# **Abstract: Jorge Antonio Amador Balderas**

*NHR@FAU***Computer-assisted molecular modelling and dynamics can explain how Glu35 and Pro240 hSMUG1 mutants differently affect excision of uracil and 5-hydroxymethyluracil from single-stranded DNA**

Mutagenic uracil (U) arises in DNA by hydrolytic deamination of cytosine (C) while nonmutagenic U results from erroneous incorporation of deoxyuridine monophosphate opposite adenine during replication. Human single-strand-selective mono-functional uracil-DNA glycosylase 1 (hSMUG1) was first described as one of several enzymes to initiate the base excision repair (BER) pathway by excising U from DNA. hSMUG1 was also found to excise bases damaged by oxidation like 5-hydroxymethyluracil (hmU) from DNA in addition to being involved in RNA metabolism, where it has been suggested that hmU is excised from RNA. Connected to the latter functionality, is the interaction of several hSMUG1 residues including Ser26 and Glu35 with the pseudouridine synthase Dyskerin (DKC1) protein. Our experimental work shows that the S26R/E35D double mutant shows no excision activity compared to that of U, even when both residues are far away from the active site. We also observe that replacing Pro240 with Gly (P240G) abolishes hSMUG1 activity for hmU while U activity is retained. Unfortunately, there is no hSMUG1 crystal structure available that can help explain our results. The Xenopus laevis SMUG1 (xSMUG1) crystal structure has been determined without association with substrate or product. Through in silico modelling and molecular dynamics (MD) simulations, we managed to produce a hSMUG1-DNA complex and study the interactions between the substrate base and the active site residues. We observed that P240 stabilizes the interaction of H239 with the substrate, which explains the reduced stabilization in the P240G mutant. We also observed that E35 is part of a hydrogen bond chain that extends to an active site residue which discriminates between hmU and thymine as substrate. When E35 is replaced with a similar amino acid as in the E35D mutant, a slight conformational change misdirects the interaction; this prevents proper hmU binding while leaving U binding unaffected. These findings, coupled with subtle differences between hmU and U inside the active site, suggest that hmU excision is more susceptible to be affected by active site residue rearrangements, even if small.

# **Abstract: Fabian Böhm**

*NHR@FAU***Simulation of Earth Mantle Convection**

Certain problems in the natural sciences, such as mantle convection (the transport of heat and matter through the Earth’s outer mantle), are not fully understood and are beyond the reach of traditional experiments. In this particular example, the domain is too large to observe and inaccessible under kilometers of solid rock, so numerical modeling and simulation are required. My current research interest is to improve the efficient solution of the very large and sparse linear systems to which these simulations boil down. This highly interdisciplinary task requires three ingredients. The mathematical solution algorithm must be scalable and converge to the solution quickly and independently of the number of unknowns, making Multigrid the method of choice. On the computational side, supercomputers provide the necessary parallelism to partition the domain, and the main computational task, a matrix-vector multiplication (which is performed matrix-free), must have high performance at the node level. The latter can be achieved by code generation, i.e. compiling a mathematical model into architecture-aware, high-performance C++ code using programming language compiler techniques.

# **Abstract: Anila Ghazanfar and Julian Kunkel**

*Georg-August Universität Göttingen, Department of Computer Science***Toward Secure and Reliable HPC Workloads: Leveraging eBPF for Syscall Filtering**

Abstract High-performance computing (HPC) workloads in shared data centers require both strong isolation and resilience to ensure secure, uninterrupted execution—especially in domains like healthcare, where patient data must be protected under regulations such as HIPAA and computational failures can disrupt critical diagnostics. Traditional isolation mechanisms, such as virtual machines and containers, often introduce performance overhead or lack the fine-grained control needed for sensitive, time-critical applications. This research proposes a lightweight, kernel-level security approach using extended Berkeley Packet Filter (eBPF) to enforce system call filtering with minimal performance impact. It focuses on developing eBPF-based filters for file I/O operations, with ongoing efforts to integrate them into HPC workflows and benchmark their efficiency against existing methods such as SELinux and seccomp. The expected contributions include a customized syscall filtering framework for HPC, performance evaluations, and open-source tools. Future work will extend support to network and GPU-related syscalls and explore fault tolerance mechanisms

# **Abstract: A F M Mohimenul Joaa, Michael Färber**

*Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/ Leipzig, Technische Universität Dresden***RAG Search Space for Function Execution**

This poster presents a framework for enhancing the reliability of function execution in Large Language Models (LLMs) through Retrieval-Augmented Generation (RAG). My research investigates how symbolic function calls and hybrid retrieval can support accurate, multi-step reasoning and mitigate hallucinations. In my master’s thesis “Curious Learner,” I showed that decoupling logic from text via symbolic functions produces deterministic outputs and reduces errors. Building on that foundation, I participated in the LiveRAG Challenge and implemented a custom RAG system: it retrieves the top 20 documents using a dense and sparse hybrid (60/40), applies coreference resolution and topic clustering, summarizes each cluster, and uses a pretrained reranker to select relevant passages, if no passage meets the relevance threshold, fallback strategies—query expansion, hypothetical answer generation (HyDE), and hybrid embedding retrieval—ensure robust coverage. Future work will extend this pipeline into a decentralized, per-action authenticated RAG search space and integrate knowledge-graph grounding. We will explore mainly three core research questions: what alternatives to system-context prompting can improve search-space awareness; under what conditions can database access alone suffice for RAG; and how can an automated, decentralized “local-to-global” vector search space be constructed?

# **Abstract: Marvin Kaster**

*NHR4CES@TUDa***Structural plasticity and homeostatic memory formation**

Learning and memory are linked to synaptic plasticity, but the human brain undergoes structural changes even in adulthood. Relying solely on synaptic plasticity limits learning capabilities in suboptimal connected networks, as the human brain mostly has sparse connectivity. Structural plasticity addresses this by forming necessary synapses on demand. Recent work showed that homeostatic structural plasticity could create implicit Hebbian learning rules, leading to silent memory engrams but only allowing single engram formation. We use the Model of Structural Plasticity (MSP), which connects neurons based on Euclidean distance, enabling multiple engrams simultaneously and simulating around 4 million neurons while learning 343 distributed memories. The MSP maintains equilibrium based on firing rates. Low firing rates lead to synaptic formation, while high levels cause pruning. The network includes 20% inhibitory neurons in 3D space, divided into boxes. We conducted large-scale simulations with up to 343 boxes, modeling a conditioned learning paradigm with multiple excitatory neuron ensembles in each box representing a conditioned and unconditioned stimulus. After the learning phase, we observed increased firing of the ensemble representing the unconditioned stimuli when the conditioned stimuli were stimulated. Our distancedependent connectivity rule ensures that only nearby ensembles are learned, resulting in separate memories in each box instead of one large memory. Our model’s growth rules explained these eRects, as pruning and regrowing synapses during stimulation led to new synapses and stronger local connectivity. Our model predicts synapse pruning is integral to homeostatic learning and enforces a spatiotemporal order. Synaptic pruning precedes and enables associative learning, as it necessitates vacant axonal and dendritic elements. This requirement goes beyond a mere homeostatic response, where pruning occurs in response to increased activity resulting from higher synaptic input after learning. This diRers from purely Hebbian plasticity, where associative learning depends solely on the co-activity of the neurons.

# **Abstract: Jonas Kupschus and Andreas Schuppert**

*RWTH Aachen University***Hybrid Modeling for Neuropathic Pain**

Chronic pain is a multifaceted condition arising from complex interactions between physiological, psychological, and environmental factors. Longitudinal studies employing pain diaries with associated environmental measures have provided valuable insights into the lived experience of chronic neuropathic pain. While mechanistic neurobiological knowledge about the physiological background of some neuropathic pain conditions exists, current approaches lack the ability to accurately model pain levels from first principles. Machine learning offers a promising avenue to address this gap, leveraging large-scale patient data in the form of electronic health records to compensate for limitations in mechanistic understanding. Crucially, high performance computing enables us to efficiently simulate and analyze biophysical models as well as deep learning models informed by patient data: modern GPU-accelerated differential equation solvers that work seamlessly within deep learning environments, can help to build models of how pain levels change over time, even when unexpected events or treatments occur. Resource intensive Bayesian techniques allow for robust uncertainty quantification of parameter estimates in neuron models. By integrating mechanistic as well as data driven insights using high performance computing, we can take meaningful steps toward developing digital twins for chronic pain conditions, ultimately leading to more personalized and effective interventions.

# **Abstract: Sepehr Mahmoodianhamedani**

*NHR@Göttingen*

**Bridging HPC and Cloud Computing: Exploring Synergies for Next-Generation Computing Workflows**

The increasing demand for large-scale computational resources across scientific and industrial domains has driven growing interest in the convergence of High-Performance Computing (HPC) and cloud computing. While HPC offers high computational throughput and low-latency performance, cloud platforms provide flexible scalability, accessibility, and cost efficiency. This research investigates hybrid HPC-cloud architectures to harness the strengths of both paradigms for next-generation computing workflows. The study identifies and examines key technical and operational challenges in this convergence, including virtualization overhead, data transfer bottlenecks, orchestration complexity, and budget constraints. A literature review is conducted to assess the current landscape of hybrid HPC-cloud solutions, followed by a practical evaluation of open-source tools such as Singularity and Kubernetes. The goal is to evaluate their suitability for hybrid deployments with respect to portability, performance, and cost-effectiveness. Preliminary findings suggest that containerization and emerging cloud-native technologies are enabling more agile and interoperable HPC environments. The project also outlines future directions, particularly in the healthcare domain, where secure and compliant integration of edge devices with hybrid infrastructures may support real-time clinical decision-making and optimized resource management.

# **Abstract: Rudo Mtengwa**

*RWTH Aachen University*

**Real-Time Streaming Process Discovery on CPU-Based HPC Architectures**

Real-Time Streaming Process Discovery on CPU-Based HPC Architectures ABSTRACT As organizations generate continuous event data through systems like healthcare platforms, financial services, and smart manufacturing, the demand for real-time process visibility has become critical. Traditional process discovery methods are designed for static, post-event logs and struggle with high-volume, low-latency streaming environments. This limitation presents a barrier to proactive monitoring and adaptive decision-making in fast-paced operational settings. This research proposes a scalable framework for real-time streaming process discovery using CPU-based High-Performance Computing (HPC) architectures. The focus is on adapting and parallelizing existing discovery techniques specifically directly-follows graph (DFG) construction and Petri net generation to run efficiently across multi-core CPU clusters without requiring GPU acceleration. The system is designed to support continuous model updates in response to realtime event streams while maintaining model accuracy and responsiveness. Using tools like Apache Flink for streaming data ingestion and multi-threaded programming (Python) for core computation, the framework distributes trace processing and DFG updates across multiple CPU cores. Formal process models are incrementally generated and updated as Petri nets, allowing structured, visual monitoring of process behavior in real time. The system will be deployed on an HPC cluster and evaluated using large-scale synthetic and realworld datasets. Key performance indicators will include model latency, throughput, CPU utilization, and model quality (fitness and soundness). This research contributes to the field of process mining by offering a GPU-independent, scalable, and low-latency process discovery framework suitable for real-world, real-time applications.

# **Abstract: Yussur Mustafa Oraji**

*NHR4CES@TUDa*

**A Modular Contract Framework for Parallel Programming Models**

Parallel programming models such as the Message Passing Interface (MPI) and OpenSHMEM facilitate the use of distributed-memory computers. While powerful, these models are error-prone: Users may forget the MPI initialization, do not initialize a datatype, or leak a request handle. Thus, a lot of work has gone into dynamic correctness checking tools such as MUST, PARCOACH and ITAC, which, while possessing great accuracy, cause significant runtime overhead. Static tools, which perform analysis at compile time, avoid this overhead but trade it for lower accuracy. Additionally, most correctness checkers are inflexible, supporting only one parallel programming model of many, where adding support for another is generally nontrivial. We propose the use of contracts to verify the correct usage of parallel programming models. Contracts are a way to annotate userdefined rules in code for the compiler to check. The analyses that run to verify the contracts apply to any contract defined, and thus are not bound to a set of known programming models. By introducing a suitably powerful contract syntax the API requirements of the parallel programming model can be embedded directly into the code, and analysis can be performed regardless of the model used. Our tool aims to perform contract verification for parallel programs at compile-time using LLVM, enabling fast checking and debugging of some error classes such as local data races, handle lifecycle errors and more. The contracts embed the API specifications of each programming model using MPI and OpenSHMEM as examples. We use contracts for extensibility: New checks and support for other models can be easily implemented using additional contract annotations. By performing analysis using the LLVM IR future development may allow additional programming languages to be used with the same analysis framework. The analyses are implemented using eTicient data flow analysis, with each analysis relying on the same generic worklist algorithm. Additionally, we evaluate our tool against other static checkers on known correctness benchmarks RMARaceBench and MPI-BugBench. In our testing, our approach improves accuracy through wider error coverage with a compile-time overhead of around 2-3x, comparable to or improving upon the other tools tested.

# **Abstract: Anushka Pahuja**

*NHR@ZIB*

**Machine Learning-Driven Exploration of Ligand-Receptor Interactions**

Machine Learning-Driven Exploration of Ligand-Receptor Interactions Understanding and predicting ligand-receptor interactions is crucial for the rational design of safer and more targeted therapeutics. The μ-opioid receptor (MOR) is a prime example, where conventional opioid analgesics activate receptors systemically, leading to severe side effects. Emerging evidence suggests that inflammation creates distinct local micro-environments that could be exploited to design site-specific analgesics with reduced systemic activity. In this study, we investigate the conformational dynamics of endomorphin-1, a tetrapeptide ligand, by exploring all 256 permutations of its four amino acids—Tyrosine, Proline, Tryptophan, and Phenylalanine. Using molecular dynamics simulations and the ISOKANN framework, we compute χ-functions that describe the probability of the system occupying distinct macro-states, providing insight into potential transition pathways. To overcome the computational burden of large-scale simulations, we also propose a transfer learning approach using a hypernetwork model. This model aims to predict χ-functions directly from peptide backbone structures, bypassing the need for exhaustive MD simulations. Our approach offers a pathway toward more efficient and data-driven exploration of ligand-receptor interactions, supporting the development of next-generation, safer analgesics.

# **Abstract: Mohammad Rezaei**

*TU Dresden, NHR@TUD*

**Optimizing Multi-Step Reasoning in Large Language Models Using Reinforcement Learning**

This research explores methods to enhance multi-step reasoning in Large Language Models (LLMs) by framing reasoning as a search process through a tree of thought. At each step, the model proposes intermediate reasoning paths, which are structured into a dynamic reasoning tree. Each path is evaluated using feedback derived from execution outcomes (e.g., code correctness), and Reinforcement Learning (RL) is used to optimize the model’s preference for reasoning paths that lead to correct solutions. This approach targets domains like mathematical problem solving, where structured, sequential thinking is essential.

# **Abstract: Hang Song**

*NHR4CES@RWTH*

**Machine Learning-Based Reconstruction of Moment Equations**

This study proposes a machine learning-enhanced framework for solving moment equations in rarefied gas dynamics, overcoming key limitations of traditional moment methods. By integrating direct simulation Monte Carlo (DSMC) datasets spanning equilibrium to highly nonequilibrium flow regimes, we develop a fully connected neural network (FCNN) capable of concurrently achieving high-order moment predictions that surpass conventional closure schemes and collision integral estimates that exceed the approximations of the BGK model. The FCNN predictions are systematically integrated into a discontinuous Galerkin spectral element method (DG-SEM) solver through an adaptive equation reconstruction scheme R13-ML (regularized 13 moment equation with machine learning). Three core innovations underpin this framework: thermodynamically consistent training data generation via DSMC sampling, characteristic-scale-based flow normalization preserving rarefaction effects, and a novel coupling algorithm integrating machine learning with the discontinuous Galerkin framework. This methodology establishes a critical connection between kinetic theory and continuum-based solvers, providing a robust computational tool for rarefied gas flow simulations.

# **Abstract: Keshvi Tuteja**

*NHR@KIT*

**pyGinkgo: Python Bindings for Ginkgo**Sparse linear algebra is a cornerstone of many scientific computing and machine learning applications. Python has become a popular choice for these applications due to its simplicity and ease of use. Yet high-performance sparse kernels in Python remain limited in functionality, especially on modern CPU and GPU architectures. While contemporary machine learning libraries such as PyTorch and TensorFlow offer decently optimized kernels for dense matrix computations, their performance for sparse matrix operations often falls short. Ginkgo is a high-performance linear algebra library with a special focus on sparse linear systems. It boasts one of the fastest sparse matrix-vector (SpMV) kernels, the core operation in neural networks. To bridge the performance gap between dense and sparse computations in the Python world, we present pyGinkgo, a lightweight and Pythonic interface to the Ginkgo library, offering high-performance sparse linear algebra support with platform portability across CUDA, HIP, and OpenMP backends. pyGinkgo bridges the gap between high-performance C++ backends and Python usability by exposing Ginkgo’s capabilities via Pybind11 and a NumPy and PyTorch compatible interface. We also benchmarked pyGinkgo’s performance against state-of-the-art Python libraries including SciPy, CuPy, PyTorch and TensorFlow. Results across hardware from different vendors demonstrate that pyGinkgo consistently outperforms existing Python tools in both Sparse Matrix Vector (SpMV) product and iterative solver performance, while maintaining performance parity with native Ginkgo C++ code. Our work positions pyGinkgo as a compelling backend for sparse machine learning models and scientific workflows.

# **Abstract: Markus Vieth**

*NHR@SW*

**Efficient Life Science: Performance optimization of life science applications**In the field of life sciences, data-intensive workloads often demand high-performance computing solutions to process large datasets efficiently. GPUs have emerged as powerful tools in this domain due to their parallel processing capabilities, offering significant advantages in both speed and energy efficiency compared to traditional CPUs. However, many life science applications remain optimized for CPUs, missing out on the potential benefits of GPU acceleration. This research focuses on optimizing life science applications by developing data structures and algorithms specifically tailored for GPU architectures. By designing these components from the ground up with GPUs in mind, the aim is to unlock their full performance potential. One part of this work is the implementation of Hierarchical Interleaved Bloom Filters(HIBFs) on GPUs. Bloom filters are essential for membership testing, a common operation in genomics. The hierarchical and interleaved design enhances memory management and vectorized performance for queries on multiple sets, particularly beneficial for GPUs. Building upon existing research by Mehringer et al., who developed HIBFs for CPUs, and Jünger et al., who implemented efficient Bloom filters on GPUs, this work tries to combine these results to create an optimized implementation suitable for adoption in life sciences applications. Benchmarks will compare the performance of the GPU-optimized implementation against the CPU version by Mehringer et al., measuring runtime, scalability, and energy consumption across various workloads. This research has broad implications for the field of life sciences, demonstrating that GPU-accelerated computing can significantly enhance both performance and sustainability. By increasing processing speed of larger datasets while reducing energy consumption, this work encourages wider adoption of GPU technology in life science research, benefiting both scientists and the environment. In summary, this study highlights the potential for optimizing life science applications on GPUs through carefully designed data structures and algorithms.