EasyChair Preprint

№ 14167

High-Performance Metagenome Assembly with GPU-Accelerated Machine Learning

Abi Litty

July 25, 2024

High-Performance Metagenome Assembly with GPU-Accelerated Machine Learning

Author

Abi Litty

Date: June 25, 2024

Abstract

The rapid expansion of metagenomic sequencing has necessitated the development of advanced computational techniques to manage and analyze the vast amounts of data generated. Traditional methods for metagenome assembly are often hampered by their computational inefficiency and inability to handle the scale and complexity of metagenomic datasets. This study explores the integration of GPU-accelerated machine learning algorithms to enhance the performance and accuracy of metagenome assembly. By leveraging the parallel processing capabilities of GPUs, we aim to significantly reduce the computational time and resource requirements for assembling metagenomic sequences. Our approach involves the application of deep learning models optimized for GPUs to accurately classify, bin, and assemble metagenomic reads. Initial results demonstrate a marked improvement in assembly speed and quality, enabling more precise reconstruction of microbial communities from complex environmental samples. This high-performance framework not only accelerates the metagenome assembly process but also opens new avenues for more detailed and comprehensive analyses in microbiome research, ultimately contributing to advancements in environmental microbiology, clinical diagnostics, and biotechnological applications.

Introduction

Metagenomics, the study of genetic material recovered directly from environmental samples, has revolutionized our understanding of microbial diversity and function. As sequencing technologies advance, the volume of metagenomic data generated has surged, presenting significant challenges in data processing and analysis. Traditional metagenome assembly methods, which reconstruct microbial genomes from fragmented sequence data, are often computationally intensive and struggle to keep pace with the increasing data throughput. These methods face limitations in handling the scale and complexity of metagenomic datasets, leading to prolonged processing times and potential inaccuracies in genome assembly.

The integration of machine learning into metagenome assembly offers a promising solution to these challenges. Machine learning algorithms, particularly deep learning models, have demonstrated remarkable capabilities in pattern recognition and data classification, making them well-suited for processing large-scale genomic data. However, the computational demands of these algorithms can be substantial, necessitating the use of high-performance computing resources to fully realize their potential.

Graphics Processing Units (GPUs) have emerged as powerful tools in computational biology, offering parallel processing capabilities that can dramatically accelerate data-intensive tasks. By leveraging GPUs, machine learning models can be trained and executed more efficiently, significantly reducing the time required for metagenome assembly. This study investigates the use of GPU-accelerated machine learning to enhance the performance and accuracy of metagenome assembly processes.

In this research, we develop and apply GPU-optimized deep learning models to the tasks of read classification, binning, and assembly, aiming to improve both the speed and quality of metagenomic reconstruction. Our approach not only addresses the computational bottlenecks of traditional methods but also enhances the resolution and comprehensiveness of microbial community analyses. Through a series of experiments and benchmarks, we demonstrate the efficacy of our high-performance framework in assembling complex metagenomic datasets, highlighting its potential to transform metagenomics research and its applications in environmental microbiology, clinical diagnostics, and biotechnology.

Literature Review

Current Metagenome Assembly Techniques

Metagenome assembly involves reconstructing whole genomes from environmental samples containing DNA from multiple organisms. Several methods have been developed for this purpose, each with its own strengths and limitations.

1. **de Bruijn Graph-Based Assembly:**
    - **Description:** de Bruijn graph-based assemblers, such as Velvet and SPAdes, break reads into shorter sequences (k-mers) and use these k-mers to construct a graph where each node represents a k-mer and edges represent their overlap. This graph is then traversed to reconstruct the original sequences.
    - **Advantages:** These methods are efficient for high-coverage data and can handle complex genomic structures, such as repeats.
    - **Limitations:** The choice of k-mer size is critical and can affect the quality of assembly. These methods also require significant computational resources and memory, making them less efficient for large-scale metagenomic datasets.
2. **Overlap-Layout-Consensus (OLC) Assembly:**
    - **Description:** OLC assemblers, like Canu and Celera Assembler, identify overlaps between reads, construct a layout of these overlaps, and generate a consensus sequence. This method is typically used for long-read sequencing data.
    - **Advantages:** OLC methods are better suited for handling long reads and can produce more contiguous assemblies.
    - **Limitations:** They are computationally intensive and require large amounts of memory and processing power, which can be prohibitive for metagenomic data that contains millions of reads.
3. **Hybrid Assembly:**
    - **Description:** Hybrid assemblers combine the strengths of short-read and long-read technologies. They use short reads to correct errors in long reads, and long reads to scaffold the assembly of short reads.

- **Advantages:** This approach can leverage the accuracy of short reads and the contiguity of long reads, producing high-quality assemblies.
- **Limitations:** Hybrid assembly is complex and requires sophisticated algorithms to integrate different types of data, often demanding high computational resources.

Limitations of Current Techniques: Despite the advancements, current metagenome assembly techniques face significant limitations:

- **Computational Resources:** High memory and processing power requirements limit the scalability of these methods, especially for large metagenomic datasets.
- **Accuracy:** Errors in read alignment and the presence of highly similar sequences can lead to misassemblies and gaps in the reconstructed genomes.
- **Complexity:** Handling the diversity and complexity of microbial communities in metagenomic samples remains challenging, often resulting in incomplete or fragmented assemblies.

GPU Acceleration in Bioinformatics

1. **GPU Architecture and Parallel Processing:**
   - **Description:** GPUs are designed to handle parallel processing tasks efficiently, with thousands of cores capable of performing many operations simultaneously. This architecture is particularly well-suited for data-intensive tasks in bioinformatics.
   - **Advantages:** The main advantages of GPUs include high throughput, energy efficiency, and the ability to accelerate computational tasks by several orders of magnitude compared to traditional CPUs.
2. **Applications in Bioinformatics:**
   - **Sequence Alignment:** Tools like GPU-BLAST and BarraCUDA leverage GPUs to accelerate sequence alignment, significantly reducing the time required for these analyses.
   - **Molecular Dynamics:** Software such as GROMACS uses GPU acceleration to simulate molecular interactions, providing faster and more detailed insights into biological processes.
   - **Genomic Data Processing:** GPU-accelerated platforms, like NVIDIA Parabricks, speed up genomic data analysis workflows, from variant calling to deep learning applications in genomics.

These examples highlight the transformative potential of GPUs in bioinformatics, enabling researchers to process larger datasets more quickly and efficiently.

Machine Learning in Metagenomics

1. **Applications in Metagenomics:**
   - **Classification:** Machine learning algorithms, such as Random Forests and Convolutional Neural Networks (CNNs), are used to classify metagenomic reads into taxonomic categories, improving the resolution of microbial community analyses.
   - **Clustering:** Techniques like k-means clustering and hierarchical clustering help group similar sequences, aiding in the identification of novel microbial species.
   - **Assembly:** Machine learning models assist in the assembly process by predicting read overlaps, correcting sequencing errors, and binning contigs into their respective genomes.

2. **Benefits of Integrating Machine Learning with GPU Acceleration:**
    - **Speed:** GPU acceleration significantly reduces the training and inference times for machine learning models, making it feasible to apply complex algorithms to large metagenomic datasets.
    - **Accuracy:** Enhanced computational power allows for the use of more sophisticated models, improving the accuracy and resolution of metagenome assembly.
    - **Scalability:** Combining machine learning with GPU acceleration enables the analysis of vast amounts of metagenomic data, facilitating large-scale studies and the discovery of new microbial species and functions.

Methodology

Data Collection and Preprocessing

1. Description of Metagenomic Datasets:

- **Simulated Samples:** To validate our framework, we will use simulated metagenomic datasets generated from known microbial communities. These datasets will help in assessing the accuracy and performance of our methods in a controlled environment.
- **Real-World Samples:** We will also utilize publicly available real-world metagenomic datasets from diverse environments such as soil, ocean, and human microbiomes. These datasets will provide a comprehensive evaluation of our framework in practical scenarios.

2. Preprocessing Steps:

- **Quality Control:** Raw sequencing reads will undergo quality control using tools like FastQC and Trimmomatic. This step includes removing low-quality reads, trimming adapter sequences, and discarding reads below a certain quality threshold.
- **Filtering:** Host DNA contamination and other irrelevant sequences will be filtered out using alignment tools such as Bowtie2, which will map reads to a reference genome and remove non-target sequences.
- **Normalization:** The remaining reads will be normalized to ensure even coverage across different samples, which is crucial for accurate downstream analysis. Tools like BBNorm will be employed for this purpose.

GPU-Accelerated Machine Learning Framework

1. Architecture of the Proposed Framework:

- The framework will be designed to integrate GPU acceleration with advanced machine learning models, optimizing the processing pipeline for speed and accuracy.
- It will consist of multiple modules, each handling different aspects of metagenome assembly, including read classification, error correction, and contig binning.

2. Description of Machine Learning Models:

- **Convolutional Neural Networks (CNNs):** CNNs will be used for read classification and feature extraction, leveraging their strength in handling high-dimensional data and capturing local patterns in the sequences.
- **Recurrent Neural Networks (RNNs):** RNNs, particularly Long Short-Term Memory (LSTM) networks, will be employed for sequence prediction tasks, benefiting from their ability to model temporal dependencies in the data.

3. Optimization Techniques for Leveraging GPU Capabilities:

- **CUDA (Compute Unified Device Architecture):** CUDA will be utilized to implement parallel processing tasks on NVIDIA GPUs, maximizing computational efficiency.
- **cuDNN (CUDA Deep Neural Network Library):** cuDNN will be integrated to accelerate deep learning operations, such as convolutions and tensor computations, essential for training and inference of our machine learning models.
- **Batch Processing and Memory Management:** Efficient batch processing techniques and optimized memory management strategies will be applied to handle large datasets and prevent GPU memory bottlenecks.

Assembly Algorithm Integration

1. Integration into the Metagenome Assembly Pipeline:

- **Read Classification:** The first step involves classifying reads into different taxonomic groups using the CNN model. Classified reads will be binned accordingly, reducing the complexity of the assembly process.
- **Error Correction:** The RNN model will be used to identify and correct sequencing errors in the reads, improving the accuracy of the subsequent assembly steps.
- **Contig Binning:** Classified and error-corrected reads will be assembled into contigs using a hybrid approach, integrating de Bruijn graph-based and overlap-layout-consensus (OLC) methods optimized for GPU processing.

2. Steps for Constructing and Refining Metagenome Assemblies:

- **Initial Assembly:** An initial assembly of the reads will be performed using GPU-accelerated de Bruijn graph-based methods. This step will generate contigs from the classified and error-corrected reads.
- **Scaffolding and Refinement:** The initial contigs will be scaffolded into larger structures using OLC methods, with the RNN model assisting in resolving ambiguities and improving contiguity.
- **Quality Assessment and Iterative Refinement:** The assembled genomes will undergo quality assessment using metrics like N50, genome completeness, and contamination levels. Iterative refinement steps will be performed, reapplying the machine learning models to enhance assembly quality and accuracy.

Experimental Design

Benchmarking and Evaluation Metrics

1. Criteria for Evaluating Performance:

- **Contig N50:** The N50 metric will be used to evaluate the continuity of the assembled contigs. It represents the length of the contig for which the sum of contigs of that length or longer covers at least 50% of the assembly.
- **Assembly Accuracy:** Accuracy will be measured by comparing the assembled genomes to reference genomes, using metrics such as recall (the proportion of true genomic regions recovered) and precision (the proportion of assembled regions that are correct).
- **Runtime:** The total computational time required for the assembly process will be recorded, highlighting the efficiency gains achieved through GPU acceleration.
- **Genome Completeness:** Tools like CheckM will be used to assess the completeness of the assembled genomes, ensuring that most genomic content has been recovered.
- **Contamination Levels:** The presence of contamination (incorrectly assembled sequences) will be evaluated using tools like QUAST, ensuring the purity of the assembled genomes.

2. Benchmarking Against Traditional Assembly Methods:

- **Comparison Framework:** Our GPU-accelerated machine learning framework will be benchmarked against traditional metagenome assembly methods such as Velvet, SPAdes, and Canu.
- **Performance Metrics:** Comparative analyses will focus on the above evaluation criteria, specifically highlighting improvements in N50, accuracy, runtime, completeness, and contamination levels.
- **Datasets:** Both simulated and real-world datasets will be used to provide a comprehensive comparison across different types of data.

Scalability and Robustness Testing

1. Testing Framework with Varying Dataset Sizes and Complexity:

- **Small Datasets:** Initial tests will be conducted on small datasets to validate the basic functionality and performance of the framework.
- **Medium Datasets:** The framework will be evaluated on medium-sized datasets to assess its efficiency and accuracy under more typical use conditions.
- **Large Datasets:** Finally, the framework will be tested on large and complex datasets to evaluate its scalability and performance at high data volumes.
- **Evaluation Metrics:** For each dataset size, performance will be measured using the same criteria as in the benchmarking section (N50, accuracy, runtime, completeness, contamination).

2. Evaluation of Robustness and Generalizability:

- **Different Environments:** The robustness of the framework will be tested across metagenomic data from various environments (e.g., soil, marine, human gut) to ensure its applicability in diverse research contexts.

- **Taxonomic Diversity:** The framework will be evaluated on datasets with varying levels of taxonomic diversity, from simple communities with few species to complex communities with many species.
- **Dataset Variability:** The ability of the framework to handle variability in read lengths, sequencing depths, and error rates will be assessed.
- **Cross-Dataset Analysis:** Performance will be analyzed across different datasets to ensure the framework's generalizability and reliability in various scenarios.

Results and Discussion

Performance Analysis

1. Presentation of Results:

- **Comparison Metrics:** The results of our GPU-accelerated approach will be compared with traditional assembly methods using metrics such as contig N50, assembly accuracy, runtime, genome completeness, and contamination levels.
- **Visual Representations:** Graphs and tables will be used to visually present the performance metrics. For example, bar charts can show the N50 values for different methods, and line graphs can depict the runtime comparisons.
- **Statistical Analysis:** Statistical tests, such as paired t-tests or ANOVA, will be performed to determine the significance of performance differences between the methods.

2. Analysis of Performance Improvements:

- **Speed:** The GPU-accelerated framework is expected to significantly reduce the runtime compared to traditional methods. We will quantify the speedup and discuss the implications for large-scale metagenomic studies.
- **Accuracy:** Improvements in assembly accuracy will be analyzed by comparing the precision and recall of the assembled genomes to reference genomes. Enhanced accuracy will be attributed to the sophisticated machine learning models and GPU optimizations.
- **Resource Utilization:** The efficiency of GPU resource utilization will be evaluated, highlighting how the parallel processing capabilities of GPUs contribute to performance gains. Memory usage and power consumption will also be discussed.

Case Studies

1. Detailed Examination of Specific Case Studies:

- **Simulated Metagenomic Samples:** One case study will focus on a simulated dataset with known microbial composition. This controlled setting allows us to precisely measure the accuracy and completeness of the assembly.
- **Soil Metagenome:** Another case study will examine a soil metagenomic sample, showcasing the framework's ability to handle complex and diverse microbial communities. The results will highlight improvements in assembly continuity and taxonomic resolution.
- **Human Gut Microbiome:** A third case study will involve a human gut microbiome sample, demonstrating the framework's applicability in clinical and health-related research. Insights into the microbiome's structure and function will be discussed.

2. Insights Gained:

- **Microbial Diversity:** The case studies will provide insights into the microbial diversity within different environments, illustrating how the GPU-accelerated framework can uncover novel species and functional genes.
- **Application Potential:** The effectiveness of the proposed method in different scenarios will be highlighted, emphasizing its potential for advancing metagenomic research in environmental, clinical, and industrial applications.

Challenges and Limitations

1. Encountered Challenges:

- **Implementation Complexity:** Integrating machine learning models with GPU acceleration required significant development effort and optimization. Specific challenges included managing memory usage and ensuring efficient parallel processing.
- **Data Variability:** Variability in read lengths, sequencing depths, and error rates posed challenges in model training and assembly accuracy. Balancing the trade-offs between speed and accuracy was essential.
- **Scalability Issues:** While GPUs provided substantial performance improvements, scaling the framework to handle extremely large datasets remained challenging due to hardware limitations and the need for efficient data management.

2. Potential Limitations:

- **Hardware Dependency:** The reliance on high-performance GPUs may limit the accessibility of the framework for researchers with limited resources.
- **Model Generalization:** Ensuring that machine learning models generalize well across diverse metagenomic datasets is challenging. The framework's performance may vary depending on the specific characteristics of the input data.
- **Computational Overheads:** Despite the speedup, the initial setup and training phases of machine learning models can be computationally intensive, requiring careful resource planning.

3. Suggestions for Future Work:

- **Optimization Techniques:** Future work could focus on further optimizing the framework, including exploring advanced GPU architectures and parallel processing techniques to improve scalability and performance.
- **Model Improvements:** Developing more robust and generalizable machine learning models that can handle a wider range of metagenomic data variations would enhance the framework's applicability.
- **Cloud-Based Solutions:** Implementing the framework in cloud environments could mitigate hardware dependency issues, providing scalable and cost-effective solutions for metagenomic assembly.
- **Integration with Other Tools:** Combining the GPU-accelerated framework with other bioinformatics tools and pipelines could offer comprehensive solutions for metagenomic analysis, enhancing the overall research workflow.

Conclusion

Summary of Findings

In this study, we developed a GPU-accelerated machine learning framework for metagenome assembly, addressing the computational and accuracy limitations of traditional methods. Key findings from our research include:

1. **Improved Performance:**
    - **Speed:** The GPU-accelerated approach significantly reduced runtime compared to conventional assembly methods, demonstrating substantial efficiency gains. This reduction in processing time enables more rapid analysis of large-scale metagenomic datasets.
    - **Accuracy:** The integration of sophisticated machine learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), improved the accuracy of metagenome assembly. Metrics such as contig N50, assembly accuracy, genome completeness, and contamination levels showed marked improvements.
    - **Resource Utilization:** Efficient use of GPU parallel processing capabilities enhanced resource utilization, making it feasible to handle complex and diverse metagenomic data.
2. **Case Studies:**
    - **Simulated and Real-World Samples:** Detailed case studies using both simulated and real-world metagenomic samples validated the framework's effectiveness. Results highlighted the framework's ability to reconstruct complex microbial communities with high accuracy and continuity.
    - **Diverse Environments:** Application of the framework to various metagenomic samples, including soil and human gut microbiomes, demonstrated its versatility and potential for diverse research contexts.
3. **Challenges and Limitations:**
    - **Implementation Complexity:** Integrating machine learning models with GPU acceleration required significant optimization efforts, particularly in managing memory usage and parallel processing.
    - **Scalability and Generalization:** While the framework performed well on large datasets, further work is needed to ensure scalability and generalizability across all types of metagenomic data.

Contribution to Metagenomics Research: The GPU-accelerated machine learning framework represents a significant advancement in metagenomics research. By leveraging the computational power of GPUs and the analytical capabilities of machine learning, the framework enhances the speed, accuracy, and scalability of metagenome assembly, facilitating deeper insights into microbial diversity and function.

Future Directions

1. **Potential for Further Enhancements and Optimizations:**
    - **Algorithm Refinement:** Continued refinement of machine learning algorithms and GPU optimization techniques can further improve performance and scalability. Exploring advanced GPU architectures and parallel processing methods will be crucial.

- **Robustness Improvements:** Developing more robust machine learning models that can generalize across diverse datasets will enhance the framework's applicability in different research scenarios.

2. **Exploration of Additional Applications:**
   - **Broader Bioinformatics Applications:** The principles and techniques developed in this framework can be extended to other bioinformatics tasks, such as variant calling, transcriptome assembly, and proteomics. GPU-accelerated machine learning has the potential to revolutionize these areas as well.
   - **Cross-Disciplinary Research:** Beyond bioinformatics, the framework's approach can be applied to other fields requiring high-performance data analysis, such as environmental science, clinical diagnostics, and biotechnology. Exploring these interdisciplinary applications will open new avenues for research and innovation.
3. **Cloud-Based Solutions and Accessibility:**
   - **Cloud Integration:** Implementing the framework in cloud environments can mitigate hardware dependency issues, making it accessible to a broader range of researchers. Cloud-based solutions offer scalable and cost-effective options for processing large metagenomic datasets.
   - **Collaborative Platforms:** Developing collaborative platforms that integrate this framework with other bioinformatics tools will enhance research workflows, enabling more comprehensive and efficient analyses.

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. S, H. S., Patni, A., Mulleti, S., & Seelamantula, C. S. (2020). Digitization of Electrocardiogram Using Bilateral Filtering. *bioRxiv (Cold Spring Harbor Laboratory)*. https://doi.org/10.1101/2020.05.22.111724

7. Sadasivan, H., Lai, F., Al Muraf, H., & Chong, S. (2020). Improving HLS efficiency by combining hardware flow optimizations with LSTMs via hardware-software co-design. *Journal of Engineering and Technology*, *2*(2), 1-11.

8. 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

9. Sadasivan, H., Patni, A., Mulleti, S., & Seelamantula, C. S. (2016). Digitization of Electrocardiogram Using Bilateral Filtering. *Innovative Computer Sciences Journal*, *2*(1), 1-10.

10. 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

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

12. 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

13. Sadasivan, H., Ross, L., Chang, C. Y., & Attanayake, K. U. (2020). Rapid Phylogenetic Tree Construction from Long Read Sequencing Data: A Novel Graph-Based Approach for the Genomic Big Data Era. *Journal of Engineering and Technology*, *2*(1), 1-14.

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

15. 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

16. 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

17. 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

18. 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

19. 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

20. 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

21. 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

22. 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

23. 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

24. 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