Parallelizing iGEMDOCK – a fundamental virtual screening software for recognizing pharmacological interactions in drug design

Wei-cheng, Chang Zheng-yu, Tong

October, 2023

1 Introduction

Virtual screening is a crucial computational technique in the field of drug design and structural biology, which aids researchers in predicting the binding affinities between various compounds and targeted proteins. It plays an important role in understanding how small molecules, often drug candidates, interact with protein subpockets, which is a fundamental step in identifying potential drug candidates. However, as the demand for faster and more accurate drug discovery processes grows, it is evident that the computational workload associated with virtual screening is a bottleneck. The computational cost of docking millions of potential compounds against a protein target can be relatively high due to complicated physical and chemical properties.

1.1 Motivation

A pair of recently released studies shed new light on the staggering cost of developing new drugs—an expense that now exceeds $2 billion per therapy on average. The average cost of developing a new drug among the top 20 global biopharmas rose 15% ($298 million) last year according to the investigation by the business services consultancy Deloitte. Therefore, developing a parallelized virtual screening software is crucial and valuable in drug design, which can provide useful information for researchers to screen promising drug-target pairs before investing a huge amount of cost in experiments.

2 Statement of the problem

iGEMDOCK provides an integrated environment for virtual screening, including a friendly interface to seamlessly combine different-stage programs for virtual screening and identifying the pharmacological interactions from screening compounds. However, for users who own computers for personal use might take long to process screening due to the limited hardware resources. Therefore, parallelizing iGEMDOCK is valuable to researchers to have better experiences by making the docking process efficient.

3 Proposed approaches

3.1 Input

iGEMDOCK allows users to input an interested protein and its drug candidates, as well as parameters for customization of cavity selection and genetic algorithm. Below are the definitions for the terms in the input component:

- PDB file: Protein Data Bank file, which is a standard file format used in structural biology and bioinformatics to store and exchange three-dimensional structural information of biological macromolecules, such as proteins, nucleic acids, and complexes of these molecules.
- Mol files: Molecular files, which is a common file format used to store information about the structure of molecules, particularly small organic compounds. It is widely employed in computational chemistry, cheminformatics, and molecular modeling software for representing and exchanging molecular data.
- Genetic algorithm: Genetic algorithm applied in virtual screening is a computational method that employs a population-based search and optimization strategy to explore and evaluate a diverse set of chemical compounds in order to identify promising drug candidates or ligands. Below are the elements of genetic algorithm which are implemented in different components:
  - Population: a population consists of a set of chemical compounds (molecules) represented by their respective molecular structures.
  - Fitness Function: A fitness function is defined to evaluate and score the binding affinity or other relevant properties of each compound with the target protein or biomolecule.
  - Selection: Compounds are selected for reproduction (crossover and mutation) based on their fitness scores.
  - Crossover: Crossover operations combine the genetic information of two substructures of the compound to create another aspect of compound.
  - Mutation: Mutation introduces random changes to the genetic information of compounds, mimicking genetic diversity.
  - Replacement Strategy: A replacement strategy determines how the new generation of compounds replaces the old generation in the population.
  - Termination Criteria: The algorithm iterates through multiple generations until a termination criterion is met, such as a fixed number of generations or a specific fitness threshold.
- Cavity selection: Cavity selection refers to defining a specific region or pocket within a protein that potentially exists strong interactions between compounds and the targeted protein.

3.2 Cavity selection component

iGEMDOCK allows users to pass parameters to customize the specific region based on their domain knowledge and regenerate a PDB file for virtual screening. If the user doesn't customize the cavity, iGEMDOCK will use the default method to define the cavity according to the information in the given PDB file.

3.3 Post-screening analysis

Post-screening analysis is to infer pharmacological interactions and cluster screening compounds based on protein-ligand complexes and compound structures. Once the docking process is finished, iGEMDOCK will generates interaction profiles and calculates the pharmacological preference of each interacting group for deriving the pharmacological interactions. iGEMDOCK supports KMeans and the hierarchical clustering method to analyze screening compounds according to interaction profiles and their atomic composition.

3.4 GUI

GUI is responsible for reading and parsing the parameter for customization, file locations as well as the visualization of the result generated by cavity selection and post-screening analysis.

3.5 Docking and Virtual Screening

- Compound data processing: A molecular file has multiple standard formats, so iGEMDOCK defines different methods to process the given file.
- Protein complex data processing: A PDB file consists of all related information about the experiment and the detail.
- Genetic algorithm: Genetic algorithm allows users to pass hyperparameters to make the search converges faster. In this component, there are multiple functions to implement the elements of genetic algorithm including initializing population, selection, crossover, mutation and replacement.
- Scoring: The component is to implement self-defined fitness functions in genetic algorithm.
- GEMDOCK: The main component in docking and virtual screening, which allows users to do virtual docking or single docking for one compound.

3.6 Interactions between components

- GUI $\rightarrow$ Cavity selection:
  Pass the parameters to generate customized cavity.
- Cavity selection $\rightarrow$ GUI:
  Visualize the result of cavity selection.
- Cavity selection $\rightarrow$ Docking and Virtual Screening:
  Pass the customized PDB file to do virtual screening
- GUI $\rightarrow$ Docking and Virtual Screening:
  Pass the parameters of genetic algorithm and file locations to do virtual screening.
- Compound Data Processing $\rightarrow$ GEMDOCK:
  Pass the encoded compound data to do virtual screening.
- Protein Complex Data Processing $\rightarrow$ GEMDOCK:
  Pass the encoded protein data to do virtual screening.

- Genetic Algorithm → GEMDOCK:
  GEMDOCK call the functions defined in Genetic Algorithm to do virtual screening.
- Scoring → GEMDOCK:
  GEMDOCK call the functions defined in Scoring to do virtual screening

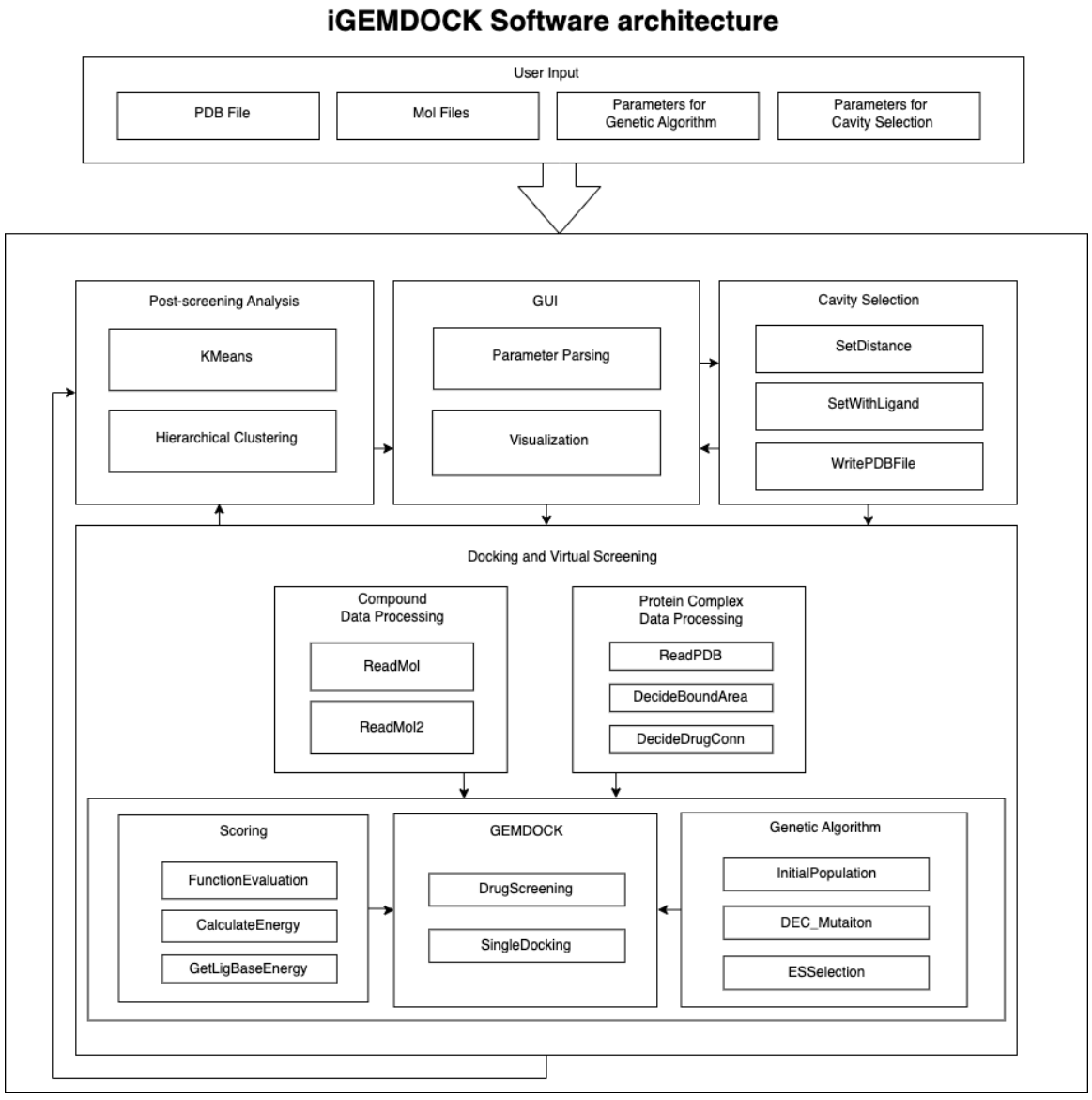

Figure 1: iGEMDOCK Software architecture

4 Language selection

We propose to use OpenCL as the parallel language for parallelizing iGEMDOCK. OpenCL is a versatile and widely supported framework for parallel computing that can be used to accelerate computations on various hardware platforms, including CPUs, GPUs, and FPGAs. Below are the reasons why we have chosen OpenCL.

4.1 Portability

OpenCL is platform-agnostic and can be deployed on a wide range of hardware, making it suitable for a diverse user base. This ensures that iGEMDOCK can benefit from parallelization on different computing devices, from high-end GPUs to multi-core CPUs.

4.2 Heterogeneous Computing

OpenCL enables heterogeneous computing, allowing us to harness the power of different types of devices simultaneously. This flexibility is especially valuable in virtual screening, where various aspects of the computation can benefit from parallelization.

4.3 Existing Support

OpenCL has a well-established user base and is supported by major GPU manufacturers, including AMD and NVIDIA. This ensures that the parallelized iGEMDOCK can be run efficiently on a variety of hardware configurations.

4.4 Performance Optimization

OpenCL provides low-level control over memory management and thread scheduling, allowing us to fine-tune performance and optimize resource usage for the specific requirements of iGEMDOCK.

5 Related works

The development of virtual screening (VS) has seen significant advancements over the years, with a focus on parallelization for improved performance and efficiency. Originally, many VS tools were sequential, running on single-core CPUs. However, as computational resources have advanced, so has the parallelization of VS. Here, we discuss the evolution of VS tools and their parallel implementations, showcasing key developments in this field[1].

Early implementations of VS relied on sequential processing, with software like Dock5[2, 3] and Dock6[4] being widely used. They primarily running on single-core CPUs and use MPI for acceleration. These tools utilized physics-based scoring functions (geometric shape-matching approach) but were limited by their inability to leverage multiple CPU cores effectively.

As computational power increased, the need for parallelization became apparent. Tools like Autodock Vina[5] emerged, offering multi-threading options to run on multi-CPU by using OpenMP. It retained the physics-based scoring functions but introduced the ability to use multiple CPU cores simultaneously for molecular docking calculations. Autodock Vina allowed for both single-threaded and multi-threaded options, providing researchers with the flexibility to choose their preferred mode of operation.

By 2011 to 2013, the introduction of MPAD4[6], VinaLC[7], and VinaMPI[8] represented a hybrid scheme for parallelization across nodes. They used MPI and OpenMP to allow efficient allocation of computing resources, resulting in these scalable and high-throughput VS tools.

The introduction of GPU acceleration marked a significant milestone in VS. Autodock-GPU[9], for example, was developed to run on multiple-node parallel computers with GPU accelerators by using OpenCL. The use of GPUs greatly improved the performance of VS tools, as they could handle massive parallelism efficiently.

Over time and technology development, machine learning-based scoring functions were introduced, offering better accuracy in predicting binding affinities and improving the efficiency of lead compound identification. These scoring functions were integrated into tools like GNINA[10], enhancing their predictive capabilities.

Parallelization wasn't limited to CPUs and GPUs. Hybrid systems that utilized both CPUs and GPUs, such as LiGen Docker-HT[11] and GeauxDock[12, 13, 14] emerged as an efficient way by using MPI or OpenMP and CUDA to perform virtual screening on high-performance computing systems. These tools harnessed the power of both hardware types to optimize the screening process.

6 Statement of expected results

- Significant Speedup:
  We anticipate a substantial reduction in the time required for virtual screening, enabling researchers to screen a larger number of drug candidates within a reasonable timeframe.
- Improved User Experience:
  Parallelization will lead to faster and more efficient virtual screening, making the software more user-friendly and accessible to a wider audience.
- Enhanced Scalability:
  iGEMDOCK's parallelization will allow it to efficiently utilize the computational resources available on a user's system, enabling scalability across different hardware configurations.
- Validation and Benchmarking:
  We will conduct extensive validation and benchmarking to assess the performance improvements achieved through parallelization and ensure the reliability and accuracy of the results.

7 Timetable

The timetable includes the following key stages:

- Project initiation and requirements analysis
- Parallelized algorithm design and development
- Performance optimization and testing
- Results evaluation and comparison
- Documentation and project completion

References

[1] Natarajan Arul Murugan, Artur Podobas, Davide Gadioli, Emanuele Vitali, Gianluca Palermo, and Stefano Markidis. A review on parallel virtual screening softwares for high-performance computers. *Pharmaceuticals*, 15(1):63, 2022.

[2] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. A high-performance, portable implementation of the mpi message passing interface standard. *Parallel computing*, 22(6):789–828, 1996.

[3] Demetri T Moustakas, P Therese Lang, Scott Pegg, Eric Pettersen, Irwin D Kuntz, Natasja Brooijmans, and Robert C Rizzo. Development and validation of a modular, extensible docking program: Dock 5. *Journal of computer-aided molecular design*, 20:601–619, 2006.

[4] William J Allen, Trent E Balius, Sudipto Mukherjee, Scott R Brozell, Demetri T Moustakas, P Therese Lang, David A Case, Irwin D Kuntz, and Robert C Rizzo. Dock 6: Impact of new features and current docking performance. *Journal of computational chemistry*, 36(15):1132–1156, 2015.

[5] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[6] Andrew P Norgan, Paul K Coffman, Jean-Pierre A Kocher, David J Katzmann, and Carlos P Sosa. Multilevel parallelization of autodock 4.2. *Journal of cheminformatics*, 3:1–9, 2011.

[7] Xiaohua Zhang, Sergio E Wong, and Felice C Lightstone. Message passing interface and multithreading hybrid for parallel molecular docking of large databases on petascale high performance computing machines. *Journal of computational chemistry*, 34(11):915–927, 2013.

[8] Sally R Ellingson, Jeremy C Smith, and Jerome Baudry. Vinampi: Facilitating multiple receptor high-throughput virtual docking on high-performance computers, 2013.

[9] Diogo Santos-Martins, Leonardo Solis-Vasquez, Andreas F Tillack, Michel F Sanner, Andreas Koch, and Stefano Forli. Accelerating autodock4 with gpus and gradient-based local search. *Journal of chemical theory and computation*, 17(2):1060–1073, 2021.

[10] Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. Gnina 1.0: molecular docking with deep learning. *Journal of cheminformatics*, 13(1):1–20, 2021.

[11] Claudia Beato, Andrea R Beccari, Carlo Cavazzoni, Simone Lorenzi, and Gabriele Costantino. Use of experimental design to optimize docking performance: The case of ligendock, the docking module of ligen, a new de novo design program, 2013.

[12] Ye Fang, Yun Ding, Wei P Feinstein, David M Koppelman, Juana Moreno, Mark Jarrell, J Ramanujam, and Michal Brylinski. Geauxdock: accelerating structure-based virtual screening with heterogeneous computing. *PloS one*, 11(7):e0158898, 2016.

[13] Jianbin Fang, Ana Lucia Varbanescu, Baldomero Imbernón, José M Cecilia, and Horacio Emilio Pérez Sánchez. Parallel computation of non-bonded interactions in drug discovery: Nvidia gpus vs. intel xeon phi. In *IWBBIO*, pages 579–588, 2014.

[14] John E Stone, David Gohara, and Guochun Shi. Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(3):66, 2010.