue, that making progress in this kind of modelling is currently hindered by the lack of a tool that enables solving the following inverse problem: Given an observation, determine all the model configurations that are able to produce it. In other words, what is the posterior probability of all model configurations given some (set of) experimental data.

Instead of just preaching that in theory a Bayesian treatment would be nice, I will then continue to present such a tool: amortized Bayesian inference (as implemented in the software package BayesFlow). I will give examples on the classical Gierer-Meinhardt pattern forming PDE and a biophysical model, the Min system, which is used by E. coli to control cell division.

I will also take a step back to give a broader picture of the newly available statistical methods that support complex spatial modelling and their limitations. The aim is to provide some guidance on what you can and cannot infer from your state-of-the-art scientific simulator given observations, and how to do it.

DY 38.5 Thu 16:15 H44

FAIR Data Management for Soft Matter Simulations using NOMAD — •Bernadette Mohr[1], Esma Boydas[1], Nathan Daelman[1], José M. Pizarro[1], Tristan Bereau[3], Claudia Draxl[1], Luca M. Ghiringhelli[4], Martin Girard[2], Denis Usvyat[6], Roser Valentí[7], Silvana Botti[5], and Joseph F. Rudzinski[1,2] — [1]CSMB, HU Berlin — [2]MPIP Mainz — [3]ITP, Heidelberg Uni. — [4]Dept. of Mater. Sci. and Eng., FAU Erlangen — [5]RC-FEMS and Faculty of Physics, RUB Bochum — [6]Inst. für Chem., HU Berlin — [7]ITP, GU FfM

NOMAD [nomad-lab.eu][1, 2] is an open-source, community-driven data infrastructure designed to facilitate FAIR data management in materials science. Currently, it supports over 60 computational codes and encompasses DFT, classical MD, and many-body methods. This contribution will focus on recent developments, following modern software practices, to enhance NOMAD's applicability to soft matter and biological systems, including support for coarse-grained representations and advanced workflows such as free energy calculations. Combined with a schema for representing force fields, molecular topologies, and hierarchical system structures, NOMAD tracks data provenance and streamlines data analysis and the creation of AI-ready datasets. The NOMAD framework meets the classical simulation community's needs for improved data management standards and provides a foundation for building a cohesive, interconnected scientific data ecosystem.

[1] Scheidgen, M. et al., JOSS 8, 5388 (2023).

[2] Scheffler, M. et al., Nature 604, 635-642 (2022).

DY 38.6 Thu 16:30 H44

Estimation of kinetic rates by constrained optimization — •Federico Marotta[1], Maria Zimmermann-Kogadeeva[1], Peer

Bork[1], Julia Mahamid[1], and Sophia Rudorf[2] — [1]European Molecular Biology Laboratory — [2]Leibniz Universität Hannover

Biological systems often rely on molecular motors to perform useful work. The kinetics of the reactions in a motor's cycle can be easily investigated *in vitro* or in model organisms, but it is difficult to generalize them to a different system. We present a method to estimate the transition kinetics in an uncharacterized system, where minimal data are available, by leveraging a reference system where the kinetics have been elucidated. The motor's activity is represented as a continuous-time Markov chain, characterized by an infinitesimal generator matrix Q whose entries are functions of the transition rates of the cycle (the vector ω) and possibly of the concentrations of external molecules. In the uncharacterized system, the available data induce a constraint on the admissible rates. By employing an extremum principle, we estimate the rates ω_{unc} that minimize the kinetic distance with respect to the reference rates ω_{ref} while respecting such constraint. As an application of this strategy, we describe a model of the translation elongation cycle, where reference data are available for *E. coli in vitro*, and estimate the rates either *in vivo* or in a different organism, under constraints on the total elongation time or the steady-state occupancies, respectively.

DY 38.7 Thu 16:45 H44

Software provisioning for HPC and RSE — •Martin Lang[1,2], Henning Glawe[1,2], Jehferson Mello[1,2], and Hans Fangohr[1,2,3] — [1]Max Planck Institute for the Structure and Dynamics of Matter, Hamburg, Germany — [2]Center for Free-Electron Laser Science, Hamburg, Germany — [3]University of Southampton, Southampton, UK

All research software relies on existing libraries for various functionalities such as low-level math operations, FFTs, IO, or other domain-specific operations. Installing these dependencies, potentially based on different compilers or in multiple versions, with all inter-dependencies fulfilled is notoriously difficult.

In the first part of this talk we introduce the open-source package manager Spack, which has a strong focus on HPC and research software. Spack can install software in multiple versions and variants, and supports optimised compilation for the underlying hardware, including compiling on exotic hardware. It comes with a large, community-provided collection of commonly used packages. Spack's packaging files make it easy to specify required dependencies, provide optional features of a software, and ensure compatibility with other libraries.

In the second part we present the concrete setup at our institute. We use Spack to provide the software stack on the local HPC, including pre-compiled packages and toolchains (sets of compilers and libraries) for users to compile their own software. We report on requirements and challenges, and how we address these with Spack. We also touch on scripting the Spack-based installation process including the option to recreate the HPC software environment on a scientist's laptop.

DY 38.8 Thu 17:00 H44

Small scale Research Software Engineering — •Simon Christ — Leibniz Universität Hannover, Institut für Zellbiologie und Biophysik, Computational Biology

While we are in dire need of research software organizations on a faculty level or larger, small scale software engineering, that is one research software engineer in a group or institute, is something that can be achieved in a short time frame and is probably the most common form today. A field report from Computational Biology where research software engineers are involved in modeling, developing solutions, teaching and maintenance.

DY 38.9 Thu 17:15 H44

Estimation of pKa values in membrane bound proteins — •Jesse Jones[1], Nereu Montserrat i Busquets[1,2], Ana Gamiz Hernandez[3], Ville Kaila[3], and Maria Andrea Mroginski[1] — [1]Technische Universität Berlin, Berlin — [2]Freie Universität Berlin, Berlin — [3]Stockholm Universität, Stockholm

Many key bioenergetic processes involving electron and proton reactions take place in membrane bound protein complexes, generating a proton motive force. Yet the ionizable groups which facilitate these reactions are often buried in hydrophobic pockets in the membrane. These processes are mainly described through pK_a values, which continue to be poorly understood and difficult to obtain despite structural, biochemical and computational advances. Hence, estimating pK_a values of these residues without the need for weeks of work in a laboratory, is important to describe the dynamics of the system, providing information on possible proton pathways. In this work we preview Karlsberg3, a software which uses a Poisson Boltzmann Equation solver (APBS) for proteins and calculates pKa values. Karlsberg3 is, in contrast to its predecessor Karlsberg2+, parallelized, running in modern software environments, and able to take membranes into consideration.

DY 38.10 Thu 17:30 H44

The teachingRSE project - Towards a professionalization of RSE education. — •Florian Goth[1] and Simon Christ[2] — [1]Universität Würzburg, Institut für theoretische Physik und Astrophysik, Am Hubland, 97074 Würzburg — [2]Leibniz Universität Hannover, Institut für Zellbiologie und Biophysik, Herrenhäuser Str. 2 30419 Hannover

At the deRSE23, the second conference for research software engineering(RSE) in Germany, a group of people came together for a small workshop to discuss how to deal with questions revolving around RSE education. Overwhelmed by the immense resonance to that workshop we took home a tremendous amount of feedback that made obvious that a short blog post will not suffice to adequately represent it. Now it is two years later, and the project produced its first output, the second position paper https://arxiv.org/abs/2311.11457 of de-RSE e.V. and it has sprawled out into a multitude of follow-up projects. In this talk, I will give an overview over the original ideas that we tried to convey in the position paper, and go into more detail on how domain sciences like physics need to change in light of this new specialization.

DY 38.11 Thu 17:45 H44

Python-Based Analysis Pipeline for the Quantification of Mechanics in Neural Organoids — •Michael Frischmann[1,2], Elijah R. Shelton[1], Achim T. Brinkop[1,2], and Friedhelm Serwane[1,2,3] — [1]Faculty of Physics & Center for NanoScience, LMU Munich, Germany — [2]Institute of Biophysics, Ulm University, Ulm, Germany — [3]SyNergy & GSN, Munich, Germany

Neuronal tissues form under the influence of mechanical forces guiding cellular movements. In the mammalian retina, neuronal translocations occur over hours. However, mechanical probing at those timescales in situ have posed experimental challenges. We employed magnetic ferrofluid droplets in mouse stem cell-derived retinal organoids to probe tissue mechanics from seconds to hours. To quantify tissue strain we have developed a Python-based analysis pipeline featuring an accessible graphical user interface (GUI). This pipeline automates strain quantification, image segmentation, and fitting procedures, enabling high-fidelity creep compliance measurements over extended durations. Our measurements reveal power-law scaling of dynamic compliance as well as tensile loss and storage modulus, consistent with soft glassy rheology just above the glass transition. These results demonstrate that neuronal tissues remodel in a scale-free manner while maintaining solid-like properties. This discovery provides a framework for understanding how mechanical signals may govern connectivity in the central nervous system. Integrating neural organoid models, mechanical probing, and computational methods, prepares us to investigate the interplay between biomechanics and neurodevelopment.