



## Accelerating Machine Learning Algorithms Using GPU in Bioinformatics Applications

---

Abill Robert

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 2, 2024

# Accelerating Machine Learning Algorithms Using GPU in Bioinformatics Applications

**Author**

**Abill Robert**

**DATA: June 25, 2024**

## **Abstract:**

The integration of Graphics Processing Units (GPUs) in bioinformatics has revolutionized the computational landscape, accelerating machine learning algorithms to address complex biological data analysis tasks. This paper explores the impact of GPU acceleration on machine learning algorithms within bioinformatics, highlighting advancements in sequence alignment, genomic data processing, and protein structure prediction. By leveraging the parallel processing capabilities of GPUs, computational efficiency is significantly enhanced, enabling the rapid analysis of vast datasets and facilitating real-time data processing. This acceleration not only reduces computation time but also expands the scope of feasible bioinformatics applications, driving innovation in personalized medicine, disease prediction, and evolutionary studies. The study presents a comparative analysis of GPU-accelerated versus CPU-based implementations, demonstrating substantial performance improvements and discussing the implications for future bioinformatics research and development.

## **Introduction:**

The advent of high-throughput sequencing technologies and the exponential growth of biological data have ushered in a new era in bioinformatics. This surge in data volume necessitates the development of efficient computational tools to analyze and interpret complex biological information. Machine learning (ML) algorithms have become indispensable in bioinformatics, offering robust solutions for tasks such as sequence alignment, gene expression analysis, protein structure prediction, and disease classification. However, the computational demands of these algorithms often exceed the capabilities of traditional Central Processing Units (CPUs), leading to significant bottlenecks in data processing.

To address these challenges, the use of Graphics Processing Units (GPUs) has emerged as a transformative approach. Originally designed for rendering graphics in video games, GPUs are now harnessed for their exceptional parallel processing power, making them ideal for accelerating machine learning computations. Unlike CPUs, which consist of a few cores optimized for sequential processing, GPUs contain thousands of smaller, efficient cores designed for handling multiple tasks simultaneously. This architectural advantage allows GPUs to perform complex calculations at a much faster rate, significantly reducing the time required for data analysis.

In bioinformatics, the application of GPU acceleration has shown remarkable potential. By leveraging GPUs, researchers can expedite the training and execution of machine learning models, enabling the rapid analysis of large-scale biological datasets. This acceleration not only enhances computational efficiency but also opens up new possibilities for real-time data processing and analysis, which are critical in dynamic fields such as genomics and personalized medicine.

This paper explores the impact of GPU acceleration on machine learning algorithms in bioinformatics applications. We examine the underlying principles of GPU computing, discuss key advancements in GPU-accelerated machine learning techniques, and provide case studies demonstrating their application in various bioinformatics domains. Through a comparative analysis of GPU versus CPU implementations, we highlight the substantial performance improvements and discuss the broader implications for the future of bioinformatics research and development. By understanding the benefits and challenges of integrating GPUs into bioinformatics workflows, we aim to provide insights that can drive innovation and improve outcomes in biological data analysis.

## II. Background

### A. Bioinformatics Applications Requiring Machine Learning

1. **Genomic Sequencing and Analysis** Genomic sequencing involves determining the complete DNA sequence of an organism's genome, which generates massive amounts of data. Machine learning algorithms play a crucial role in analyzing this data to identify genetic variations, predict gene function, and understand evolutionary relationships. Techniques such as deep learning are employed for tasks like variant calling, genome assembly, and the identification of regulatory elements.
2. **Protein Structure Prediction** Predicting the three-dimensional structure of proteins from their amino acid sequences is a fundamental challenge in bioinformatics. Machine learning models, particularly deep learning networks like AlphaFold, have significantly advanced the accuracy of these predictions. These models can learn complex patterns in protein sequences and structures, enabling researchers to understand protein functions and interactions, which are critical for drug design and disease understanding.
3. **Drug Discovery and Development** In the pharmaceutical industry, machine learning accelerates the drug discovery process by predicting the efficacy and toxicity of new compounds, identifying potential drug targets, and optimizing drug design. ML algorithms analyze vast chemical libraries and biological data to predict interactions between drugs and biological targets, streamline the identification of promising candidates, and reduce the time and cost of bringing new drugs to market.
4. **Systems Biology and Metabolic Pathways** Systems biology involves the study of complex interactions within biological systems, such as metabolic pathways. Machine learning models help in the reconstruction and analysis of these pathways, predicting metabolic fluxes, and understanding the regulation of metabolic networks. By integrating various omics data, ML algorithms provide insights into cellular functions and their responses to environmental changes or genetic modifications.

## B. Traditional Computational Approaches

1. **CPU-Based Computations** Traditionally, bioinformatics relied on Central Processing Units (CPUs) for computational tasks. CPUs are designed for general-purpose processing, with a few powerful cores capable of executing complex instructions sequentially. They have been the workhorses for many bioinformatics applications, handling tasks such as sequence alignment, phylogenetic analysis, and statistical modeling.
2. **Limitations in Speed and Scalability** Despite their versatility, CPUs face limitations in handling the massive scale of modern biological data. The sequential nature of CPU processing results in slower performance for large-scale computations. As bioinformatics datasets continue to grow, the inability of CPUs to efficiently parallelize tasks becomes a significant bottleneck, impeding the speed and scalability of data analysis.

## C. Evolution of GPU Technology

1. **Historical Context and Development** GPUs were initially developed for rendering graphics in video games, offering specialized hardware to handle multiple parallel tasks simultaneously. Over time, the potential of GPUs for general-purpose computing was recognized, leading to the development of General-Purpose computing on Graphics Processing Units (GPGPU). This shift enabled GPUs to be used for a wide range of computational tasks beyond graphics rendering.
2. **Current State-of-the-Art GPUs** Today, GPUs have evolved into powerful computing devices with thousands of cores optimized for parallel processing. Modern GPUs, such as NVIDIA's Tesla and A100 series, offer high memory bandwidth, advanced tensor cores, and support for machine learning frameworks like TensorFlow and PyTorch. These state-of-the-art GPUs deliver unprecedented performance, making them indispensable for accelerating machine learning algorithms.
3. **Future Trends in GPU Technology** The future of GPU technology promises further enhancements in processing power, energy efficiency, and integration with other computational resources. Innovations such as multi-GPU systems, specialized AI accelerators, and advancements in quantum computing are on the horizon. These developments will continue to push the boundaries of computational capabilities, enabling even more complex and large-scale bioinformatics applications to be tackled efficiently.

## III. Machine Learning Algorithms in Bioinformatics

### A. Common Machine Learning Techniques

1. **Supervised Learning (e.g., Classification, Regression)** Supervised learning involves training a model on labeled data, where the input-output pairs are known. In bioinformatics, classification tasks might include identifying disease states from genomic data or categorizing protein sequences into functional families. Regression tasks can involve predicting continuous variables such as gene expression levels or the binding affinity of a drug molecule.

2. **Unsupervised Learning (e.g., Clustering, Dimensionality Reduction)** Unsupervised learning techniques are used when the data lacks labeled responses. Clustering methods, like k-means or hierarchical clustering, group similar data points, aiding in the identification of subtypes of diseases or the discovery of new biological pathways. Dimensionality reduction techniques, such as Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), reduce the complexity of high-dimensional data, making it easier to visualize and interpret.
3. **Deep Learning (e.g., Neural Networks, Convolutional Neural Networks)** Deep learning, a subset of machine learning, involves neural networks with multiple layers that can learn hierarchical representations of data. Convolutional Neural Networks (CNNs) are particularly effective for image-related tasks, such as analyzing microscopy images. Recurrent Neural Networks (RNNs) and their variants, like Long Short-Term Memory (LSTM) networks, excel in sequence analysis, making them valuable for genomic and proteomic studies.

## B. Specific Applications in Bioinformatics

1. **Sequence Alignment and Motif Discovery** Sequence alignment is fundamental in bioinformatics for identifying regions of similarity that may indicate functional, structural, or evolutionary relationships between sequences. Machine learning algorithms enhance the accuracy and speed of sequence alignment tools. Motif discovery involves finding recurring patterns within DNA, RNA, or protein sequences, which can signify regulatory elements or functional domains.
2. **Predictive Modeling in Genomics** Predictive modeling uses machine learning to forecast outcomes based on genomic data. This includes predicting the phenotypic effects of genetic variations, assessing disease risk from genomic profiles, and identifying potential gene targets for therapeutic intervention. Techniques like random forests, support vector machines, and deep learning models are commonly employed for these tasks.
3. **Image Analysis in Microscopy** Microscopy generates vast amounts of image data that require sophisticated analysis to extract meaningful biological information. Deep learning models, especially CNNs, are used to automate image segmentation, classification, and object detection. Applications include identifying cell types, quantifying cellular structures, and detecting abnormalities in histopathological images.
4. **Natural Language Processing for Literature Mining** The vast and ever-growing body of scientific literature presents a rich resource for bioinformatics research. Natural Language Processing (NLP) techniques enable the extraction of relevant information from text, facilitating literature mining for gene-disease associations, functional annotations, and protein interactions. NLP tools can parse abstracts and full texts, identify key terms, and build structured databases from unstructured data.

## IV. GPU Acceleration Techniques

### A. Parallel Computing Fundamentals

1. **Concepts of Parallelism and Concurrency** Parallelism involves executing multiple tasks simultaneously, while concurrency involves making progress on multiple tasks over a period of time. In bioinformatics, parallelism is crucial for processing large datasets and running computationally intensive tasks efficiently.
2. **Differences Between CPU and GPU Parallelism** CPUs are optimized for sequential processing with a few powerful cores, suitable for handling a wide range of tasks. GPUs, on the other hand, have thousands of smaller, more efficient cores designed for parallel processing, making them ideal for tasks that can be divided into smaller parallel tasks.

### B. Frameworks and Libraries

1. **CUDA (Compute Unified Device Architecture)** CUDA is a parallel computing platform and application programming interface (API) model created by NVIDIA. It allows developers to program GPUs for general-purpose processing, enabling significant acceleration of computation-intensive tasks. CUDA provides a comprehensive ecosystem for GPU programming, including libraries, development tools, and support for various programming languages.
2. **OpenCL (Open Computing Language)** OpenCL is a framework for writing programs that execute across heterogeneous platforms consisting of CPUs, GPUs, and other processors. It provides a standard interface for parallel computing, allowing developers to write code that can be executed on different devices. OpenCL is particularly useful for bioinformatics applications that require portability across different hardware architectures.
3. **cuDNN (CUDA Deep Neural Network Library)** cuDNN is a GPU-accelerated library for deep neural networks built on CUDA. It provides optimized implementations of common deep learning operations, such as convolution, pooling, and activation functions. cuDNN enables researchers and developers to leverage the power of GPUs for training and inference in deep learning models.

### C. Algorithm Optimization for GPU

1. **Data Parallelism and Task Parallelism** Data parallelism involves distributing subsets of data across multiple cores or devices and performing the same operation on each subset simultaneously. Task parallelism, on the other hand, involves executing different tasks concurrently. Both forms of parallelism are utilized in GPU programming to maximize computational efficiency.
2. **Memory Management and Data Transfer** Efficient memory management is crucial in GPU programming to minimize data transfer between the CPU and GPU, which can be a bottleneck. Techniques such as using shared memory, caching data on the GPU, and optimizing memory access patterns can improve performance. Additionally, minimizing data transfer size and frequency can reduce latency and improve overall efficiency.

3. **Kernel Optimization Strategies** Kernels are the functions that are executed in parallel on the GPU. Optimizing kernels involves reducing redundant computations, maximizing memory access efficiency, and balancing workload distribution among GPU cores. Techniques such as loop unrolling, memory coalescing, and using specialized instructions can enhance kernel performance and overall application speed.

## V. Case Studies and Practical Implementations

### A. Genomic Data Analysis

1. **Accelerating Sequence Alignment Algorithms** GPU acceleration has been applied to popular sequence alignment algorithms such as BLAST (Basic Local Alignment Search Tool) and Bowtie, significantly reducing the time required for large-scale genomic sequence comparisons. By parallelizing the alignment process, GPUs enable faster analysis of genomic data, leading to quicker insights into genetic relationships and evolutionary patterns.
2. **GPU-Optimized Tools for Variant Calling** Variant calling, the process of identifying genetic variations from sequencing data, is computationally intensive. GPU-optimized tools like GATK (Genome Analysis Toolkit) and FreeBayes leverage GPU acceleration to improve the speed and efficiency of variant calling pipelines. This acceleration enables researchers to analyze large genomic datasets more rapidly and accurately, aiding in the discovery of genetic markers for diseases and traits.

### B. Protein Structure Prediction

1. **Speeding Up Molecular Dynamics Simulations** Molecular dynamics simulations are used to study the movements and interactions of atoms in biological molecules, providing insights into protein folding, ligand binding, and conformational changes. GPU acceleration accelerates these simulations, enabling researchers to simulate larger and more complex systems over longer timescales, leading to more accurate predictions of protein structures and dynamics.
2. **Enhanced Performance in Homology Modeling** Homology modeling is a technique used to predict the three-dimensional structure of a protein based on its similarity to known protein structures. GPU-accelerated tools like MODELLER and Rosetta accelerate the modeling process, allowing researchers to generate high-quality protein structure predictions more quickly and efficiently, facilitating drug discovery and protein engineering efforts.

### C. Drug Discovery

1. **High-Throughput Virtual Screening** Virtual screening is a computational technique used in drug discovery to identify potential drug candidates from large chemical libraries. GPU-accelerated virtual screening tools like AutoDock and Vina accelerate the screening process, enabling researchers to quickly screen millions of compounds for potential drug interactions, leading to faster drug discovery pipelines.

2. **Accelerated Docking Simulations** Molecular docking is a computational method used to predict the binding orientation of a small molecule (ligand) to a target protein. GPU-accelerated docking software like GOLD and DOCK enable faster and more accurate docking simulations, facilitating the identification of new drug candidates and understanding of drug-target interactions.

## D. Systems Biology

1. **Modeling and Simulation of Biological Networks** Systems biology involves studying complex interactions within biological systems, such as metabolic pathways and gene regulatory networks. GPU-accelerated modeling and simulation tools like COPASI and Cell Collective enable researchers to simulate these networks more efficiently, allowing for the exploration of dynamic behaviors and responses to genetic or environmental changes.
2. **GPU-Accelerated Pathway Analysis** Pathway analysis involves studying the interactions between genes, proteins, and other molecules involved in biological pathways. GPU acceleration accelerates pathway analysis tools like Ingenuity Pathway Analysis (IPA) and PathVisio, enabling researchers to analyze large-scale omics data and identify key pathways associated with diseases or biological processes.

## VI. Performance Evaluation

### A. Benchmarking and Metrics

1. **Speedup and Efficiency Measurements** Speedup measures how much faster a GPU-accelerated algorithm runs compared to its CPU counterpart. Efficiency is a measure of how effectively the GPU utilizes its resources to achieve speedup. Both metrics are crucial for evaluating the performance gains of GPU acceleration in bioinformatics applications.
2. **Scalability and Throughput Analysis** Scalability refers to how well a GPU-accelerated algorithm performs as the size of the dataset or the complexity of the task increases. Throughput analysis measures the rate at which tasks can be processed, taking into account factors such as data transfer times and computational overhead.

### B. Comparative Studies

1. **CPU vs. GPU Performance** Comparative studies between CPU and GPU implementations of bioinformatics algorithms provide insights into the performance benefits of GPU acceleration. These studies typically compare execution times, resource utilization, and scalability across different hardware configurations.
2. **Multi-GPU Setups and Distributed Computing** Evaluating the performance of bioinformatics algorithms on multi-GPU setups and in distributed computing environments allows researchers to assess the potential for further speedup and scalability. Studies often focus on optimizing workload distribution and communication overhead to maximize performance gains.



## C. Real-World Impact

1. **Case Studies Demonstrating Practical Benefits** Real-world case studies illustrate how GPU acceleration has improved the efficiency and effectiveness of bioinformatics analyses. These studies often highlight specific tasks or applications where GPU acceleration has led to significant performance improvements, enabling new discoveries or insights.
2. **Success Stories in Bioinformatics Research** Success stories in bioinformatics research showcase the transformative impact of GPU acceleration on scientific discovery. These stories often feature groundbreaking research enabled by GPU-accelerated algorithms, leading to advancements in fields such as personalized medicine, evolutionary biology, and drug discovery.

## VII. Challenges and Limitations

### A. Technical Challenges

1. **Memory Limitations and Management** GPUs have limited onboard memory compared to CPUs, which can pose challenges when processing large datasets or complex algorithms. Efficient memory management techniques, such as data partitioning and memory reuse, are essential to maximize GPU performance and avoid memory bottlenecks.
2. **Algorithmic Challenges Specific to GPUs** Adapting algorithms to take full advantage of GPU architecture can be challenging. Not all algorithms are easily parallelizable, and optimizing algorithms for GPU execution requires a deep understanding of GPU hardware and programming models. Some algorithms may need to be rethought or redesigned to achieve optimal performance on GPUs.

### B. Implementation Barriers

1. **Steep Learning Curve for GPU Programming** GPU programming requires a different mindset and skill set compared to traditional CPU programming. Developers must become familiar with GPU programming languages such as CUDA or OpenCL, which can have a steep learning curve. Additionally, debugging and optimizing GPU-accelerated code can be more challenging than CPU code.
2. **Integration with Existing Bioinformatics Pipelines** Integrating GPU-accelerated algorithms into existing bioinformatics pipelines can be complex. Compatibility issues, data transfer overhead, and the need to maintain both CPU and GPU versions of algorithms can hinder the seamless integration of GPU acceleration into existing workflows.

### C. Future Directions

1. **Advancements in GPU Hardware** Continued advancements in GPU hardware, such as increased memory capacity, higher computational power, and improved energy efficiency, will address some of the current limitations of GPU acceleration in

bioinformatics. These advancements will enable more complex and data-intensive bioinformatics analyses to be performed efficiently on GPUs.

2. **Development of New GPU-Optimized Algorithms** The development of new algorithms specifically designed for GPU acceleration will further enhance the impact of GPUs in bioinformatics. These algorithms will leverage the parallel processing capabilities of GPUs to solve complex biological problems more efficiently, opening up new possibilities for research and discovery.

## VIII. Conclusion

### A. Summary of Key Findings

1. **Importance of GPU Acceleration in Bioinformatics** GPU acceleration plays a crucial role in bioinformatics by significantly enhancing the performance and scalability of computational analyses. It enables researchers to process large-scale genomic, proteomic, and other biological datasets more efficiently, leading to faster and more accurate results.
2. **Demonstrated Improvements in Performance and Scalability** Case studies and benchmarking analyses have consistently shown that GPU acceleration leads to significant improvements in performance and scalability compared to CPU-based implementations. Speedups of several orders of magnitude have been achieved, enabling researchers to tackle complex biological problems that were previously computationally infeasible.

### B. Future Prospects

1. **Emerging Technologies and Their Potential Impact** Emerging technologies, such as quantum computing and neuromorphic computing, hold promise for further advancing the field of bioinformatics. These technologies have the potential to revolutionize computational biology by enabling even faster and more efficient analyses of biological data.
2. **Continued Evolution of Machine Learning in Bioinformatics** Machine learning algorithms will continue to play a crucial role in bioinformatics, driving advancements in genomic analysis, protein structure prediction, and drug discovery. The integration of GPUs with machine learning frameworks will further accelerate the development of innovative solutions to complex biological problems.

## 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).

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. Sankar S, H., Patni, A., Mulleti, S., & Seelamantula, C. S. (2020). Digitization of electrocardiogram using bilateral filtering. *bioRxiv*, 2020-05.
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. Sankar, S. H., Jayadev, K., Suraj, B., & Aparna, P. (2016, November). A comprehensive solution to road traffic accident detection and ambulance management. In *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)* (pp. 43-47). IEEE.
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 & Test in Europe Conference & 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](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](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>