



Redesigning Vina@QNLN for Ultra-Large-Scale Molecular Docking and Screening on a Sunway Supercomputer

Hao Lu^{1,2†}, Zhiqiang Wei^{1,2†}, Cunji Wang¹, Jingjing Guo¹, Yuandong Zhou¹, Zhuoya Wang² and Hao Liu^{1,2*}

¹College of Computer Science and Technology, Ocean University of China, Qingdao, China, ²Pilot National Laboratory for Marine Science and Technology, Qingdao, China

OPEN ACCESS

Edited by:

Antonio Monari,
UMR7019 Laboratoire de Physique et
Chimie Théoriques, France

Reviewed by:

Isabel Iriepa,
University of Alcalá, Spain
Rodrigo Quiroga,
National University of Cordoba,
Argentina

*Correspondence:

Hao Liu
liu.hao@ouc.edu.cn

[†]These authors have contributed
equally to this work and share first
authorship

Specialty section:

This article was submitted to
Theoretical and Computational
Chemistry,
a section of the journal
Frontiers in Chemistry

Received: 30 July 2021

Accepted: 14 September 2021

Published: 28 October 2021

Citation:

Lu H, Wei Z, Wang C, Guo J, Zhou Y,
Wang Z and Liu H (2021) Redesigning
Vina@QNLN for Ultra-Large-Scale
Molecular Docking and Screening on a
Sunway Supercomputer.
Front. Chem. 9:750325.
doi: 10.3389/fchem.2021.750325

Ultra-large-scale molecular docking can improve the accuracy of lead compounds in drug discovery. In this study, we developed a molecular docking piece of software, Vina@QNLN, which can use more than 4,80,000 parallel processes to search for potential lead compounds from hundreds of millions of compounds. We proposed a task scheduling mechanism for large-scale parallelism based on Vinardo and Sunway supercomputer architecture. Then, we readopted the core docking algorithm to incorporate the full advantage of the heterogeneous multicore processor architecture in intensive computing. We successfully expanded it to 10, 465, 065 cores (1,61,001 management process elements and 0, 465, 065 computing process elements), with a strong scalability of 55.92%. To the best of our knowledge, this is the first time that 10 million cores are used for molecular docking on Sunway. The introduction of the heterogeneous multicore processor architecture achieved the best speedup, which is 11x more than that of the management process element of Sunway. The performance of Vina@QNLN was comprehensively evaluated using the CASF-2013 and CASF-2016 protein–ligand benchmarks, and the screening power was the highest out of the 27 pieces of software tested in the CASF-2013 benchmark. In some existing applications, we used Vina@QNLN to dock more than 10 million molecules to nine rigid proteins related to SARS-CoV-2 within 8.5 h on 10 million cores. We also developed a platform for the general public to use the software.

Keywords: molecular docking, virtual screening, parallel computing, Sunway supercomputer, drug discovery

INTRODUCTION

The virtual screening technology can be employed to obtain lead compounds from a large number of candidate compounds, thus significantly reducing the cost of drug discovery (Pagadala et al., 2017). Currently, the molecular docking technology is one of the most commonly used virtual screening technology. The accuracy of virtual screening depends on the scale of molecular docking. The mainstream ZINC15 database in the field of drug screening currently contains 750 million compounds that can be purchased (Irwin et al., 2012). To determine the ideal lead compounds in molecular docking, each target needs to be docked with each compound. In addition, more than 20,000 drug targets have been identified. In such a massive computing scenario, if only a single

computer or small cluster is used for docking, then the running computing time is unacceptable. Therefore, large-scale parallel computing is required to solve this problem.

Super-large-scale molecular docking parallel computing has large bottlenecks in multiple links, such as input/output (I/O), task allocation, and communication. To realize large-scale parallel computing, we need to encapsulate a series of interfaces at the job scheduling level. Meanwhile, the massive read and write processes overwhelm the file system, and the necessary integration and optimization of I/O operations must be conducted. The thread and cache utilization of the docking algorithm should be optimized to totally improve parallelism while reducing inefficient operations, such as access and storage (Dong et al., 2018). The accuracy of virtual screening, such as molecular docking, is highly dependent on the size of the candidate molecular library and the performance of the screening algorithm. The misprediction of early lead compounds can be costly and time-consuming in the subsequent stages of drug development. The molecular docking algorithm estimates the most likely binding structure and energy between a pair of ligands and receptors. Currently, we aim to obtain the ideal conformation by considering the atomic space interaction, hydrophobic interaction, hydrogen bonding, and other factors. However, excessive physical descriptors limit the running speed, thus affecting the screening scale and reducing the screening accuracy. Therefore, it is necessary to reasonably design a docking algorithm and improve the speed of molecular docking and the screening scale (Zhang et al., 2013; Li et al., 2019).

The first molecular docking software DOCK (Allen et al., 2015) was developed in 1980. However, because of the limitations of the algorithm in this software, its calculation accuracy is poor. In addition, it is a serial program with limited computing power. Accordingly, a parallel strategy was introduced to solve the computing efficiency problem. The message passing interface (MPI) was introduced to achieve process-level parallelism using a large number of cores to reduce the docking completion time (e.g., VinaMPI) (Nocedal and Wright, 1999; Ellingson et al., 2013; Liu et al., 2020; Zhang et al., 2013). At the same time, the docking efficiency is improved through thread-level parallelism. Multiple CPUs of a single processor—each of which simultaneously performs optimized search algorithms—implement thread-level parallelism [e.g., AutoDock Vina (herein referred to as Vina)] (Trott and Olson, 2009).

In the past decade, graphical processing units (GPUs) have shown superior performance in the field of image processing. In the field of molecular docking, a GPU is gradually introduced to facilitate the computation. For example, MolDock (Thomsen and Christensen, 2006) was used in the multithreading technology to accelerate the calculation process. Heterogeneous multicore systems adopt the main processor + coprocessor architecture. In this architecture, the main processor handles complex logic control tasks, such as the distribution of molecular docking tasks and reading of files. The coprocessor processes the computational part of the molecular docking and docking algorithm that performs multiple iterations. This type of computation has a high density and a simple logical branch of large-scale data. The two works together provide an efficient computing platform for specific applications, and different parallelism strategies can be

used to solve different problems in the field of molecular docking (Altuntaş et al., 2016). Heterogeneous multicore architectures have advantages in the field of molecular docking owing to their unique architecture (Fu et al., 2016; Chen et al., 2017; Xiao et al., 2021).

In this study, we successfully extended the molecular docking system to 10 million cores in Sunway supercomputer by introducing new strategies and algorithms and demonstrate parallelism in the molecular docking. We evaluated our algorithms and strategies on Sunway supercomputer, a heterogeneous multicore processor system. Experimental results show that the docking accuracy is better than that of traditional molecular docking methods and has a good parallelism efficiency and computation speed. Our main contributions are as follows:

- We added an ultra-large-scale molecular docking scheduling parallelism for large-scale parallelism based on Vinardo's and Sunway supercomputer architecture. We managed to scale to 1,61,001 management processing elements (MPEs) and 10,304,064 computing processing elements (CPEs) with a strong scalability of 55.92%.
- We readopted the core docking algorithm to take full advantage of the heterogeneous multicore processor architecture in intensive computing. The introduction of the heterogeneous multicore processor architecture achieved the best speedup, which is 11x more than that of the management process element of Sunway. The accuracy of the software was evaluated using the CASF-2013 and CASF-2016 benchmarks.
- We developed a docking platform, including core docking functions, to meet the docking needs of general users. Existing docking practices include docking more than 10 million molecules to nine rigid proteins about SARS-CoV-2 within 8.5 h.

BACKGROUND

Sunway's Processor

Sunway's processor was developed based on the previous-generation SW26010 processor. Sunway's multicore processors integrate several core groups (CGs), each of which contains an MPE and an array of 8×8 CPEs, which are connected through a ring network. MPEs mainly include computation, control, communication, and I/O functions. CPEs are mainly used for computation. A CPE and MPE are called a master core and slave core, respectively. The CPEs in the CGs are interconnected with one another and with external interactions through an intra-array network. Data communication (peer-to-peer and aggregate communication) between any two CPEs in the array can be performed via a register-level data communication method (see **Supplementary Figure S1B**).

The MPE uses an autonomous SW64 instruction set with a data cache. The computing processing element supports double-precision, single-precision, and half-precision floating-point computation and integer operations. Each CPE has its own independent instruction cache and data storage space. The

data storage space can be configured as a fully user-controlled local data memory (LDM), or part of the data storage space can be configured as a hardware piece's automatically managed local data cache (LDcache). The data transfer between the LDM and main memory can be achieved using the direct memory access (DMA) or conventional load/store instructions between the LDcache and main memory.

Sunway's Supercomputer

Sunway supercomputer system inherits and develops the architecture of "Sunway TaihuLight" based on high-performance heterogeneous multicore processors and interconnection network chips. The system consists of a computing system, interconnection network system, software system, peripheral service system, maintenance and diagnosis system, power supply system, and cooling system (see **Supplementary Figure S1B**).

The software system supports the management and scheduling of super-large-scale programming resources with more than 10 million cores; it provides system-wide monitoring management, fault-tolerant mechanisms, multilevel debugging, and performance tuning of application software. In addition, it supports parallel programming environments, such as MPI, OpenMP, and OpenAcc. Thus, it can boost the implementation of the molecular docking parallel strategy.

Vina@QNLN

Vina is a popular molecular docking program in drug discovery at early stages and opts for the iterated local search global optimizer to search for the minimum affinity for molecular docking. Specifically, the software uses the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (Nocedal and Wright, 1999) to find the local optimum.

The value of the scoring function and its gradient, such as the derivative of the scoring function with respect to its parameters, are used in the BFGS algorithm. The parameters of the scoring function include the orientation and position of the small molecule and the values of the torsions for the active rotatable bonds. As for the binding energy, the sum of distance-dependent atom pair interactions is considered in **Eq. 1**:

$$E = \sum e_{pair}(sd). \quad (1)$$

Here, *sd* is the surface distance, which is equal to the difference between the interatomic distance and the radii of the atoms in the pair. There are five terms in total.

As can be seen in **Eq. 2**, the final predicted binding affinity is that each term is multiplied by a constant and then added to the sum. For more details, please refer to the original article by Trott and Olson (2009).

$$\begin{aligned} e_{pair}(sd) = & w_1 * \text{Gauss}_1(sd) + w_2 * \text{Gauss}_2(sd) \\ & + w_3 * \text{Repulsion}(sd) + w_4 * \text{Hydrophobic} \\ & + w_5 * \text{HBond}(sd). \end{aligned} \quad (2)$$

Vinardo (Quiroga and Villarreal, 2016) is based on AutoDock Vina and uses a simpler scoring function, while improving the

docking, scoring, ranking, and virtual screening results of the scoring function. First, the main difference of Vinardo is that the second minimum, which is physically unreasonable in the Vina space interaction, is eliminated, and then, Vinardo adjusts atomic radii, such as nitrogen and oxygen aliphatic carbons, to further improve the performance of the scoring function. Vinardo uses a combination of Gaussian gravity and quadratic repulsion to manage steric interactions. Finally, a hydrophobic term in Vinardo, which is approximately the same as the diameter of a water molecule, can be obtained. The scoring function of Vinardo shows good performance on the CASF-2013 benchmark (Li et al., 2018).

In our implementation, we present a molecular docking piece of software called Vina@QNLN, which adopts the scoring function of Vinardo and is reconstructed based on the architectures of Vina and Sunway supercomputer. We added an ultra-large-scale molecular docking scheduling parallelism.

DESIGN AND OPTIMIZATION

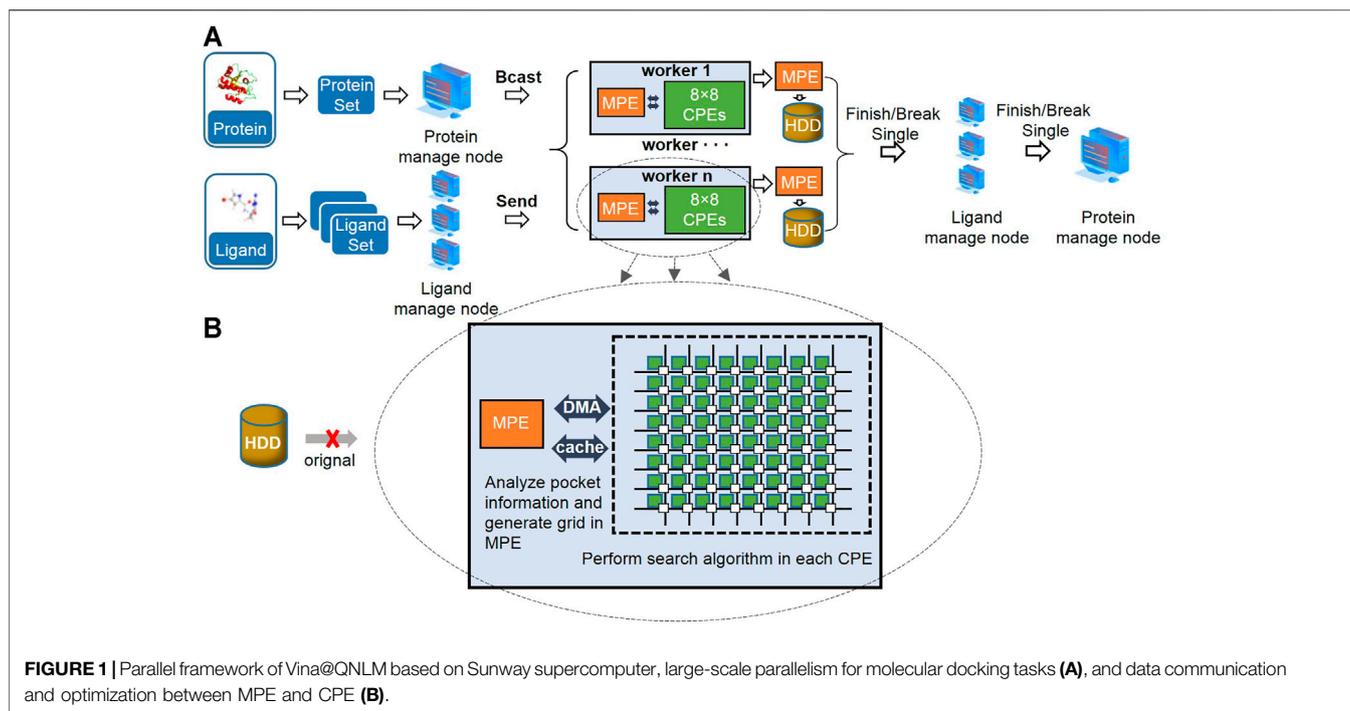
In this section, we introduce the parallel implementation of Vina@QNLN on Sunway supercomputer in detail. A two-level parallelization strategy, including the combination of the master–master core and multi-thread heterogeneous parallelism of a new processor on Sunway supercomputer, is introduced to achieve the scalability and speedup of Vina@QNLN.

Large-Scale Parallelism for Molecular Docking Tasks

In the process of ultra-large-scale docking, scheduling and I/O operations have been a problem. First, this is a typical many-to-many problem. Multiple targets are required, and there are multiple pocket files. Each molecule must be docked with the target. Second, the program needs to be carefully designed so that pocket information and grid data can be reused. To reduce the number of I/O operations, the data must be packaged and stored.

There are a lot of previous successful experiences in parallel computing on Sunway [e.g., Parallel CFD Simulation Software, reactive force field simulation, and parallel implementation of the discontinuous Galerkin density functional theory (DGDFT) on Sunway's supercomputer] (Tsutsui and Collet, 2013; Gao et al., 2020; Hu et al., 2021). According to the current software foundation, we employed MPI for data transmission, which supports point-to-point communication and is mature on Sunway supercomputer.

As the first level of parallelism, we adopted MPI to realize the docking of proteins and ligands between different MPEs, addressing the task scheduling and allocation problems (see **Figure 1A**). First, a process called the "protein manage node" was allocated to read all the target and pocket information in batches, and subsequently, the information was broadcasted to each work node. Thereafter, a certain amount of "ligand manage node" was allocated to read small-molecule information, thus realizing the reuse of pocket and grid information. The number of



ligand manage nodes must be carefully selected. If it is considerably large, then it will cause a decrease in computing efficiency; otherwise, it will cause a performance bottleneck. The size of the PDBQT format data of small molecules is approximately 1,500 bytes, and the number of ligand manage nodes is positively correlated with the total number of processes. Thus, we selected the total number of processes/ligand manage nodes as ~300. Subsequently, a certain number of small molecules were evenly distributed to each work node to form a small-molecule library. The number of molecules that each ligand manages to send to the worker was calculated by Eq. 3, where n_l , n_{tp} , and n_{lm} represent the number of molecules in each ligand set, the number of all processes, and the number of ligand manage nodes, respectively. By docking each target with the molecules in the small-molecule library, docking simulation data were generated. Thereafter, the results were returned to the MPE, and the docking result data were saved. The MPE in the worker sends a finish or break signal to the ligand manage node, and the ligand manage node sends the signal to the protein manage node. Finally, the program ends.

$$f(x) = \frac{n_l}{(d_{lp} - 1 - n_{lm}) * n_{lm}}. \quad (3)$$

To ensure that each process remains busy and reduces the idle time based on the characteristics of multicores and heterogeneity of the processor and to increase the I/O efficiency, a multilevel processing scheme was adopted to separately open up multiple processes to read a large number of small-molecule input data files, target data files, and pocket files. This process plays the role of a scheduler. Because the target data are cumbersome and have many coordinates, when a large number of targets need to be docked, it may result in an insufficient stack space. Therefore, we gradually eliminated the target data that had completed docking,

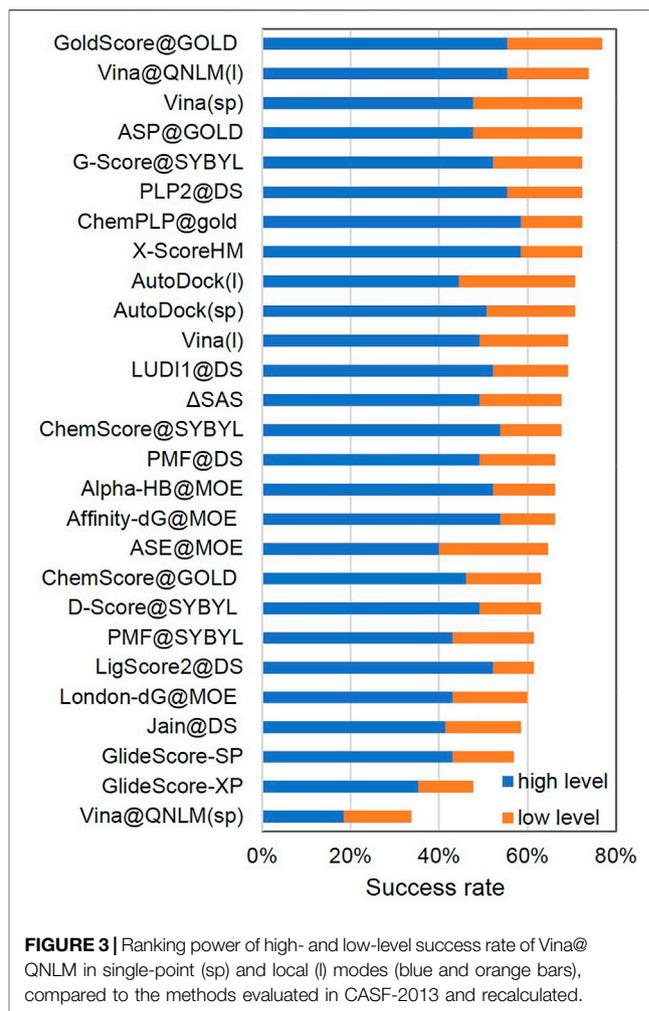
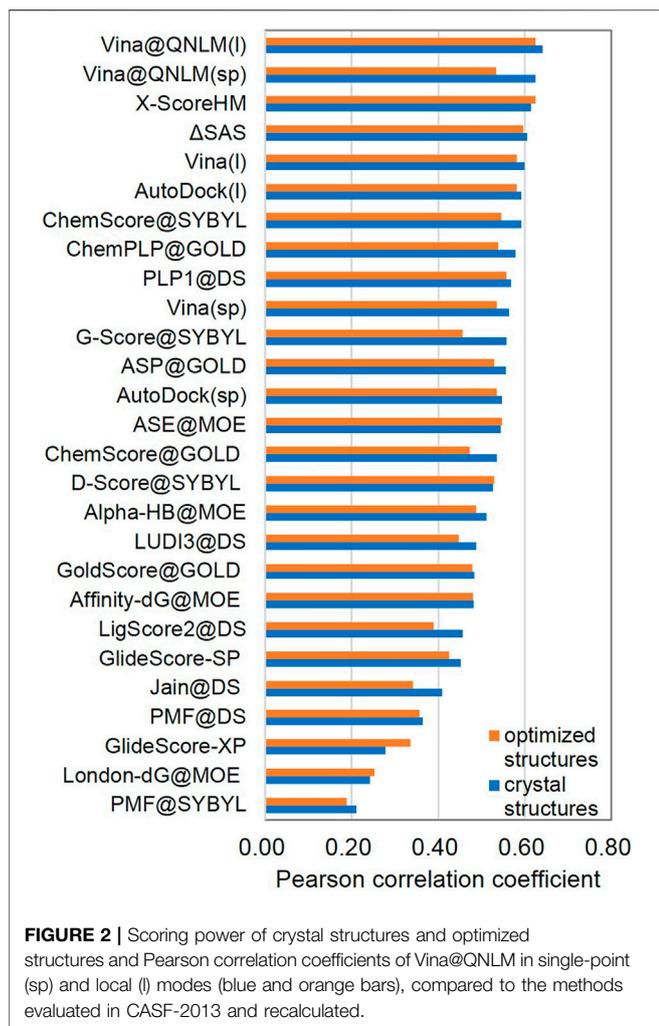
thereby reducing the number of times the file was read, while increasing the I/O efficiency.

Adaptation of the Molecular Docking Algorithm

The original strategy is that each MPE uses a hard disk to read the data. However, this approach causes performance bottlenecks. Through the scheduling strategy mentioned in Section 3.1, the target data can be reused considerably. Here, the MPE in a CG reads pocket information in the memory and generates grid data. This information can be reused for the same protein. Considering the docking time and efficiency, we transplanted the most time-consuming search algorithms as a second level of parallel and adopted the Monte Carlo simulated annealing algorithm using time as the random number seed to search for the conformation of the molecule. The algorithm also maintains a high computational efficiency, accuracy of the docking results, and speed of large-scale docking. Inspired by Vina, we executed algorithms on each CME (Figure 1B). We employed the BFGS algorithm to determine the minimum local energy for the optimal local spatial orientation and location information. This process is accompanied by data interactions by the DMA and cache method. It reserves the best 20 poses in each CME and then transfers them to the MPE. After comparison, the MPE retained the specified poses.

Acceleration of the Docking Algorithm Based on Sunway

The CPE contains local data management (LDM). To achieve a high memory bandwidth, the CPE core accesses the data by



directly accessing the main memory or transferring the data to the LDM through a DMA operation and then loads it from the LDM. Three types of data should be loaded into the LDM: configuration information, parameters, and molecular coordinates. These configurations were loaded into the LDM prior to the task cycle. Therefore, during the actual conformation search and position-matching process, the data could be accessed quickly.

The values of the four parameters, including the van der Waals force and Gauss term between the receptor and ligand atoms, are not related to the small ligand molecule. The software distributes the four parameters in a virtual three-dimensional grid containing a receptor small molecule. The data volume of the energy grid is too large to be completely incorporated into the local memory of the CPE. For the conformation search and energy calculation process, we used the cache method to access the main memory data to facilitate the docking calculation. The force, which is a combination of quadratic repulsion and Gaussian gravity to handle steric interactions, adopted the same method.

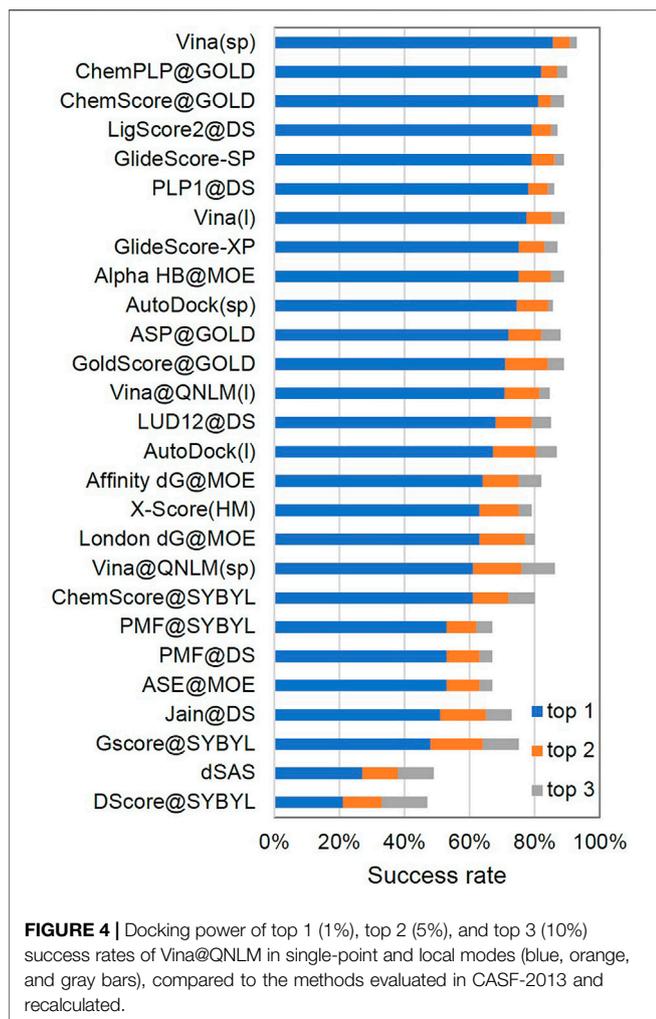
RESULTS AND DISCUSSION

Scoring Function and Sample Power

To evaluate the accuracy of the scoring function, we conducted evaluations on the CASF-2013 (Li et al., 2018) and CASF-2016 benchmarks (Su et al., 2019). The two benchmarks evaluate scoring functions using four metrics: ranking power, scoring power, docking power, and screening power. We also added another metric: sample power for evaluating Vina@QNLN predicting the ligand pose. In this article, we only briefly review the definitions; a more precise and complete definition can be found in the CASF-2013 article (Li et al., 2014) and CASF-2016 article (Su et al., 2019).

Evaluation on the CASF-2013 Benchmark

The CASF-2013 benchmark consists of 195 ligand–receptor complex structures with associated affinities. In addition, 16,123 computer-generated ligand poses were provided for docking power evaluation and 6,36,010 for screening power evaluation. The correctness of Vina@QNLN after transplantation was verified using the entire CASF-2013



benchmark, and it was added to 27 docking software models with the results of the CASF-2013 test for a comparative analysis.

Scoring Power and Ranking Power

The scoring power (see **Figure 2; Supplementary Table S1**) determines whether a score generated by the scoring function in Vina@QNLM could be linearly correlated with the experimental binding constants. Pearson's correlation coefficient was used to measure this metric. The ranking power (see **Figure 3; Supplementary Table S2**) evaluates the ability of a scoring function to correctly rank the ligands of a target protein based on their experimental affinities. The ranking power is measured by the "high-level" success rate, and the number of targets where the three complexes are correctly ranked is counted (i.e., best > median > poorest). A "low-level" success rate is counted with a relaxed criterion (best > median and best > poorest). For the scoring power in single-point and local modes, Vina@QNLM achieved correlation coefficients of 0.625 and 0.641 for crystal structures and 0.534 and 0.624 for optimized structures. Compared to the 27 methods, Vina@QNLM is the best in relation to experimental data. In addition, the two modes of Vina@QNLM showed two extreme

results in terms of the ranking power. The local mode showed the best success rate in the high and low levels, whereas the single-point mode showed the worst success rate.

Docking Power and Screening Power

The docking power is used to evaluate the ability of Vina@QNLM to distinguish between the native pose and decoy ligand binding poses for a given receptor. If a ligand binding pose is less than or equal to 2 Å as compared to the experimental ligand pose, then we consider the pose to be native. Typically, a ligand pose is evaluated using the root-mean-square deviation (RMSD) and does not take into account the hydrogen atoms of the ligand. For each receptor, the poses considered include the computer-generated poses and ligand crystallographic structure. The docking power was measured by the success rate: the percentage of proteins with a native pose as the best (top 1), second (top 2), or third (top 3) scored pose. For the docking power, Vina@QNLM nearly ranked in the middle of the 27 methods (see **Figure 4; Supplementary Table S3**).

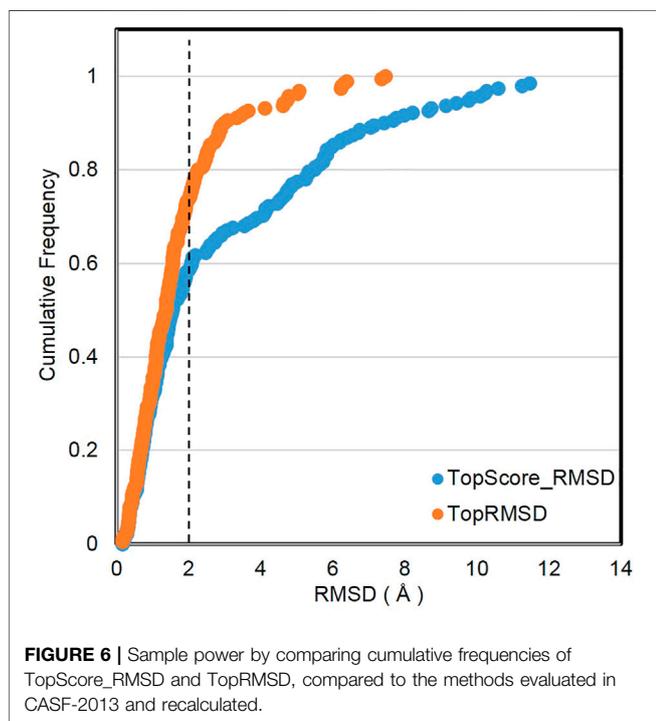
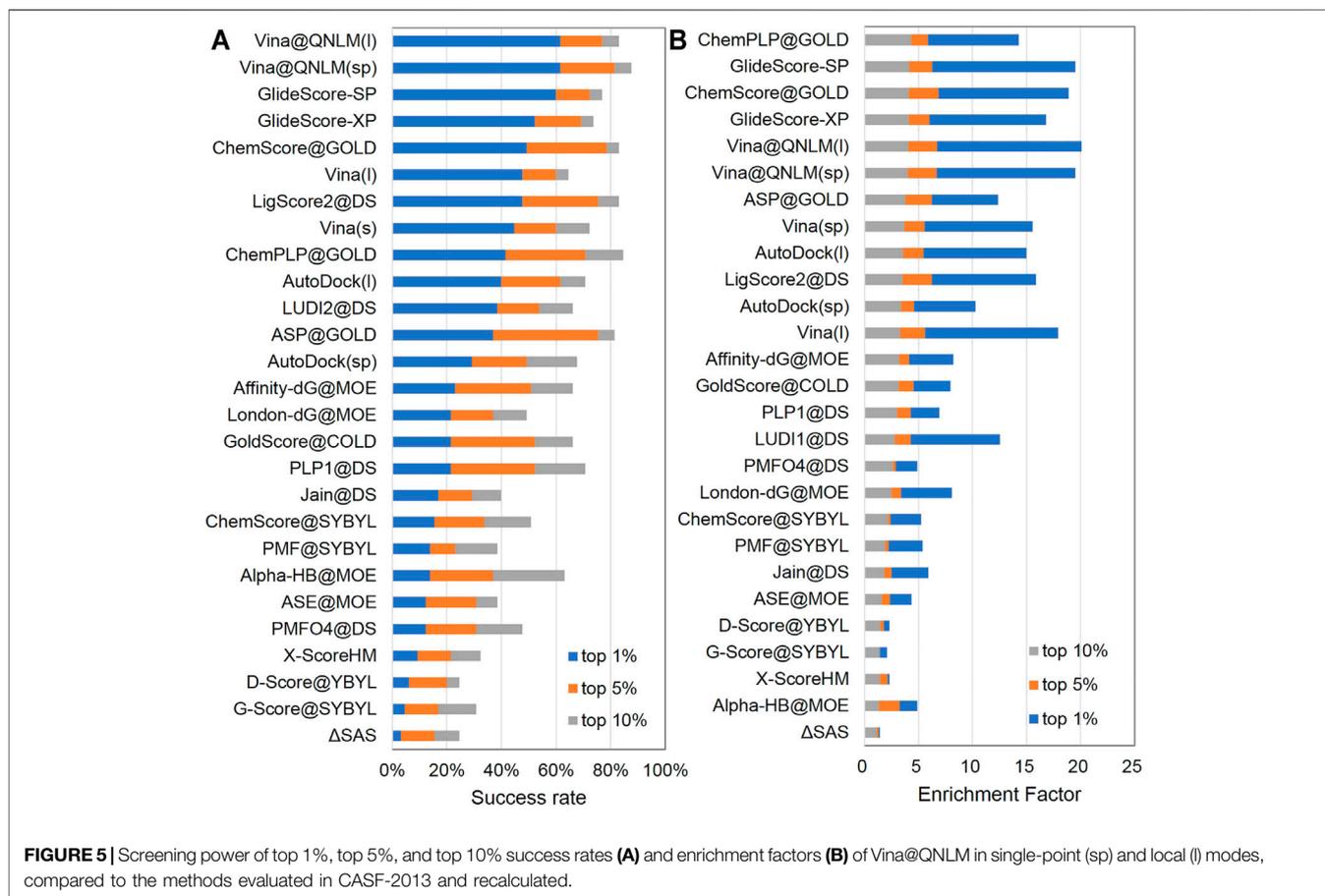
The purpose of the screening power is to evaluate the ability of the molecular docking software to separate the two ligands. We adopt the enrichment factor and success rate to assess the screening power of our molecular docking software. The enrichment factor is defined as the number of binder ligands among the best 1% (top 1%), 5% (top 5%), or 10% (top 10%) scored poses, divided by the total number of binder ligands times 1, 5, or 10%. Among the 27 methods, Vina@QNLM was almost the best (see **Figure 5A; Supplementary Table S4**). The success rate is the percentage of targets for which the ligand with the best experimental affinity is among the best 1% (top 1%), 5% (top 5%), or 10% (top 10%) scored poses. Vina@QNLM is at the top among the 27 methods (see **Figure 5B; Supplementary Table S5**).

Sample Power

The sample power refers to the ability of the molecular docking software to predict the binding poses of a ligand. We used RMSD to evaluate the sampling power. The prediction of Vina@QNLM is regarded to be successful if the RMSD between the native binding pose and docked binding pose is below 2.0 Å. Two methods were used to evaluate the sample capability in our work, including the highest conformation generated by the molecular docking software (referred to as the top-score RMSD) and the closest conformation to the native binding pose (referred to as the top RMSD). The figure indicates that the success rate of Vina@QNLM for the top-scored pose is 59%, but that for the top RMSD pose is significantly higher (73%) (see **Figure 6**). With reference to other molecular docking software (not the same benchmark) (see Wang et al., 2016), Vina@QNLM is in between the middle to upper levels.

Evaluation on the CASF-2016 Benchmark

We also use the updated benchmark to evaluate the scoring function of Vina@QNLM. CASF-2016 is basically the same as the CASF-2013 benchmark, only slightly different in some indicators. The same data were prepared with some definitions of terms that we do not describe here but can be referred to Wang et al. (2016).



Scoring Power and Ranking Power

The scoring power of the scoring function was measured by the correlation between the calculated binding score and the experimental binding constants of a given protein ligand. **Supplementary Table S6** shows the Pearson correlation coefficient and standard deviation (SD) generated by 35 scoring functions. **Figure 7** shows the evaluation results of all scoring functions on the structure of the original crystal complex. The highest Pearson linear correlation coefficient is that of Δ VinaRF₂₀, that is, 0.816, significantly higher than those of other scoring functions. The second highest is Vina@QNLN, whose Pearson linear correlation coefficients are 0.641 and 0.649 in the single-point and local modes, respectively.

In addition to using the original crystal complex structure, the scoring ability of the locally optimized complex structure was also tested. The Pearson correlation coefficients and SDs are presented in **Supplementary Tables S6, S7**; **Figure 7**. The Pearson correlation coefficient of Δ VinaRF₂₀ is the highest among the 35 scoring functions, reaching 0.774, followed by Vina@QNLN, that is, 0.633 and 0.642 in the single-point and local modes, respectively.

In the CASF-2016 benchmark, the average Spearman rank correlation coefficient (ρ), Kendall correlation coefficient (τ), and prediction index (PI) are equivalent for the score function.

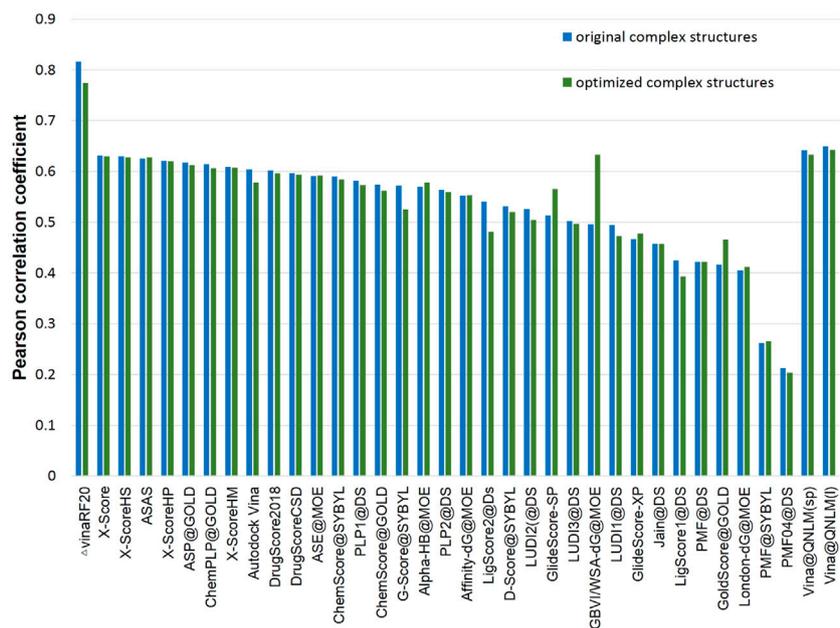


FIGURE 7 | Scoring power of original complex structures and optimized complex structures and Pearson correlation coefficients of Vina@QNLM in single-point (sp) and local (l) modes (blue and green bars), compared to the methods evaluated in CASF-2016 and recalculated.

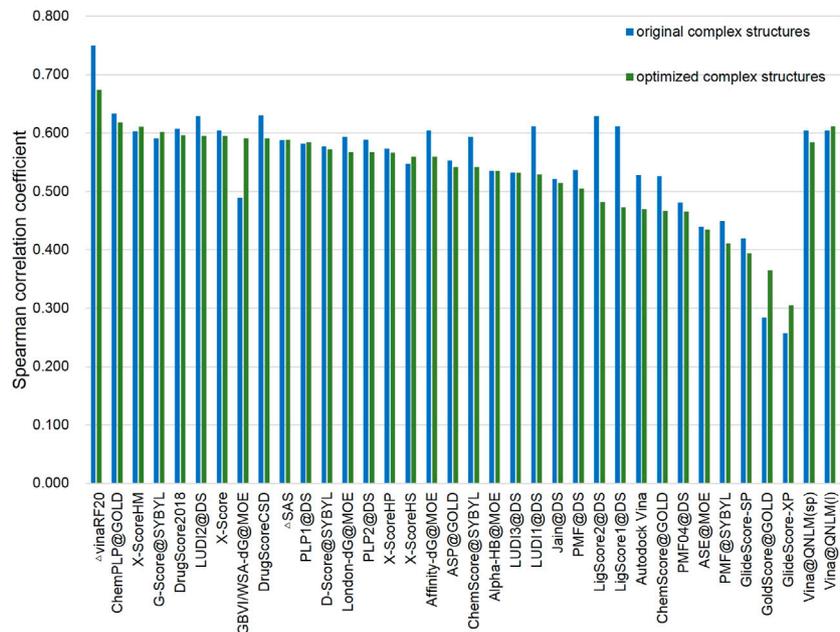
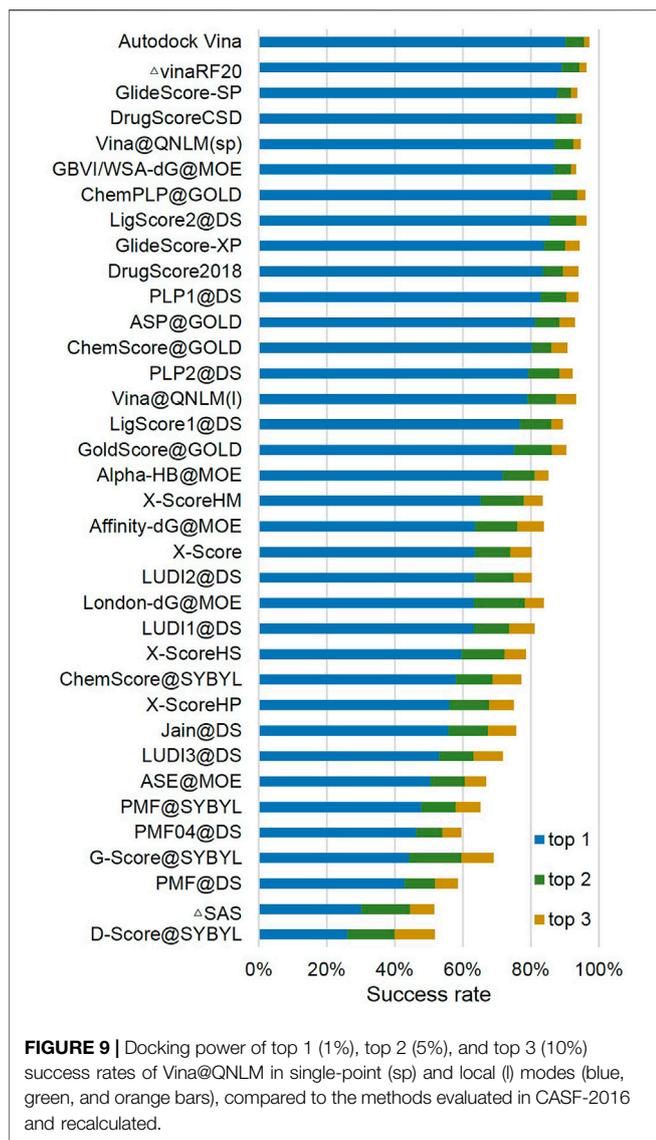


FIGURE 8 | Scoring power of original complex structures and optimized complex structures and Spearman's correlation coefficients of Vina@QNLM in single-point (sp) and local (l) modes (blue and green bars), compared to the methods evaluated in CASF-2016 and recalculated.

Accordingly, the Spearman rank correlation coefficient is shown in **Figure 8**; **Supplementary Tables S8, S9**. Vina@QNLM scoring function ranked fourth and fifth, and its Spearman's rank correlation coefficient was 0.616 and 0.604 in the single-point

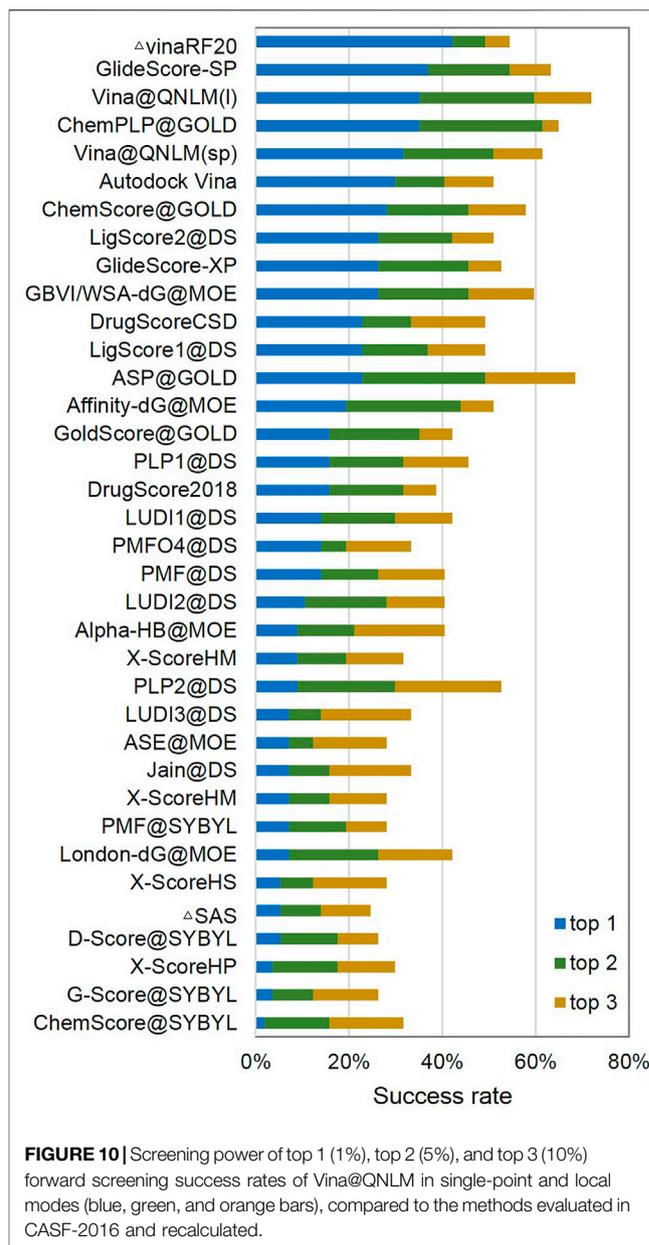
and local modes. After the local optimization of the compound structure, its Spearman's rank correlation coefficients were 0.584 and 0.612, respectively, ranking 10th and second in the single-point and local model.



Docking Power and Screening Power

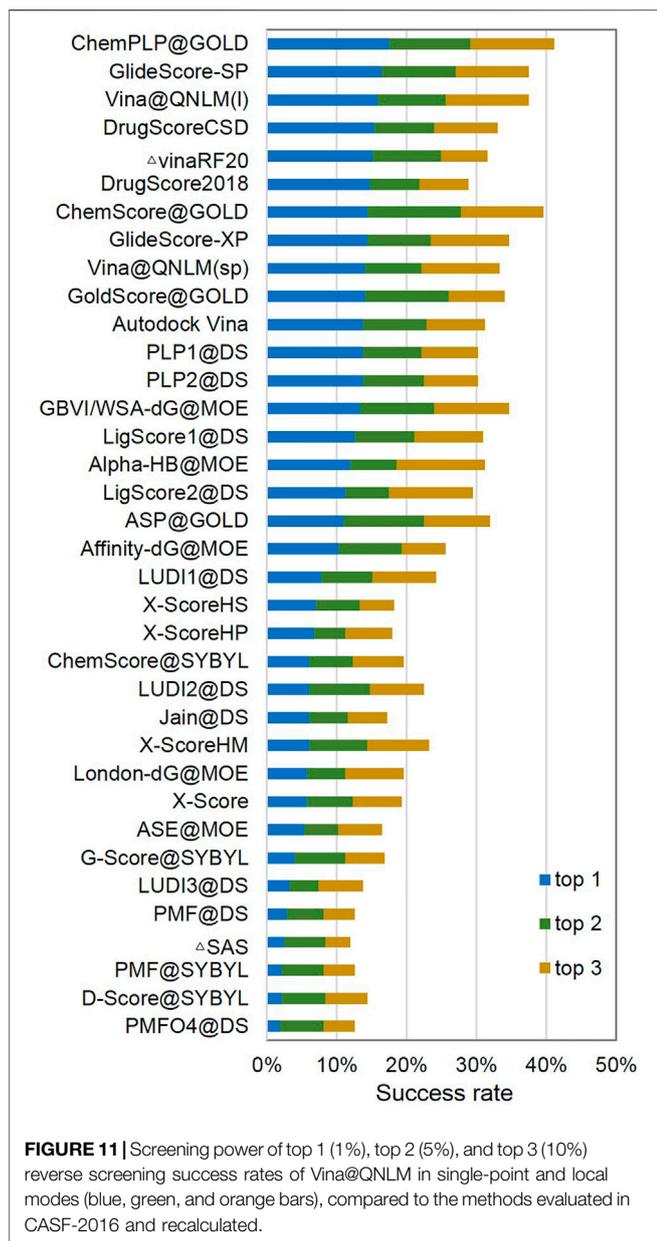
In the docking power test, each scoring function detected the success rate of the native ligand-binding site (RMSD <2.0 Å). Similar to the CASF-2013 dataset, we also divided the docking ability into three parts, namely, top 1 (1%), top 2 (5%), and top 3 (10%). AutoDock Vina showed superior performance in all the three sections. Vina@QNLM is in the middle of the top 1% but not far behind in the top 5%. It reached 93.3% in the local mode and 94.70% in the single-point mode (see **Figure 9**; **Supplementary Table S10**).

In CASF-2016, the forward screening power of a scoring function is measured by its success rate of identifying the highest-affinity ligand for a given target protein. The success rates by considering the 1, 5, and 10% candidates in screening for all the scoring functions are illustrated in **Figure 10**. The data for making this figure are presented in **Supplementary Table S11**. In this test, the top five scoring functions, namely, VinaRF20, GlideScore-SP, ChemPLP@GOLD, Vina@QNLM(I), and Vina@QNLM(sp), have success rates above 30% at the top 1%



level. Vina@QNLM(I) has the third highest success rate of 35.10%, only a little worse than those of VinaRF20 and GlideScore-SP. Considering the size of our test set, this scoring function has a chance of 35.10% to rank the highest-affinity ligand among the top three candidates.

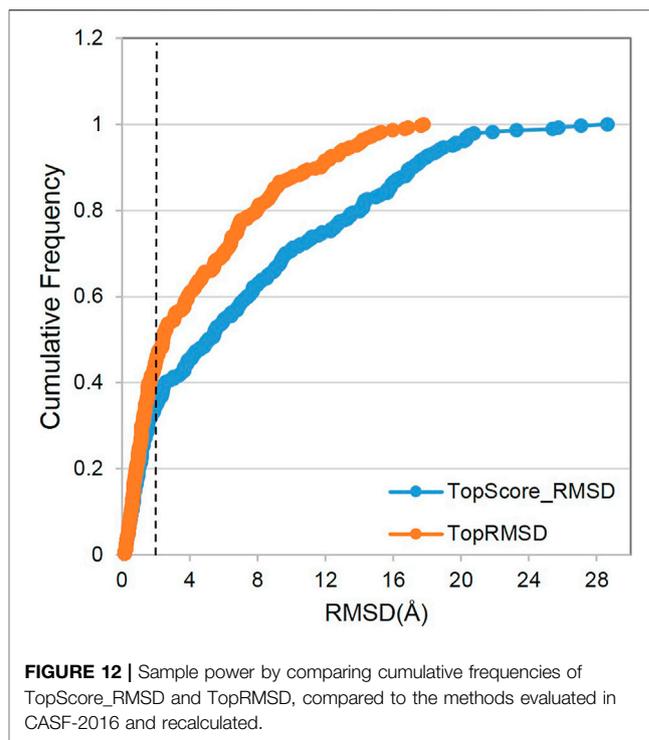
In CASF-2016, the screening power was also evaluated in a reverse screening trial. Here, the reverse screening power of a scoring function was measured by its success rate of identifying the true target protein for a given ligand molecule. The success rates by considering the top 1 (1%), top 5 (5%), and top 10 (10%) candidates in the screening for all the scoring functions are illustrated in **Figure 11**; **Supplementary Table S12**. In this test, the top five scoring functions are ChemPLP@GOLD, GlideScore-SP, Vina@QNLM(I), DrugScoreCSD, and VinaRF20. Their success



rates of selecting the true target protein as the best-scored candidate are over 15%. Here, Vina@QNLN(I) has the third highest success rate of 15.80%, only a little worse than those of ChemPLP@GOLD and GlideScore-SP. These top-ranked scoring functions are also generally top-ranked in the forward screening trial. However, their success rates in the reverse screening trial are only half (or even lower) compared to their performance in the forward screening trial. For example, the success rates of Vina@QNLN(I) are 35.10 and 15.80% in the forward screening and reverse screening trials, respectively.

Sample Power

In the evaluation of the conformation search ability of molecular docking, when the RMSD between the simulated value and



experimental value is less than 2.0 Å, the simulation is considered to be effective, ranking fifth in the top magician RMSD mode and sixth in the best magician RMSD mode. The figure indicates that the success rate of Vina@QNLN for the top-scored pose is 38%, but that for the top RMSD pose is slightly higher (43%) (see **Figure 12**). By referring to other molecular docking software, the prediction ability of Vina@QNLN is at a medium level.

Speedup Effect Comparison

The docking time is proportional to the number of active rotatable bonds in a ligand. Therefore, we used two sets of data to measure the acceleration effect, including the number of rotating bonds between 0–6 and 7–12. The elapsed time of molecular docking was measured by calculating the average time per receptor–ligand pair of a single molecule and a single target.

In this study, we used an Intel Core i7-4610 M processor to test the molecular docking times under the X86 platform. The docking time was positively correlated with the number of receptor–ligand pairs, with the figure increasing modestly from 1,976.35 s to approximately 19,378.34 s, and the receptor–ligand pairs were 100 and 800 (see **Supplementary Table S13**). Molecular docking with the number of rotating bonds between 7 and 12 showed the same tendency (see **Supplementary Table S14**).

The docking speed on Sunway's MPE was significantly reduced as compared to the elapsed time of X86 owing to the hardware characteristics of the processor (see **Figure 13A**). Furthermore, we employed different compilation optimization methods provided by Sunway, but the computing time did not significantly improve. As a heterogeneous multicore processor, it

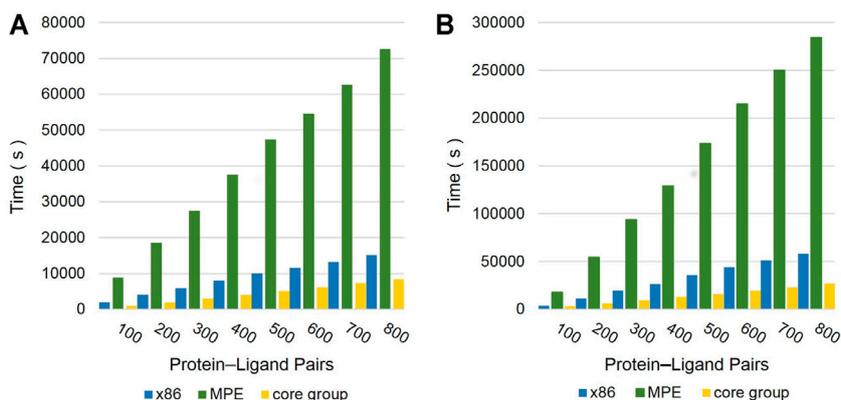


FIGURE 13 | Comparison of the calculation speed in three ways: X86, MPE, and the core group. **(A)** The torsion of the docking molecule is 0–6. **(B)** The torsion of the docking molecule is 7–12.

TABLE 1 | Parallel efficiency of the total elapsed time and docking elapsed time.

Protein–ligand pair	Process	Total elapsed time (s)	Docking elapsed time (s)	Parallel efficiency (docking) (%)	Parallel efficiency (total) (%)
9,59,000	4,80,001	240.66	117.26	55.92	53.68
9,59,000	2,40,501	441.75	250.86	52.17	58.37
9,59,000	1,61,001	467.68	303.00	64.52	82.35
9,59,000	80,501	929.45	495.65	78.88	82.87
9,59,000	41,001	1752.41	844.38	90.91	86.30
9,59,000	20,501	3,319.41	2015.56	76.17	91.12
9,59,000	10,501	5,905.00	2,997.23	100.00	100.00

is important to properly use the core to speed up the computation time. We completed the migration of the algorithm and guaranteed the maximum utilization of the CPE. By calculating the average docking time per receptor–ligand pair of a single molecule and single target, we can observe that the optimized Vina@QNLM elapsed time is faster than that of the optimized Vina@QNLM elapsed time is faster than that of the X86 platform and previous version of Vina@QNLM (Table 1). For both ligands, the number of rotating bonds between 0–6 and 7–12 has a large speedup. Compared with the execution time of the serial version on MPE, speedups of approximately 10× on average and 11.0× at best were achieved.

Parallel Efficiency

As previously described, there are some barriers in the original molecular docking program to heterogeneous computers, such as the Sunway TaihuLight. Each process can be allocated a certain number of molecules to ensure a load balance based on MPI. Therefore, we employed two methods to evaluate the docking efficiency, that is, docking time and overall docking time, which included I/O and the time of sending molecular data. The computing time gradually decreased with the increase in the number of computing processes (see Figure 14).

We randomly selected 9,59,000 small molecules in the Zinc12 (Irwin et al., 2012) library, docked them with 6BFA protein, and recorded the elapsed time. The parallel efficiency of 10,501 processes was 100%, and the parallel efficiency slightly

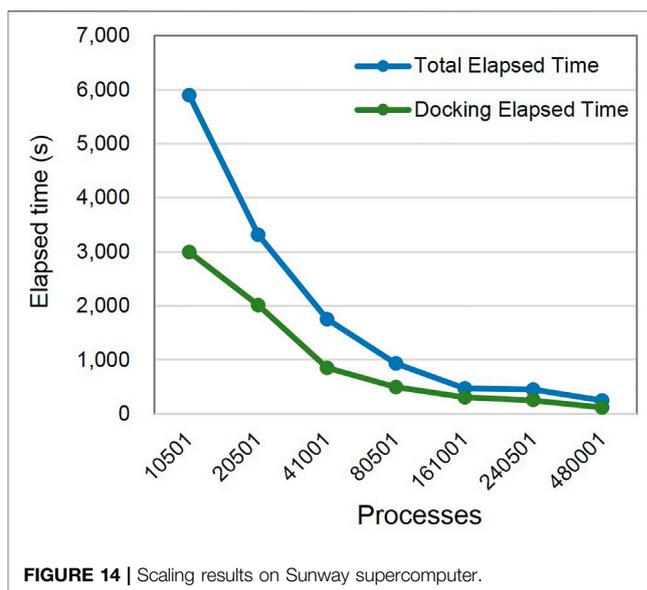


FIGURE 14 | Scaling results on Sunway supercomputer.

dropped with an increase in computing processes. A strong parallel efficiency of up to 55.92% was realized when Vina@QNLM was expanded to 4,80,001 processes. As for the parallel efficiency calculated based on the total elapsed time, the value was

lower than the docking elapsed method, and 4,80,001 processes were used for docking, with a scalability of 53.68% (Table 1). Thus, Vina@QNLM can guarantee good scalability when run on Sunway supercomputer.

Typical Applications

SARS-CoV-2

The global spread of the coronavirus disease continues to harm the public health. To screen molecules with promising medicinal purposes, first, we obtained 837,370,487 molecules in ZINC20 (<http://zinc20.docking.org/tranches/home/>). Then, almost 10 million (10,593,000) of these molecules were randomly selected. The molecules have a LogP between 1 and 4.5 and a molecular weight between 300 and 500. The 10 million molecules were docked with nine targets related to SARS-CoV-2: M^{Pro}, RdRp, RdRp_noMg, Spike-RBD, ACE2, NSP16, PL^{Pro}1, PL^{Pro}2, and X-domain. We used 161,001 processes (10,465,065 cores) for virtual screening and completed 95,337,000 times docking in 8.5 h. Some of the molecules were selected for further molecular dynamic experiments and clinical trials. The docking results for the top 1,000 compounds at each target can be seen in Supplementary Material result_Top1000.zip.

TRIP13 Target

Another typical application is the docking demand from the Shanghai Institute of Materia Medica of the Chinese Academy of Sciences. The Thyroid Hormone Receptor Interacting Protein 13 (TRIP13) target plays an important role in the preparation of tumor therapy drugs (Wang et al., 2019). Accordingly, Vina@QNLM is used to complete the docking of 100 million compounds with the TRIP13 target in 12 h on a parallel scale of 1,00,000 processes.

AVAILABILITY OF VINA@QNLM

To ensure that the vast number of users can use our software, we developed an online system (<http://vina.qnlm.ac/index.html>). General users can submit docking tasks after registration and can use Vina@QNLM for molecular docking after being reviewed by the administrator.

CONCLUSION

Molecular docking is important in the field of drug discovery. Large-scale and efficient molecular docking procedures can significantly shorten the lead compound discovery time. In this study, we propose an ultra-large molecular docking scheduling mechanism. We adopted the Vinardo software to fit the hardware architecture of the Sunway processor, which consisted of heterogeneous CGs (1 MPE and 64 CPEs). We realized molecular docking in the CPEs of Sunway's CG through data exchange between the MPE and CPEs. The performance of Vina@QNLM was comprehensively evaluated using the CASF-2013 and CASF-2016 protein–ligand complex

benchmarks and achieved an optimum of 11× speedup. We managed to expand to 10,465,065 cores, including 161,001 MPEs, reaching a strong scalability of 55.92%. Nine targets related to SARS-CoV-2 were docked with more than 10 million molecules, with a total of 95,337,000 times docking completed in 8.5 h. We also made the top 1,000 best-rated compounds available to the public for targets related to SARS-CoV-2 and developed a platform for ordinary users to submit docking tasks. Our ultra-large molecular docking scheduling mechanism and optimization methods can inspire other software adaptations in Sunway and other heterogeneous multicore supercomputers.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, and further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

HL performed the redesigning and optimization of the software, carried out parallel experiments, and edited the manuscript. CW optimized part of the code. ZW and JG participated in the implementation of the CASF experiment and the drawing of some pictures. YZ wrote part of the script. HL and ZW designed the architecture of the work and modified the paper.

FUNDING

This work was financially supported by the Ultra Large-Scale Intelligent Computing System for Drug Development (2018YFB0****04), the Qingdao Independent Innovation Major Project (20-3-2-12-xx), the Marine S&T Fund of Shandong Province for Pilot National Laboratory for Marine Science and Technology (Qingdao) (No.2018SDKJ0402), and Major Projects for Replacing Old Growth Drivers with New Ones in Shandong Province (2019-3-23).

ACKNOWLEDGMENTS

The authors would like to thank ZZ, XW, and MY for their advice and discussion. The authors thank the National Supercomputing Centre in Wuxi for the suggestions on performance optimizations.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fchem.2021.750325/full#supplementary-material>

REFERENCES

- Allen, W. J., Balius, T. E., Mukherjee, S., BrozellBrozell, Demetri T. MoustakasLang, S. R. P. Therese., Moustakas, D. T. Irwin. D. Kuntz., Lang, P. T., et al. (2015). DOCK 6: Impact of New Features and Current Docking Performance. *J. Comput. Chem.* 36 (15), 1132–1156. doi:10.1002/jcc.23905
- Altuntaş, Serkan., Bozkus, Zeki., and Fragueta, Basilio. B. (2016). “GPU Accelerated Molecular Docking Simulation with Genetic Algorithms,” in *In Applications of Evolutionary Computation*. Editors Giovanni. Squillero and Paolo. Burelli (New York, US: Springer International Publishing), 134–146.
- Chen, J., Lin, H., Liang, W., Yu, Y., Han, W., An, H., Chen, Y., and Liu, X. (2017). . IEEE, 1031–1038. doi:10.1109/ISPA/IUCC.2017.00157/Refactoring the Molecular Docking Simulation for Heterogeneous, Manycore Processors Systems In 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC) Guangzhou China
- Dong, D., Xu, Z., Zhong, W., and Peng, S. (2018). Parallelization of Molecular Docking: A Review. *Ctmc* 18 (12), 1015–1028. doi:10.2174/1568026618666180821145215
- Ellingson, S. R., Smith, J. C., and Baudry, J. (2013). VinaMPI: Facilitating Multiple Receptor High-Throughput Virtual Docking on High-Performance Computers. *J. Comput. Chem.* 34 (25), 2212–2221. doi:10.1002/jcc.23367
- Fu, H., Liao, J., Yang, J., Wang, L., Song, Z., Huang, X., et al. (2016). The Sunway TaihuLight Supercomputer: System and Applications. *Sci. China Inf. Sci.* 59 (7), 072001. doi:10.1007/s11432-016-5588-7
- Gao, P., Duan, X., Zhang, T., Zhang, M., Schmidt, B., Zhang, X., et al. (2020). Millimeter-Scale and Billion-Atom Reactive Force Field Simulation on Sunway TaihuLight. *IEEE Trans. Parallel Distrib. Syst.* 31 (12), 2954–2967. doi:10.1109/TPDS.2020.3008499
- Hu, W., Qin, X., Jiang, Q., Chen, J., An, H., Jia, W., et al. (2021). High Performance Computing of DGDFT for Tens of Thousands of Atoms Using Millions of Cores on Sunway TaihuLight. *Sci. Bull.* 66 (2), 111–119. doi:10.1016/j.scib.2020.06.025
- Irwin, J. J., SterlingBolstad, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. (2012). ZINC: A Free Tool to Discover Chemistry for Biology. *J. Chem. Inf. Model.* 52 (7), 1757–1768. doi:10.1021/ci3001277
- Li, J., Fu, A., and Zhang, L. (2019). An Overview of Scoring Functions Used for Protein-Ligand Interactions in Molecular Docking. *Interdiscip. Sci. Comput. Life Sci.* 11 (2), 320–328. doi:10.1007/s12539-019-00327-w
- Li, Y., Han, L., Liu, Z., and Wang, R. (2014). Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results. *J. Chem. Inf. Model.* 54 (6), 1717–1736. doi:10.1021/ci500081m
- Li, Y., Su, M., Liu, Z., Li, J., Liu, J., Han, L., et al. (2018). Assessing Protein-Ligand Interaction Scoring Functions with the CASF-2013 Benchmark. *Nat. Protoc.* 13 (4), 666–680. doi:10.1038/nprot.2017.114
- Liu, X., Lu, Z., Yuan, W., Ma, W., and Zhang, J. (2020). Massively Parallel CFD Simulation Software: CCFD Development and Optimization Based on Sunway TaihuLight. *Scientific Programming* 2020 (July), 1–17. doi:10.1155/2020/8847481
- Nocedal, Jorge., and Wright, Stephen. J. (1999). *Numerical Optimization*. in. New York US: Springer Series in Operations Research. Springer.
- Pagadala, N. S., Syed, K., and Tuszynski, J. (2017). Software for Molecular Docking: A Review. *Biophys. Rev.* 9 (2), 91–102. doi:10.1007/s12551-016-0247-1
- Quiroga, R., and Villarreal, M. A. (2016). Vinardo: A Scoring Function Based on Autodock Vina Improves Scoring, Docking, and Virtual Screening. *PLOS ONE* 11 (5), e0155183. doi:10.1371/journal.pone.0155183
- Su, M., Yang, Q., Du, Y., Feng, G., Liu, Z., Li, Y., et al. (2019). Comparative Assessment of Scoring Functions: The CASF-2016 Update. *J. Chem. Inf. Model.* 59 (2), 895–913. doi:10.1021/acs.jcim.8b00545
- Thomsen, R., and Christensen, M. H. (2006). MolDock: A New Technique for High-Accuracy Molecular Docking. *J. Med. Chem.* 49 (11), 3315–3321. doi:10.1021/jm051197e
- Trott, O., and Olson, A. J. (2009). AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *J. Comput. Chem.* 31 (2), 455–461. doi:10.1002/jcc.21334
- Tsutsui, S., and Collet, P. (2013). *Massively Parallel Evolutionary Computation on GPGPUs*. New York US: Springer Publishing Company.
- Wang, K., Sturt-Gillespie, B., Hittle, J. C., Macdonald, D., Chan, G. K., Yen, T. J., et al. (2019). Correction: Thyroid Hormone Receptor Interacting Protein 13 (TRIP13) AAA-ATPase Is a Novel Mitotic Checkpoint-Silencing Protein. *J. Biol. Chem.* 294 (25), 10019. doi:10.1074/jbc.AAC119.009554
- Wang, Z., Sun, H., Yao, X., Li, D., Xu, L., Li, Y., et al. (2016). Comprehensive Evaluation of Ten Docking Programs on a Diverse Set of Protein-Ligand Complexes: the Prediction Accuracy of Sampling Power and Scoring Power. *Phys. Chem. Chem. Phys.* 18 (18), 12964–12975. doi:10.1039/C6CP01555G
- Xiao, Z., Liu, X., Xu, J., Sun, Q., and Gan, L. (2021). Highly Scalable Parallel Genetic Algorithm on Sunway Many-Core Processors. *Future Generation Comput. Syst.* 114 (January), 679–691. doi:10.1016/j.future.2020.08.028
- Zhang, X., Wong, S. E., and Lightstone, F. C. (2013). Message Passing Interface and Multithreading Hybrid for Parallel Molecular Docking of Large Databases on Petascale High Performance Computing Machines. *J. Comput. Chem.* 34 (11), 915–927. doi:10.1002/jcc.23214

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

Copyright © 2021 Lu, Wei, Wang, Guo, Zhou, Wang and Liu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.