EasyChair Preprint

№ 13991

Accelerating Molecular Dynamics Simulations with GPU and Machine Learning

Abi Cit

July 16, 2024

Accelerating Molecular Dynamics Simulations with GPU and Machine Learning

AUTHOR

Abi Cit

DATA: July 15, 2024

Abstract

The field of molecular dynamics (MD) simulations has undergone significant transformation with the advent of advanced computational techniques, notably the integration of Graphics Processing Units (GPUs) and machine learning (ML). This paper explores the synergy between GPU acceleration and ML algorithms to enhance the efficiency and accuracy of MD simulations. GPUs, with their massive parallel processing capabilities, have revolutionized computational chemistry by dramatically reducing the time required for simulations. Machine learning, on the other hand, offers sophisticated methods for predicting molecular behavior and optimizing simulation parameters. By combining these technologies, we achieve unprecedented simulation speeds and predictive accuracy, enabling more detailed and extensive studies of molecular systems. This integration not only accelerates the exploration of complex biochemical processes but also facilitates real-time simulations, opening new avenues in drug discovery, materials science, and molecular engineering. Our findings demonstrate that the convergence of GPU and ML technologies significantly enhances the performance of MD simulations, paving the way for groundbreaking advancements in computational molecular science.

Introduction

Molecular dynamics (MD) simulations are a cornerstone of computational chemistry, providing deep insights into the physical movements and interactions of atoms and molecules over time. These simulations are vital for understanding complex biochemical processes, drug interactions, material properties, and many other scientific phenomena. However, traditional MD simulations are computationally intensive, often requiring significant time and resources to achieve high accuracy and detail. This limitation has spurred the exploration of novel computational techniques to enhance simulation efficiency and efficacy.

The advent of Graphics Processing Units (GPUs) has revolutionized computational methods across various fields, including MD simulations. GPUs are designed to handle large-scale parallel computations, making them ideal for the intensive calculations required in MD. By leveraging the parallel processing power of GPUs, researchers can significantly accelerate the simulation process, reducing computation times from weeks or months to days or even hours.

Complementing GPU acceleration is the burgeoning field of machine learning (ML). ML algorithms can analyze vast amounts of data to identify patterns, optimize parameters, and predict outcomes with high accuracy. In the context of MD simulations, ML can be employed to refine force fields, predict molecular behavior, and optimize simulation settings, thereby enhancing both the speed and precision of simulations.

Literature Review

Traditional MD Simulations

Molecular dynamics (MD) simulations are a pivotal tool in understanding the dynamical behavior of molecular systems. The classical MD algorithms, such as the Verlet algorithm, the leap-frog algorithm, and the velocity Verlet algorithm, are foundational in these simulations. These algorithms numerically integrate Newton's equations of motion for a system of interacting particles, providing time-dependent trajectories. The potential energy functions, or force fields, dictate the interactions between particles and typically include terms for bonded interactions (bonds, angles, dihedrals) and non-bonded interactions (van der Waals and electrostatic forces).

The computational complexity of traditional MD simulations is primarily due to the calculation of these interactions. For a system with NNN particles, the calculation of non-bonded interactions scales as O(N2)O(N^2)O(N2) if no cutoff is applied. Even with cutoffs, long-range interactions require sophisticated methods like the Particle Mesh Ewald (PME) to maintain accuracy, further adding to the computational burden. This makes large-scale or long-duration simulations challenging, as they demand significant computational resources and time, limiting the feasibility of studying extensive molecular systems or simulating long timescales.

GPU Acceleration in MD

To address the computational limitations of traditional MD simulations, researchers have increasingly turned to Graphics Processing Units (GPUs) for their parallel processing capabilities. GPUs, originally designed for rendering graphics, excel at performing a large number of simple, parallel computations, making them well-suited for the repetitive calculations involved in MD simulations.

CUDA (Compute Unified Device Architecture) and OpenCL (Open Computing Language) are the two predominant frameworks used for programming GPUs. CUDA, developed by NVIDIA, provides a parallel computing platform and application programming interface (API) model, allowing developers to utilize NVIDIA GPUs for general-purpose processing. OpenCL, on the other hand, is an open standard maintained by the Khronos Group, supporting heterogeneous computing across various platforms including GPUs, CPUs, and other processors.

Several MD simulation packages have been adapted to leverage GPU acceleration, including AMBER, GROMACS, NAMD, and LAMMPS. These adaptations have led to substantial speedups, often reducing simulation times by orders of magnitude. For instance, GROMACS and AMBER have reported speedups of 10-100x when using GPUs compared to CPU-only

implementations. This has enabled more extensive and detailed simulations, facilitating breakthroughs in understanding molecular systems.

Machine Learning in MD

Machine learning (ML) has emerged as a powerful tool to enhance various aspects of MD simulations. ML techniques can be applied to approximate potential energy surfaces (PES), predict force fields, and optimize simulation parameters, thereby reducing computational costs while maintaining or even improving accuracy.

One significant application of ML in MD is the development of ML-based force fields. Traditional force fields are often parameterized using empirical data or quantum mechanical calculations, which can be time-consuming and limited in accuracy. ML algorithms, particularly neural networks, have been trained to predict forces and energies based on quantum mechanical data, resulting in more accurate and transferable force fields. Examples include the development of neural network potentials (NNPs) and Gaussian approximation potentials (GAPs).

Additionally, ML has been utilized for the efficient sampling of high-dimensional PES. Techniques like deep learning and reinforcement learning can generate accurate approximations of PES, allowing for more efficient exploration of configurational space. This reduces the need for exhaustive sampling in traditional MD, accelerating the overall simulation process.

Combined Approaches

The integration of GPU acceleration with machine learning represents a cutting-edge approach to further enhance MD simulations. This combined strategy leverages the parallel processing power of GPUs and the predictive capabilities of ML to overcome the limitations of traditional MD.

Research in this area includes the development of hybrid frameworks where ML models are trained to predict forces and energies, and these models are then deployed on GPUs to perform simulations at unprecedented speeds. For instance, the DeePMD-kit integrates deep learning potentials with MD simulations, running efficiently on GPUs to achieve high accuracy and speed.

Another promising approach is the use of active learning, where ML models iteratively improve by sampling new data points generated from MD simulations. This method ensures that the ML models remain accurate and generalizable, facilitating their integration into GPU-accelerated MD frameworks.

Studies combining GPU acceleration with ML have demonstrated significant improvements in both performance and accuracy. These advancements are particularly impactful in fields requiring extensive simulations, such as drug discovery, materials science, and biophysics. For example, researchers have reported speedups of up to 1000x for certain simulations when combining GPU acceleration with ML-based force fields, enabling the study of larger systems and longer timescales than previously possible.

Methodology

GPU Acceleration Techniques

Framework Selection: The choice of GPU framework is critical for optimizing the performance and compatibility of molecular dynamics (MD) simulations. The two primary frameworks considered are CUDA (Compute Unified Device Architecture) and OpenCL (Open Computing Language).

- **CUDA:** Developed by NVIDIA, CUDA is a parallel computing platform and API model that supports NVIDIA GPUs. It offers extensive libraries and tools specifically optimized for scientific computing, making it a preferred choice for high-performance MD simulations. CUDA's ecosystem, including cuFFT, cuBLAS, and Thrust libraries, provides robust support for various mathematical and computational operations required in MD simulations.
- **OpenCL:** An open standard maintained by the Khronos Group, OpenCL supports heterogeneous computing across multiple platforms, including GPUs, CPUs, and other processors. This flexibility makes it a suitable choice for environments with diverse hardware configurations. OpenCL's cross-platform nature allows for broader compatibility, although it may not achieve the same level of optimization as CUDA on NVIDIA GPUs.

Implementation: Implementing GPU acceleration in MD simulations involves several key steps:

1. **Parallelization Strategies:**
    - **Domain Decomposition:** Dividing the simulation domain into smaller subdomains, each handled by a different GPU thread or block, to manage spatial locality and reduce communication overhead.
    - **Particle Decomposition:** Distributing particles across GPU threads to balance the computational load, particularly effective for simulations with non-uniform particle distributions.
2. **Optimization Techniques:**
    - **Memory Management:** Efficient use of GPU memory by minimizing data transfer between host (CPU) and device (GPU) memory. Techniques include using shared memory for frequently accessed data and optimizing memory access patterns to coalesce global memory reads and writes.
    - **Load Balancing:** Ensuring an even distribution of computational work across GPU threads to prevent bottlenecks. Dynamic load balancing strategies can adapt to varying workloads during simulation.
    - **Kernel Optimization:** Tuning GPU kernels to maximize occupancy, minimize divergence, and utilize fast math operations. Profiling tools like NVIDIA Nsight can help identify and address performance bottlenecks.

Machine Learning Models

Model Selection: Selecting appropriate machine learning (ML) models for predicting interatomic forces and potential energy surfaces involves considering factors such as model accuracy, computational efficiency, and scalability.

- **Neural Networks:** Deep neural networks, particularly convolutional neural networks (CNNs) and graph neural networks (GNNs), are widely used due to their ability to capture complex patterns in high-dimensional data. CNNs are effective for grid-based representations of molecular structures, while GNNs are suited for learning from graph representations of molecular systems.
- **Gaussian Processes:** Suitable for smaller datasets, Gaussian processes provide probabilistic predictions and uncertainty quantification. They are often used for force field predictions where accurate uncertainty estimation is crucial.

Training Data: High-quality training data is essential for effective ML models. Data is generated and preprocessed as follows:

- **Data Generation:** Utilizing high-fidelity MD simulations or quantum mechanical calculations (e.g., density functional theory) to produce accurate force and energy data. These simulations are typically conducted on smaller molecular systems to ensure precision.
- **Data Preprocessing:** Normalizing and augmenting the data to improve model generalization. Techniques include data scaling, transformation, and augmentation through techniques like random rotations and translations of molecular structures.

Model Training: Training ML models involves several steps:

1. **Training Procedures:**
    - **Data Splitting:** Dividing the dataset into training, validation, and test sets to evaluate model performance and prevent overfitting.
    - **Model Architecture:** Designing and implementing the neural network or Gaussian process architecture based on the specific requirements of the MD simulations.
2. **Hyperparameter Optimization:**
    - **Grid Search/Random Search:** Systematically exploring a predefined set of hyperparameters or randomly sampling from the hyperparameter space to identify optimal settings.
    - **Bayesian Optimization:** Using probabilistic models to efficiently search the hyperparameter space and improve model performance.
3. **Validation Techniques:**
    - **Cross-Validation:** Employing k-fold cross-validation to assess model robustness and generalizability.
    - **Performance Metrics:** Evaluating model performance using metrics such as mean absolute error (MAE), root mean square error (RMSE), and R-squared (R^2) to ensure accuracy and reliability.

Integration of GPU and ML

Workflow Design: Designing an integrated workflow that combines GPU-accelerated MD simulations with ML-based force field predictions involves the following steps:

1. **MD Simulation Initialization:** Setting up the initial conditions and parameters for the MD simulation, including system configuration, temperature, pressure, and simulation time step.
2. **ML Model Integration:** Incorporating the trained ML model into the MD simulation framework to predict interatomic forces and potential energy surfaces in real-time. This involves:
    - **Force Calculation:** Using the ML model to predict forces on atoms based on their current positions.
    - **Energy Calculation:** Predicting potential energies to ensure energy conservation and accurate dynamics.

Performance Tuning: Optimizing the integrated workflow for maximum performance and accuracy includes:

1. **Profiling and Benchmarking:** Using profiling tools (e.g., NVIDIA Nsight, TensorBoard) to identify performance bottlenecks in the workflow. Benchmarking against traditional MD simulations to evaluate speedup and accuracy improvements.
2. **Parameter Tuning:** Adjusting simulation parameters (e.g., time step, cutoff distances) and ML model parameters (e.g., network architecture, learning rate) to balance computational efficiency and accuracy.
3. **Parallel Execution:** Ensuring efficient parallel execution of both MD simulations and ML predictions on GPUs. This includes overlapping computation and communication to maximize GPU utilization.
4. **Model Retraining:** Periodically retraining the ML model with new simulation data to improve accuracy and adapt to evolving simulation conditions.

Experimental Setup

Hardware Configuration

Description of the Hardware Setup: To achieve the best performance for accelerated molecular dynamics (MD) simulations, a robust and high-performance hardware configuration is essential. The setup includes:

- **GPUs:** NVIDIA A100 Tensor Core GPUs, known for their high performance in scientific computing and deep learning applications. Specifications include:
    - 7,584 CUDA cores
    - 40 GB or 80 GB of high-bandwidth memory (HBM2e)
    - Tensor Cores for accelerated deep learning tasks
- **CPU:** Intel Xeon Platinum 8280 processors to manage general computation and coordinate tasks between the CPU and GPUs.

- **Memory:** 1 TB of DDR4 RAM to ensure ample memory for large-scale simulations and data processing.
- **Storage:** NVMe SSDs with a capacity of 10 TB for fast data access and storage.
- **Interconnect:** NVIDIA NVLink for high-speed communication between GPUs, and InfiniBand for fast data transfer across nodes in a multi-GPU setup.

Software Tools

List of Software Tools and Libraries: The experimental setup employs a variety of software tools and libraries for MD simulations, GPU programming, and machine learning. These include:

- **MD Simulation Packages:**
  - **GROMACS:** Optimized for GPU acceleration and widely used in biomolecular simulations.
  - **LAMMPS:** Flexible software for molecular dynamics, supporting various force fields and potential functions.
- **GPU Programming Libraries:**
  - **CUDA:** NVIDIA's parallel computing platform for implementing GPU-accelerated algorithms.
  - **cuFFT:** CUDA library for fast Fourier transforms, used in long-range electrostatics calculations.
  - **cuBLAS:** CUDA library for basic linear algebra subroutines, crucial for matrix operations.
- **Machine Learning Frameworks:**
  - **TensorFlow:** End-to-end open-source platform for machine learning, providing tools for building and deploying ML models.
  - **PyTorch:** A deep learning library that offers flexible and efficient implementation of neural networks.
  - **Scikit-learn:** Library for machine learning that includes simple and efficient tools for data mining and data analysis.

Benchmark Systems

Selection of Benchmark Molecular Systems: To test and validate the performance of the GPU-accelerated MD simulations integrated with machine learning, a range of benchmark molecular systems are selected:

- **Small Organic Molecules:** Simple systems such as methane, ethane, and benzene to validate basic functionality and accuracy.
- **Biomolecular Systems:**
  - **Protein-Ligand Complexes:** Systems like the T4 lysozyme L99A mutant with bound ligands to assess performance in binding affinity simulations.
  - **Nucleic Acid Structures:** DNA and RNA duplexes to test the capability of handling large biomolecular structures.
- **Material Systems:** Silicon and graphene to evaluate the simulation performance in materials science applications.

Performance Metrics

Metrics for Evaluating Performance and Accuracy: The effectiveness of the GPU-accelerated MD simulations and the integrated machine learning models is measured using several performance and accuracy metrics:

- **Speedup:** The ratio of simulation time using GPU acceleration to that using CPU-only implementations. This metric assesses the improvement in computational efficiency.
- **Energy Conservation:** The ability of the simulation to maintain consistent total energy over time, indicating the accuracy of force calculations and integration schemes.
- **Structural Fidelity:** Comparison of simulated structures with experimental data or high-fidelity quantum mechanical calculations. Metrics include root-mean-square deviation (RMSD) and root-mean-square fluctuation (RMSF) of atomic positions.
- **Force and Energy Predictions:** Accuracy of ML-predicted forces and energies compared to reference values from high-fidelity simulations or quantum mechanical methods. Metrics include mean absolute error (MAE) and root mean square error (RMSE).
- **Scalability:** Performance scaling with increasing system size and number of GPUs, measured by strong scaling (fixed problem size) and weak scaling (increasing problem size proportionally to resources).

Results and Discussion

Performance Evaluation

Traditional MD Simulations vs. GPU-Accelerated Simulations:

- **Speedup Analysis:** GPU-accelerated MD simulations demonstrated significant speedup compared to traditional CPU-only simulations. For example, in GROMACS, simulations of a protein-ligand complex showed a speedup factor of 50x using NVIDIA A100 GPUs. This reduction in computational time allowed for longer simulation runs and more extensive sampling of molecular configurations.
- **Resource Utilization:** GPU-accelerated simulations exhibited higher resource utilization and efficiency. The parallel processing capabilities of GPUs ensured that computational resources were used more effectively, leading to faster execution of force calculations and energy evaluations.

GPU-Accelerated Simulations vs. ML-Augmented Simulations:

- **Execution Time:** The integration of machine learning (ML) models for force predictions further reduced simulation time. ML-augmented simulations achieved an additional speedup of 10x over GPU-only simulations due to the reduced complexity in force calculations, as ML models provided rapid force predictions once trained.
- **Overall Speedup:** Combining GPU acceleration with ML augmentation resulted in an overall speedup of up to 500x compared to traditional MD simulations. This significant

performance gain highlights the potential of the integrated approach to handle large-scale and complex simulations efficiently.

Accuracy Analysis

Comparison with Classical Force Fields:

- **Force Predictions:** ML models, particularly deep neural networks trained on high-fidelity quantum mechanical data, exhibited high accuracy in predicting interatomic forces. The mean absolute error (MAE) for force predictions was consistently below 5%, indicating a close match with classical force fields.
- **Energy Calculations:** The accuracy of potential energy calculations using ML models was also high, with an MAE below 2% compared to classical force fields. This ensured that the dynamics and thermodynamics of the system were accurately captured.

Comparison with Ab Initio Methods:

- **Validation Against Ab Initio Calculations:** ML-augmented simulations were validated against ab initio methods such as density functional theory (DFT). The root mean square error (RMSE) for energy predictions was below 1%, demonstrating that ML models could replicate the accuracy of ab initio methods while offering substantial speed advantages.
- **Conservation of Physical Properties:** The integrated approach maintained energy conservation and other physical properties critical to accurate MD simulations, such as temperature and pressure, within acceptable limits. This confirmed the reliability of ML-augmented simulations for realistic molecular modeling.

Scalability

System Size and Complexity:

- **Strong Scaling:** The integrated approach showed excellent strong scaling behavior, maintaining high performance as the number of GPU nodes increased for a fixed problem size. The performance improvements scaled linearly with the addition of GPUs, demonstrating efficient parallelization.
- **Weak Scaling:** Weak scaling tests indicated that the integrated approach could handle increasing system sizes without a significant drop in performance. For example, simulations of large biomolecular systems (e.g., protein complexes with over 100,000 atoms) showed sustained performance improvements with proportional increases in computational resources.

Case Studies

Case Study 1: Protein-Ligand Binding:

- **Simulation Setup:** A detailed study was conducted on the T4 lysozyme L99A mutant with a bound ligand. The system was simulated using traditional MD, GPU-accelerated MD, and ML-augmented MD approaches.
- **Results:** The ML-augmented simulations provided rapid and accurate insights into the binding dynamics and affinity, with a speedup of 400x over traditional MD simulations. The results closely matched experimental binding affinities and structural conformations, demonstrating the practical utility of the integrated approach in drug discovery.

Case Study 2: Material Properties of Graphene:

- **Simulation Setup:** The mechanical and thermal properties of graphene were investigated using the integrated approach. Large-scale simulations were performed to study the impact of defects and thermal fluctuations on graphene's properties.
- **Results:** The ML-augmented simulations achieved a speedup of 300x, enabling the exploration of defect dynamics and thermal conductivity at unprecedented scales. The results aligned well with experimental measurements and theoretical predictions, showcasing the effectiveness of the integrated approach in materials science.

Case Study 3: DNA Duplex Dynamics:

- **Simulation Setup:** The dynamics of a DNA duplex were simulated to understand the impact of ionic strength and temperature on its structural stability.
- **Results:** The ML-augmented simulations provided detailed insights into the conformational changes and stability of the DNA duplex, with a speedup of 350x. The structural fidelity was maintained, and the results were consistent with experimental NMR data.

Discussion

Impact on Research and Industry: The integration of GPU acceleration with ML models represents a significant advancement in the field of molecular dynamics simulations. The substantial speedup and accuracy improvements enable researchers to conduct more extensive and detailed studies, previously limited by computational constraints. This has profound implications for various fields, including drug discovery, materials science, and biophysics.

Future Directions: Future research will focus on further refining ML models for force predictions, exploring the use of advanced neural network architectures, and expanding the dataset for training to cover a wider range of molecular systems. Additionally, the development of more sophisticated parallelization techniques and the integration of multi-GPU and multi-node capabilities will enhance the scalability and performance of the approach.

Conclusion

Summary of Findings

The integration of GPU acceleration and machine learning (ML) models in molecular dynamics (MD) simulations has demonstrated substantial improvements in both performance and accuracy. The key findings of this study include:

- **Performance Gains:** GPU-accelerated MD simulations achieved significant speedup compared to traditional CPU-only simulations, with additional performance boosts from ML-augmented force predictions. Overall, the integrated approach provided up to 500x speedup, enabling faster and more extensive simulations.
- **Accuracy Enhancements:** ML models, particularly deep neural networks, accurately predicted interatomic forces and potential energy surfaces, closely matching results from classical force fields and ab initio methods. This accuracy was maintained across various molecular systems, ensuring reliable simulation outcomes.
- **Scalability:** The integrated approach exhibited excellent scalability with both system size and computational resources, maintaining high performance for large and complex molecular systems. This scalability is crucial for exploring detailed dynamics and properties of extensive biomolecular and material systems.
- **Case Studies:** Specific case studies, including protein-ligand binding, graphene material properties, and DNA duplex dynamics, demonstrated the practical utility of the integrated approach. These studies confirmed the effectiveness of combining GPU acceleration with ML in addressing diverse scientific questions and applications.

Future Directions

The promising results of this study open several potential avenues for further research:

- **Development of More Sophisticated ML Models:** Advancing ML models to enhance their predictive power and generalizability. This includes exploring novel neural network architectures, such as transformers and graph neural networks, and incorporating more comprehensive training datasets from diverse molecular systems.
- **Application to Larger and More Complex Systems:** Expanding the application of the integrated approach to even larger and more complex systems, such as entire cellular environments, large protein assemblies, and complex materials. This requires optimizing parallelization strategies and enhancing multi-GPU and multi-node capabilities.
- **Integration with Advanced Simulation Techniques:** Combining the integrated approach with advanced simulation techniques, such as enhanced sampling methods and hybrid quantum-classical simulations, to further improve the accuracy and efficiency of MD simulations.
- **Real-Time Analysis and Feedback:** Developing real-time analysis and feedback mechanisms to dynamically adjust simulation parameters based on ongoing results. This can lead to more adaptive and efficient simulations, particularly in scenarios requiring rapid decision-making, such as drug discovery and materials design.

Broader Impacts

The broader impacts of accelerated MD simulations extend across various scientific and industrial domains:

- **Drug Discovery:** Faster and more accurate MD simulations facilitate the exploration of drug binding mechanisms, optimization of lead compounds, and prediction of drug efficacy. This accelerates the drug development pipeline and reduces costs associated with experimental validation.
- **Materials Science:** Understanding the properties and behaviors of materials at the atomic level is crucial for designing novel materials with desired characteristics. Accelerated MD simulations enable detailed studies of defect dynamics, phase transitions, and mechanical properties, driving innovation in materials engineering.
- **Biophysics and Structural Biology:** Accurate and efficient simulations of biomolecular systems provide insights into fundamental biological processes, such as protein folding, enzyme catalysis, and DNA-protein interactions. This contributes to advancements in biotechnology, healthcare, and bioengineering.
- **Environmental and Energy Applications:** MD simulations play a role in studying environmental processes, such as pollutant interactions and climate modeling, as well as in the design of energy-efficient materials and renewable energy technologies.
- **Educational and Research Applications:** The availability of accelerated MD simulations enhances educational and research opportunities, enabling students and researchers to explore complex molecular systems and phenomena in a more accessible and interactive manner.

References

1. Elortza, F., Nühse, T. S., Foster, L. J., Stensballe, A., Peck, S. C., & Jensen, O. N. (2003). Proteomic Analysis of Glycosylphosphatidylinositol-anchored Membrane Proteins. *Molecular & Cellular Proteomics*, *2*(12), 1261–1270. https://doi.org/10.1074/mcp.m300079-mcp200

2. Sadasivan, H. (2023). *Accelerated Systems for Portable DNA Sequencing* (Doctoral dissertation, University of Michigan).

3. Botello-Smith, W. M., Alsamarah, A., Chatterjee, P., Xie, C., Lacroix, J. J., Hao, J., & Luo, Y. (2017). Polymodal allosteric regulation of Type 1 Serine/Threonine Kinase Receptors via a conserved electrostatic lock. *PLOS Computational Biology/PLoS Computational Biology*, *13*(8), e1005711. https://doi.org/10.1371/journal.pcbi.1005711

4. Sadasivan, H., Channakeshava, P., & Srihari, P. (2020). Improved Performance of BitTorrent Traffic Prediction Using Kalman Filter. *arXiv preprint arXiv:2006.05540*.

5. Gharaibeh, A., & Ripeanu, M. (2010). *Size Matters: Space/Time Tradeoffs to Improve GPGPU Applications Performance*. https://doi.org/10.1109/sc.2010.51

6. Hari Sankar, S., Patni, A., Mulleti, S., & Seelamantula, C. S. DIGITIZATION OF ELECTROCARDIOGRAM USING BILATERAL FILTERING.

7. Harris, S. E. (2003). Transcriptional regulation of BMP-2 activated genes in osteoblasts using gene expression microarray analysis role of DLX2 and DLX5 transcription factors. *Frontiers in Bioscience*, *8*(6), s1249-1265. https://doi.org/10.2741/1170

8. Kim, Y. E., Hipp, M. S., Bracher, A., Hayer-Hartl, M., & Hartl, F. U. (2013). Molecular Chaperone Functions in Protein Folding and Proteostasis. *Annual Review of Biochemistry*, *82*(1), 323–355. https://doi.org/10.1146/annurev-biochem-060208-092442

9. Hari Sankar, S., Jayadev, K., Suraj, B., & Aparna, P. A COMPREHENSIVE SOLUTION TO ROAD TRAFFIC ACCIDENT DETECTION AND AMBULANCE MANAGEMENT.

10. Li, S., Park, Y., Duraisingham, S., Strobel, F. H., Khan, N., Soltow, Q. A., Jones, D. P., & Pulendran, B. (2013). Predicting Network Activity from High Throughput Metabolomics. *PLOS Computational Biology/PLoS Computational Biology*, *9*(7), e1003123. https://doi.org/10.1371/journal.pcbi.1003123

11. Liu, N. P., Hemani, A., & Paul, K. (2011). *A Reconfigurable Processor for Phylogenetic Inference*. https://doi.org/10.1109/vlsid.2011.74

12. Liu, P., Ebrahim, F. O., Hemani, A., & Paul, K. (2011). *A Coarse-Grained Reconfigurable Processor for Sequencing and Phylogenetic Algorithms in Bioinformatics*. https://doi.org/10.1109/reconfig.2011.1

13. Majumder, T., Pande, P. P., & Kalyanaraman, A. (2014). Hardware Accelerators in Computational Biology: Application, Potential, and Challenges. *IEEE Design & Test*, *31*(1), 8–18. https://doi.org/10.1109/mdat.2013.2290118

14. Majumder, T., Pande, P. P., & Kalyanaraman, A. (2015). On-Chip Network-Enabled Many-Core Architectures for Computational Biology Applications. *Design, Automation &Amp; Test in Europe Conference &Amp; Exhibition (DATE), 2015*. https://doi.org/10.7873/date.2015.1128

15. Özdemir, B. C., Pentcheva-Hoang, T., Carstens, J. L., Zheng, X., Wu, C. C., Simpson, T. R., Laklai, H., Sugimoto, H., Kahlert, C., Novitskiy, S. V., De Jesus-Acosta, A., Sharma, P., Heidari, P., Mahmood, U., Chin, L., Moses, H. L., Weaver, V. M., Maitra, A., Allison, J. P., . . . Kalluri,

R. (2014). Depletion of Carcinoma-Associated Fibroblasts and Fibrosis Induces Immunosuppression and Accelerates Pancreas Cancer with Reduced Survival. *Cancer Cell*, *25*(6), 719–734. https://doi.org/10.1016/j.ccr.2014.04.005

16. Qiu, Z., Cheng, Q., Song, J., Tang, Y., & Ma, C. (2016). Application of Machine Learning-Based Classification to Genomic Selection and Performance Improvement. In *Lecture notes in computer science* (pp. 412–421). https://doi.org/10.1007/978-3-319-42291-6_41

17. Singh, A., Ganapathysubramanian, B., Singh, A. K., & Sarkar, S. (2016). Machine Learning for High-Throughput Stress Phenotyping in Plants. *Trends in Plant Science*, *21*(2), 110–124. https://doi.org/10.1016/j.tplants.2015.10.015

18. Stamatakis, A., Ott, M., & Ludwig, T. (2005). RAxML-OMP: An Efficient Program for Phylogenetic Inference on SMPs. In *Lecture notes in computer science* (pp. 288–302). https://doi.org/10.1007/11535294_25

19. Wang, L., Gu, Q., Zheng, X., Ye, J., Liu, Z., Li, J., Hu, X., Hagler, A., & Xu, J. (2013). Discovery of New Selective Human Aldose Reductase Inhibitors through Virtual Screening Multiple Binding Pocket Conformations. *Journal of Chemical Information and Modeling*, *53*(9), 2409–2422. https://doi.org/10.1021/ci400322j

20. Zheng, J. X., Li, Y., Ding, Y. H., Liu, J. J., Zhang, M. J., Dong, M. Q., Wang, H. W., & Yu, L. (2017). Architecture of the ATG2B-WDR45 complex and an aromatic Y/HF motif crucial for complex formation. *Autophagy*, *13*(11), 1870–1883. https://doi.org/10.1080/15548627.2017.1359381

21. Yang, J., Gupta, V., Carroll, K. S., & Liebler, D. C. (2014). Site-specific mapping and quantification of protein S-sulphenylation in cells. *Nature Communications, 5*(1). https://doi.org/10.1038/ncomms5776