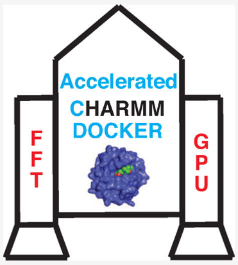
JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

Accelerated CDOCKER with GPUs, Parallel Simulated Annealing, and Fast Fourier Transforms

Xinqiang Ding, Yujin Wu, Yanming Wang, Jonah Z. Vilseck, and Charles L. Brooks, III*

Cite This: https://dx.doi.org/10.1021/acs.jctc.0c00145 Read Online



ABSTRACT: Fast Fourier transform (FFT)-based protein ligand docking together with parallel simulated annealing for both rigid and flexible receptor docking are implemented on graphical processing unit (GPU) accelerated platforms to significantly enhance the throughput of the CDOCKER and flexible CDOCKER — the docking algorithms in the CHARMM program for biomolecule modeling. The FFT-based approach for docking, first applied in protein−protein docking to efficiently search for the binding position and orientation of proteins, is adapted here to search ligand translational and rotational spaces given a ligand conformation in protein−ligand docking. Running on GPUs, our implementation of FFT docking in CDOCKER achieves a 15 000 fold speedup in the ligand translational and rotational space search in protein−ligand docking problems. With this significant speedup it becomes practical to exhaustively search ligand translational and rotational space when docking a rigid ligand into a protein receptor. We demonstrate in this paper that this provides an efficient way to calculate an upper bound for docking accuracy in the assessment of scoring functions for protein−ligand docking, which can be useful for improving scoring functions. The parallel molecular dynamics (MD) simulated annealing, also running on GPUs, aims to accelerate the search algorithm in CDOCKER by running MD simulated annealing in parallel on GPUs. When utilized as part of the general CDOCKER docking protocol, acceleration in excess of 20 times is achieved. With this acceleration, we demonstrate that the performance of CDOCKER for redocking is significantly improved compared with three other popular protein−ligand docking programs on two widely used protein ligand complex data sets: the Astex diverse set and the SB2012 test set. The flexible CDOCKER is similarly improved by the parallel MD simulated annealing on GPUs. Based on the results presented here, we suggest that the accelerated CDOCKER platform provides a highly competitive docking engine for both rigid-receptor and flexible-receptor docking studies and will further facilitate continued improvement in the physics-based scoring function employed in CDOCKER docking studies.

1. INTRODUCTION

Protein−ligand docking methods aim to predict how ligands bind with a target protein, i.e., binding poses of ligands and their binding affinities.[1] They are widely employed in drug discovery processes to virtually screen libraries of a large number of small molecules to search for hit compounds that could provide a basis for the development of novel small molecule therapeutics.[2] Multiple off-the-shelf protein−ligand docking programs, either commercial or free, are available for use,[3] such as CDOCKER,[4] Autodock,[5] Autodock Vina,[6] DOCK,[7] and Glide.[8,9] Most protein−ligand docking programs consist of two essential components: a scoring function for the assessment and ranking of ligand-binding poses and a search algorithm that facilitates the search for and discovery of those low free energy binding poses.[4] The scoring function quantifies the fit between a ligand's binding pose and the protein receptor and is expected to be able to differentiate the correct binding pose from incorrect ones through the assertion that the correct binding pose has the best score. When used to predict binding affinities, the scoring function is also expected to approximate the binding free energy between ligands and target proteins. The search algorithm is utilized to sample potential ligand binding poses and identify the binding pose with the best score. Because the scoring functions used in protein−ligand docking programs are not convex functions and typically have multiple local minima, heuristic search algorithms such as genetic algorithms and simulated annealing are often used in search for optimal binding poses.[4,6]

CDOCKER,[4] a CHARMM[10] module for protein−ligand docking, is one of the protein−ligand docking programs that are widely used in both academia and industry for drug discovery. It uses the interaction energies between proteins and ligands calculated with the CHARMM force field[10] for proteins and the CGenFF force field[11] for ligands as its scoring function. To search for the lowest energy pose of ligands, CDOCKER utilizes molecular dynamics (MD) based simulated annealing followed by energy minimization. In the MD based simulated annealing, MD is used to simulate the dynamics of protein−ligand interaction, and the temperature of the MD first increases to a high value and then slowly

Received: February 12, 2020
Published: May 6, 2020

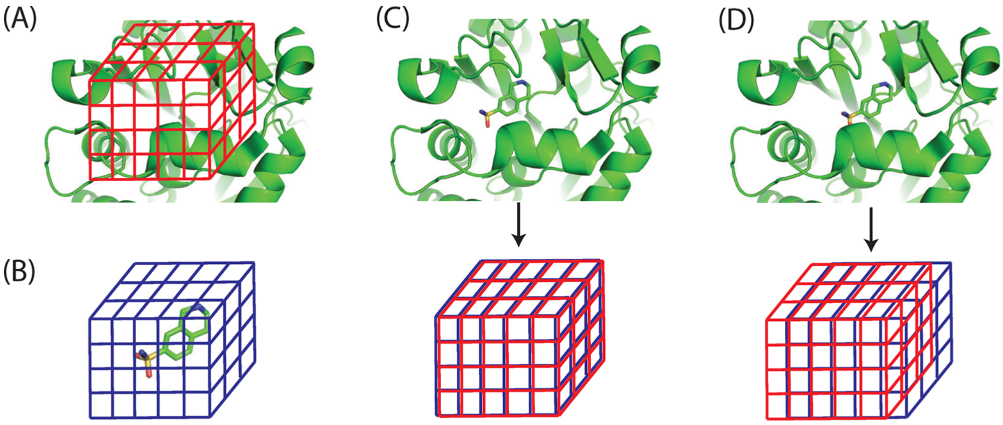


Figure 1. Electrostatic interaction energy between proteins and ligands can be calculated as a cross-correlation function between the protein electrostatic potential grid and the ligand charge grid. (A) The binding pocket in the protein is discretized into a cubic grid with equally spaced grid points. (B) Charges of ligand atoms are distributed onto a cubic grid which has the same spacing and the same number of grid points as the potential grid in panel A. (C and D) As the ligand translates within the binding pocket by multiple units of the spacing distance, the electrostatic interaction energy can be approximated using a cross-correlation function $U_{\text{elec}}(i, j, k)$ between the protein potential grid (red grids) and the ligand charge grid (blue grids). Panels C and D correspond to the cases $U_{\text{elec}}(0, 0, 0)$ and $U_{\text{elec}}(1, 0, 0)$, respectively.

decreases. As the temperature of the MD decreases, ligands are expected to adopt low energy poses. The resulting ligand poses from simulated annealing are further optimized by energy minimization. As the MD-based simulated annealing is a heuristic search approach, it is not guaranteed that the ligand will converge to the lowest energy pose in each trial of MD-based simulated annealing. To increase the chance that the lowest energy pose of the ligand is identified, multiple trials of simulated annealing are employed. In each trial, the ligand is first initialized with a random conformation,[12] a random orientation, and a random position within the binding pocket before going through the MD-based simulated annealing and energy minimization. After the energy minimization, the resulting poses, one from each trial, are ranked by their energies that include the intrainteraction energy of the ligand and the interaction energy between the ligand and the protein. The pose with the lowest energy is predicted to be the binding pose. In a typical application of CDOCKER, a large number of ligands need to be docked with a protein. Therefore, the docking procedure has to run fast enough to make the method practical. To accelerate the docking procedure and help search for the lowest energy pose of ligands, CDOCKER utilizes a cubic grid representation of the binding pocket and soft-core potentials,[1,4] respectively, both of which will be described in detail in following sections.

In this paper two new features, fast Fourier transform (FFT)[13] docking and parallel MD simulated annealing, are added to CDOCKER to further accelerate the search algorithm in CDOCKER. Both features are implemented such that they can take advantage of the parallel computing power of graphical processing units (GPUs). In addition, the original CDOCKER routine used for computing protein grid potentials is also updated such that it can run on GPUs.

2. METHODS

2.1. Accelerated Routine for Computing Soft-Core Grid Potentials. In CDOCKER's docking protocol, most of the computational time is spent on calculating forces on ligand atoms and the ligand's interaction energy with the protein for a large number of ligand poses. To accelerate the force and energy calculation, a cubic grid representation of the binding pocket is used. Specifically, the binding pocket inside a protein is discretized onto a cubic grid (Figure 1A). Probe atoms are placed on each of the grid points, and their interaction energies with the protein are saved in a lookup table. Then the interaction energy of a ligand atom with the protein can be rapidly calculated by looking up values in the tables instead of explicitly calculating its interaction with all of the protein atoms. When a ligand atom's position is not on any grid points, its interaction energy with protein atoms is calculated using a trilinear interpolation of the energy values on the eight grid points of the cell which contains the ligand atom. The force from protein atoms on a ligand atom is approximated using the local energy gradient. To accurately approximate the interaction energy between different ligand atoms and a protein, multiple grids are needed and calculated based on the protein. One of the grids is associated with the electrostatic interaction energy and the remaining grids are associated with the van der Waals interaction energy for ligand atoms with different atom types. In total, 26 grid potentials are computed based on the structure of the protein receptor in CDOCKER.

Soft-core potentials in CDOCKER are used to smooth the energy landscape, which can help the MD-based simulated annealing search escape from local minima and identify the ligand pose with the lowest energy. Specifically, when using soft-core potentials, the van der Waals, electrostatic attractive, and electrostatic repulsive energies are approximate using the formula:

$$E_{ij} = E_{\max} - a \cdot r_{ij}^{b} \text{ if } |E_{ij}^{*}| > \frac{|E_{\max}|}{2} \tag{1}$$

where E_{ij}^{*} is the regular interaction energy; $E_{\max}$ is a parameter controlling the "softness" of the potential. Given $E_{\max}$, a and b are automatically determined using the condition that the energy and the force calculated using eq 1 have to be equal to that calculated using the regular formula at the switch distance where $|E_{ij}^{*}| = |E_{\max}|/2$.

In the previous version of CDOCKER, the routine for computing soft-core grid potentials for the protein receptor ran on central processing units (CPUs), which was slow due to the large number of grid points and protein atoms. For instance, to compute 26 grid potentials, each of which has 43 × 43 × 43 lattice points, for a target protein of 8757 atoms, the number of floating point operations is on the order of 43 × 43 × 43 × 26 × 8, 757 = 18, 102, 312, 774. It took 2100 s in the previous implementation running on a CPU (Intel Xeon Processor E5520) to compute these grid potentials. Here, we added a parallel implementation of the routine such that it can calculate the soft-core grid potentials on GPUs. To do the same computation it only takes 2.5 s for the new implementation running on a GPU (NVIDIA GEFORCE GTX 1080). Therefore, our new implementation on GPUs is more than 800 times faster than the previous implementation on CPUs. The details of the new implementation on GPUs are included in the Supporting Information.

2.2. Fast Fourier Transform Docking. The FFT approach for docking was first used in rigid protein−protein docking.[14] In this approach, proteins are represented as three-dimensional grids such that the surface complementarity of two proteins can be formulated as the cross-correlation function between two grids.[14] Calculating the cross-correlation function between two grids can be greatly accelerated using the FFT algorithm.[15] Since its first use in protein−protein docking,[14] the FFT approach has been extended and improved in several aspects. In addition to the original potential term representing protein shape complementarity,[14] potential terms representing desolvation and electrostatic interactions were added into the scoring function[16−18] to more accurately model the physical interaction between proteins. Moreover, the FFT approach was further accelerated by using spherical polar Fourier correlations[19−21] and GPUs.[22,23] With these extensions and improvements, the FFT approach has been widely adopted in multiple protein−protein docking programs.[18,20,24]

In contrast to FFT's wide application in protein−protein docking, its application in protein−ligand docking is largely unexplored.[25] One difficulty in adopting the FFT approach for protein−ligand docking is to represent the scoring function as a cross-correlation function between grids, because the scoring function used in protein−ligand docking is often more complicated than that in protein−protein docking. In addition, the FFT approach assumes both protein and ligand are rigid bodies, whereas in protein−ligand docking, at least the ligand needs to be modeled as flexible. Therefore, multiple FFTs are required to search the ligand's conformation space. This in turn requires a fast implementation of the FFT for it to be practical. Otherwise running multiple FFTs will take too much time.

Here we investigated the use of the FFT approach for protein−ligand docking in the context of CDOCKER where the CHARMM force field[10,11] is used as the scoring function. The interaction energy, including electrostatic and van der Waals energy, between proteins and ligands is represented as the sum of multiple cross-correlation functions between multiple pairs of grids and the calculation of cross-correlation functions is accelerated using FFTs. Moreover, calculating multiple FFTs is further accelerated using GPUs.

2.2.1. Representing the Nonbonded Interaction Energy between Proteins and Ligands As Cross-Correlation Functions between Grids. In order to use the FFT approach for protein−ligand docking, the interaction energy between a protein and a ligand needs to be expressed as a set of cross-correlation functions between grids. Because CDOCKER uses the CHARMM force field[10,11] as its scoring function, the interaction energy between proteins and ligands includes electrostatic and van der Waals interactions terms.[4]

The electrostatic interaction energy between proteins and ligands is calculated as

$$U_{\text{elec}} = \sum_{i \in L} \sum_{j \in P} \frac{1}{4\pi\epsilon} \frac{q_i q_j}{|r_i - r_j|} = \sum_{i \in L} q_i \sum_{j \in P} \frac{1}{4\pi\epsilon} \frac{q_j}{|r_i - r_j|} = \sum_{i \in L} q_i V_{\text{elec}}(r_i) \tag{2}$$

where L and P are collections of ligand atoms and protein atoms, respectively, q_i and q_j are atom partial charges, and r_i and r_j are atom coordinates. $V_{\text{elec}}(r_i) = \sum_{j \in P} \frac{1}{4\pi\epsilon} \frac{q_j}{|r_i - r_j|}$ is the protein electrostatic potential at position r_i. As eq 2 shows, the electrostatic interaction energy between protein and ligand atoms can be calculated as an inner-product between the ligand charge vector $q_L = (q_i)_{i \in L}$ and the protein electrostatic potential vector $V_{\text{elec}} = (V_{\text{elec}}(r_i))_{i \in L}$. However, the protein electrostatic potential vector V_{elec} still depends on the positions of the ligand atoms that are not known in advance. To get rid of this dependency, grid representations are used for both the protein electrostatic potential and the charges of ligand atoms (Figure 1). Specifically, the binding pocket of a protein is discretized using a three-dimensional grid and protein electrostatic potentials at all of the grid points are calculated and saved in a lookup table (Figure 1A). The protein electrostatic potential at the grid point (l, m, n) is represented as $V_{\text{elec}}^{\text{grid}}(l,m,n)$. Because the protein electrostatic potential is calculated only at the grid points, in order to calculate the electrostatic interaction energy between proteins and ligands, the partial charges of ligand atoms are also distributed onto a three-dimensional grid (Figure 1B) in a trilinear manner. The aggregated charge at the grid point (l, m, n) is represented as $Q^{\text{grid}}(l, m, n)$. Then the electrostatic interaction energy between protein atoms and ligand atoms can be approximated using the inner-product of protein electrostatic potential grid and ligand charge grid (Figure 1C):

$$U_{\text{elec}} \approx \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \sum_{n=0}^{N_z-1} Q^{\text{grid}}(l, m, n) V_{\text{elec}}^{\text{grid}}(l, m, n) \tag{3}$$

where N_x, N_y, and N_z are the numbers of grid points along the X, Y, and Z directions, respectively. Moreover, when the ligand is translated within the binding pocket by i, j, and k grid spacing units in the X, Y, and Z directions, respectively, the electrostatic potential energy between the protein and ligand can be similarly approximated using (Figure 1D):

$$U_{\text{elec}}(i, j, k) \approx \sum_{l=0}^{N_x-1} \sum_{m=0}^{N_y-1} \sum_{n=0}^{N_z-1} Q^{\text{grid}}(l, m, n) V_{\text{elec}}^{\text{grid}}(l + i, m + j, n + k) \tag{4}$$

where $V_{\text{elec}}^{\text{grid}}$ is extended into a periodic grid, i.e., $V_{\text{elec}}^{\text{grid}}(l,m,n) = V_{\text{elec}}^{\text{grid}}(l \pmod{N_x}, m \pmod{N_y}, n \pmod{N_z}))$. As shown in eq 4, as the ligand moves within the binding pocket by distances of multiple units of grid spacing in each direction, the electrostatic interaction energy between the protein and ligand can be approximated as a cross-correlation function between the protein electrostatic potential grid $V_{\text{elec}}^{\text{grid}}$ and the ligand charge grid Q^{grid}. An advantage of using the grid representation, as in eq 4, over that in eq 2 is that $V_{\text{elec}}^{\text{grid}}$ is independent of the ligand and can be calculated with only the protein. Similarly, grid Q^{grid} is independent of the protein and can be calculated with only the ligand.

The van der Waals interaction energy between proteins and ligands is calculated using the Lennard-Jones potential:

$$\begin{aligned} U_{\mathrm{vdw}} &= \sum_{i \in L} \sum_{j \in P} \epsilon_{ij} \left[\left(\frac{r_{ij}^{\min}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{ij}^{\min}}{r_{ij}} \right)^{6} \right] \\ &= \sum_{i \in L} \sum_{j \in P} \sqrt{\epsilon_i \epsilon_j} \left[\left(\frac{(r_i^{\min} + r_j^{\min})/2}{|r_i - r_j|} \right)^{12} \right. \\ &\qquad \left. - 2 \left(\frac{(r_i^{\min} + r_j^{\min})/2}{|r_i - r_j|} \right)^{6} \right] \\ &= \sum_{i \in L} \sqrt{\epsilon_i} \cdot V_{\mathrm{vdw}}(r_i, r_i^{\min}), \end{aligned} \tag{5}$$

where $r_i^{\min}$, $r_j^{\min}$, ϵ_i, and ϵ_j are parameters of the Lennard-Jones potential and are parts of the CHARMM force field, and

$$\begin{aligned} V_{\mathrm{vdw}}(r_i, r_i^{\min}) = \sum_{j \in P} \sqrt{\epsilon_j} &\left[\left(\frac{(r_i^{\min} + r_j^{\min})/2}{|r_i - r_j|} \right)^{12} \right. \\ &\left. - 2 \left(\frac{(r_i^{\min} + r_j^{\min})/2}{|r_i - r_j|} \right)^{6} \right] \end{aligned} \tag{6}$$

Equation 5 for the van der Waals energy is similar to that in eq 2, except that $V_{\mathrm{vdw}}(r_i,r_i^{\min})$ depends on both ligand coordinates r_i and parameters $r_i^{\min}$, whereas $V_{\mathrm{elec}}(r_i)$ only depends on ligand coordinates r_i. Because of this difference, the approach used to represent the electrostatic energy between proteins and ligands as a cross-correlation function between a pair of grids can not be directly applied to the van der Waals interaction. In the CHARMM force field, parameters $r^{\min}$ of ligand atoms depend on their atom types, and the total number of atom types is finite. Therefore, there are only a finite number of possible values for $r^{\min}$. Taking advantage of this fact, we can group the terms in eq 5 based on the value of $r^{\min}$:

$$\begin{aligned} U_{\mathrm{vdw}} &= \sum_{i \in L} \sqrt{\epsilon_i} V_{\mathrm{vdw}}(r_i, r_i^{\min}) = \sum_{r^{\min} \in R^{\min}} \sum_{i \in L_{r^{\min}}} \sqrt{\epsilon_i} V_{\mathrm{vdw}}^{r^{\min}}(r_i) \\ &= \sum_{r^{\min} \in R^{\min}} U_{\mathrm{vdw}}^{r^{\min}} \end{aligned} \tag{7}$$

where $R^{\min}$ is the set of possible values of $r^{\min}$ for ligand atoms and $L_{r^{\min}}$ is the set of ligand atoms that have the parameter of $r^{\min}$. The individual van der Waals energy corresponding to $r^{\min}$ is $U_{\mathrm{vdw}}^{r^{\min}} = \sum_{i \in L_{r^{\min}}} \sqrt{\epsilon_i} V_{\mathrm{vdw}}^{r^{\min}}(r_i)$, which is similar to eq 2 and can be calculated as a cross-correlation function between grids using the same approach used for calculating the electrostatic energy. Therefore, the total van der Waals interaction energy can be approximated as the sum of multiple cross-correlation functions between multiple pairs of grids.

2.2.2. Calculating Cross-Correlation Functions between Grids Using FFTs in Parallel on GPUs.

Based on the convolution theorem,[15] the cross-correlation function for the electrostatic energy in eq 4 can be calculated by applying a Fourier transform and an inverse Fourier transform successively on both sides of the equation, i.e.

$$U_{\mathrm{elec}} = \mathcal{F}^{-1}\{\mathcal{F}\{Q^{\mathrm{grid}}\}^{*}\mathcal{F}\{V_{\mathrm{elec}}^{\mathrm{grid}}\}\} \tag{8}$$

The FFT algorithm is utilized to efficiently calculate both the Fourier transform and the inverse Fourier transform operations. In contrast to the naive algorithm which requires $O((N_xN_yN_z)^2)$ number of operations to calculate the cross-correlation function, the FFT algorithm only needs $O((N_xN_yN_z)\log(N_xN_yN_z))$ number of operations. Similarly, the FFT algorithm can also be used to calculate the van der Waals interaction energy in eq 7. Although the FFT algorithm can significantly accelerate the calculation of cross-correlation functions, one cross-correlation function can only provide interaction energies between a proteins and a ligand as the ligand translates within the binding pocket with a fixed conformation and a fixed orientation. In other words, FFTs only accelerate the search of the ligand translational space. However, in protein–ligand docking where at least the ligand is modeled as flexible, the interaction energies need to be calculated for different conformations and orientations of the ligand, in addition to different positions. Therefore, multiple FFTs, each for one particular conformation and orientation of the ligand, are needed in protein–ligand docking. To accelerate this calculation, multiple FFTs are run on GPUs in a batch mode to take advantage of the pipelined parallel computing power of GPUs.[26]

2.3. Parallel MD-Based Simulated Annealing with GPUs.

One of the advances in using MD simulations to study both chemical and biological systems has been the utilization of GPUs.[27−30] Compared with CPUs, the parallel computing power of GPUs enables us to run MD simulations orders of magnitude faster and simulate longer time scale dynamics of chemical and biological systems, which makes MD suitable to study processes that were not accessible before.[27−30] Although GPUs have been widely employed in running MD simulations of large chemical and biological systems, they are rarely used to accelerate protein–ligand docking methods. Here we investigated the utilization of GPU computing to accelerate CDOCKER for protein–ligand docking by running MD-based simulated annealing of multiple copies of ligands and any included flexible receptor regions in parallel on one GPU.

Because protein–ligand interaction energy landscapes have many local minima and the MD-based simulated annealing is a heuristic search method, multiple trials of simulated annealing have to be employed to search for the lowest energy pose. As the number of trials increases, the docking accuracy usually improves until it reaches a plateau. In addition, in a typical application, CDOCKER needs to dock a large number of ligands with a protein. Therefore, accelerating multiple trials of MD-based simulated annealing can help CDOCKER to dock a large number of ligands in a limited time while maintaining docking accuracy. Because trials of MD-based simulated annealing are independent, one way to accelerate the calculation is to run them in parallel with multiple processors. In the existing implementation of CDOCKER, multiple trials of MD-based simulated annealing can be run in parallel with multiple CPUs. Here we introduce a new feature into CDOCKER to enable it to run multiple trials of MD-based simulated annealing simultaneously on GPUs.

As there are already implementations of MD engines running on GPUs, instead of writing a new MD engine specifically for running multiple trials of MD-based simulated annealing on GPUs, we utilize the existing GPU-enabled MD engine that is part of the CHARMM/OpenMM interface.[31] To utilize the MD engine from OpenMM for our purpose, we make a customized system consisting of multiple copies of a ligand and any included flexible receptor regions and one copy of the potential grids of the protein. Atoms in each copy of the ligand and the flexible receptor group interact with atoms in the same copy and the potential grids but do not interact with atoms in all other copies of flexible groups. Therefore, although the system includes multiple copies of the flexible atoms, these copies are independent from each other and the dynamics of each copy is the same as if there is just one copy. Running one trial of MD-based simulated annealing with this customized system on GPUs is equivalent to running multiple trials of simulated annealing using the previous implementation on CPUs.

The parallel MD-based simulated annealing with GPUs is implemented for both rigid receptor and flexible receptor docking.[1] In rigid receptor docking, protein grid potentials are computed using all protein atoms and the customized OpenMM system consists of multiple copies of ligand atoms. In flexible receptor docking, side chains of multiple amino acids near the binding pocket are modeled as flexible regions. The protein atoms of these flexible side chains are modeled the same as ligand atoms, except that they are attached to fixed protein backbone atoms through bonded interactions. These protein atoms of flexible side chains are excluded when computing the protein grid potentials and the customized OpenMM system consists of multiple copies of both ligand atoms and atoms in flexible side chains. More detailed information on flexible receptor docking in CDOCKER is available in the original paper describing flexible CDOCKER.[1]

3. RESULTS

3.1. Benchmark Data Sets and Computational Details. Three sets of protein−ligand complexes are used as benchmark data sets to evaluate the protein−ligand docking methods just described. The Astex diverse set[32] and the SB2012 set[33] are used for evaluating rigid receptor docking. The SEQ17 data set[34] is used for evaluating flexible receptor docking. The Astex diverse set contains 85 diverse high-resolution protein−ligand complexes and has been widely used for benchmarking different protein−ligand docking methods.[32] In this study, 70 of the 85 protein−ligand complexes that do not include cofactors are used. Compared to the Astex diverse set, the SB2012 set[33] is a much larger set of protein−ligand complexes. It contains 1043 protein−ligand complexes, out of which the 1003 complexes that do not have cofactors and can be typed using CGenFF[35] are used in this study. The 1003 protein−ligand complexes from the SB2012 set overlap with 69 out of 70 complexes from the Astex diverse set. Protein−ligand complexes with cofactors are excluded because force field parameters of the cofactors are not readily available in the CHARMM force field. Unlike protein−ligand docking methods that use empirical scoring functions, CDOCKER uses physical interaction energies based on the all-atom CHARMM force field. Cofactors have to be parametrized based on the CHARMM force field in order to be included in docking. Because not all cofactors in the benchmark data sets are readily available in CGenFF, we excluded all protein−ligand complexes with cofactors to enable us to treat complexes in the benchmark data sets in a consistent manner. In focused studies on specific receptors, if cofactors were present they could be consistently included or parametrized and incorporated into docking studies. However, for our purposes of benchmarking here, we have just ignored complexes with cofactors. The SEQ17 data set, which was originally used to benchmark the flexible receptor docking method AutodockFR,[34] contains 17 pairs of apo-holo structures. These 17 systems were selected to represent a wide range of receptors by the original authors and we include them here for our studies.

The MD-based simulated annealing in CDOCKER is conducted in several stages and different stages use protein grid potentials with different softness values, i.e., grid potentials calculated using different $E_{\max}$ (eq 1). Three sets of values for the parameter $E_{\max}$ are used for rigid receptor docking, and they are summarized in Table 1. The stages of simulated annealing for rigid receptor docking are as follows. Using the soft-core grid potential I, the ligand is heated from 300 to 700 K over 3000 MD integration steps with a step size of 1.5 fs and then cooled from 700 to 300 K over 14 000 steps. Then using the soft-core grid potential II, the ligand is further cooled from 500 to 300 K over 7000 steps and then from 400 to 50 K over 3000 steps. Finally, the ligand is minimized using the soft-core grid potential III for 200 steps. The softness of the potentials decreases as the annealing progresses to provide more realistic representations of the underlying protein−ligand interactions.

Table 1. Soft-Core Potentials Used in Rigid Receptor Docking

name	$E_{\max}$(vdw)[a]	$E_{\max}$(att)[a]	$E_{\max}$(rep)[a]
soft-core potential I	0.6	−0.4	8.0
soft-core potential II	3.0	−20.0	40.0
soft-core potential III	100	−100	100

[a] $E_{\max}$(vdw), $E_{\max}$(att), and $E_{\max}$(rep) in the unit of kcal/mol are parameters for the van der Waals, electrostatic attractive, and electrostatic repulsive interactions, respectively.

Similar stages of MD-based simulated annealing are used in flexible receptor docking except that soft-core grid potentials with different softness are used Table 2. Specifically, the soft-core grid potential I is changed and is adopted from the flexible docking protocol outlined by Gagnon et al.[1] to prevent ligands from leaving the binding pocket during the initial searching and heating stages of simulated annealing. The soft-core potential III is designed to give a more native-like energy landscape. Since there are more explicit atoms involved in flexible receptor docking, these values are larger than those used for rigid receptor docking. Detailed values for the softness parameter $E_{\max}$ used in flexible receptor docking are summarized in Table 2. More details on simulated annealing

Table 2. Soft-Core Potentials Used in Flexible Receptor Docking

name	$E_{\max}$(vdw)[a]	$E_{\max}$(att)[a]	$E_{\max}$(rep)[a]
soft-core potential I	15.0	−120.0	−2.0
soft-core potential III	10 000	−10 000	10 000

[a] $E_{\max}$(vdw), $E_{\max}$(att), and $E_{\max}$(rep) in the unit of kcal/mol are parameters for the van der Waals, electrostatic attractive, and electrostatic repulsive interactions, respectively.

procedures used in the present study are available in the CHARMM scripts included in the Supporting Information.

3.2. Fast Fourier Transform Docking. *3.2.1. Energy Calculation Acceleration with FFTs and GPUs.* When a ligand has a fixed conformation and a fixed orientation, its interaction energy with a protein as the ligand translates on grid points can be represented as the cross-correlation function between grids and both FFTs and GPUs are used to accelerate the calculation of these cross-correlation functions. To see the extent to which FFTs and GPUs can accelerate the calculation, we applied the FFT approach to a test example utilizing the protein−ligand complex 1G9V (PDB ID). The ligand in 1G9V has dimensions of 5.8 Å × 14.5 Å × 8.5 Å in the *X*, *Y*, and *Z* directions, respectively. With a grid spacing distance of 0.5 Å, the ligand grid has 13 × 30 × 18 points. The binding pocket is defined as a cubic box with a dimension of 29.5 Å, and the protein potential grid with the same grid spacing distance as the ligand grid has 60 grid points in all three directions. Therefore, within the binding pocket, the ligand has 59 220 = 47 × 30 × 42 possible positions. The interaction energy between the protein and the ligand for all possible positions of the ligand are calculated using three methods: the naive method, which explicitly calculates the interaction energy for each position on a CPU, FFTs running on a CPU, and FFTs running on a GPU. The wall times used by the three methods are summarized in Table 3. Compared with the naive method, the FFT approach with CPUs accelerates the calculation by more than 100 times and running FFTs on GPUs in a batch mode further accelerates the calculation by 140 fold. Overall, compared with the naive method, the speedup of using both FFTs and GPUs is about 15 000 fold.

Table 3. Wall Time Used by the Three Methods: the Naive Method Looping through All Positions on a CPU, FFTs (CPU), and FFTs (GPU) to Calculate Interaction Energies between the Protein and the Ligand in 1G9V for the Ligand's 59 220 Positions

methods	naive (CPU[a])	FFTs (CPU[a])	FFTs (GPU[b])
wall time (seconds)	31.20	0.28	0.002[c]

[a]The CPU is an Intel Xeon Processor E5645 2.4 GHz. [b]The GPU is a NVIDIA GeForce GTX 1080. [c]Multiple FFTs run in parallel on one GPU in batch mode. The wall time is calculated as the wall time used to run one batch of FFTs divided by the batch size which is 100.

3.2.2. Scoring Function Accuracy: Identifying Ligand Native Orientations and Positions. With the acceleration from both FFTs and GPUs for calculating the interaction energy between ligands and proteins, it becomes feasible to systematically search ligand orientations and positions in a reasonable computation time. This, in turn, enables us to investigate the scoring function's accuracy in terms of identifying native orientations and positions given the conformations of both the ligand and the protein. Using the Astex diverse set and the SB2012 set as test sets, we applied the FFT-based approach with GPUs to rigidly dock ligands onto proteins using the native conformations of ligands and proteins. To systematically search the orientation and translation space of ligands, 100 000 randomly sampled orientations of each ligand are used. For each orientation, the ligand's translational space is uniformly covered by a three-dimensional grid with a grid spacing distance of 0.5 Å. The docked pose of a ligand is chosen to be the lowest energy pose among the poses with all possible combinations of sampled orientations and translations.

For both test sets, the docking accuracy first increases as the number of randomly sampled ligand orientations increases and reaches a plateau when 100 000 random orientations are used (Figure 2A). Here the docking accuracy is defined as the percentage of protein−ligand complexes in the benchmark data sets for which the root mean-square-deviation (RMSD) of the lowest energy docked pose is within 2.0 Å of the native pose. Because the scoring function is not perfect, the native pose for all the protein−ligand complexes in the benchmark data sets is not achieved, and hence the accuracy on the Y axis does not reach 100%. The plateau occurs at a docking accuracy of about 66.34% and 66.22% for the Astex diverse set and the SB2012 set, respectively (Figure 2B). When the native orientation is included, in addition to the 100 000 random orientations, the docking accuracy increases to about 67.75% and 67.63% for the Astex diverse set and the SB2012 set, respectively (Figure 2B). It is notable that this small difference suggests that the use of 100 000 rotational samples is sufficiently dense to cover the rotational space. The docking

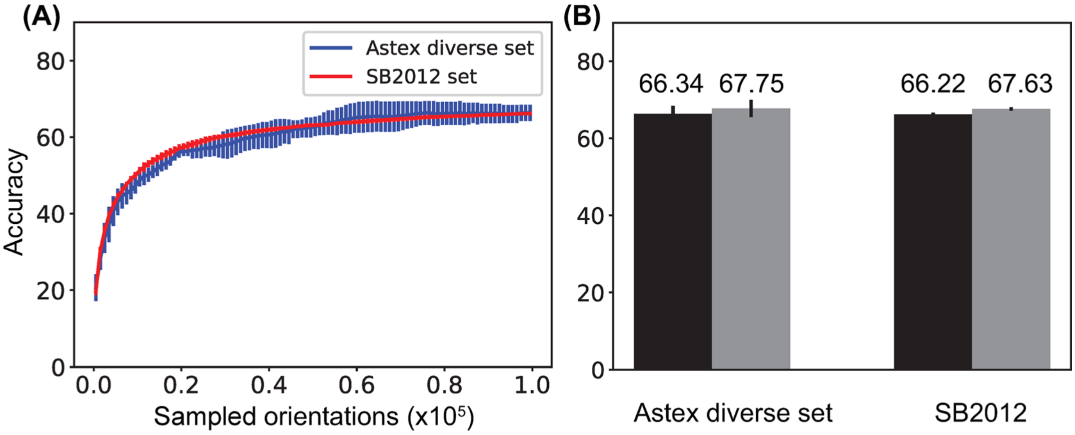


Figure 2. Docking accuracy of the FFT approach for docking rigid ligands onto rigid proteins with the native conformations of both ligands and proteins using the Astex diverse set and the SB2012 set. (A) Docking accuracy increases as the number of randomly sampled orientations increases. The error bars are estimated using 10 independent repeats. (B) Docking accuracy when 100 000 randomly sampled orientations are used (black) and when 100 000 randomly sampled orientations plus the native orientation are used (gray).

accuracy at the plateau, which is around 68%, represents the accuracy of the CHARMM force field in identifying the native orientations and positions of ligands assuming the native conformations of ligands are given. This accuracy should be an upper bound of the CHARMM force field's accuracy in identifying native ligand poses, which includes the native conformations in addition to the native orientations and positions. Although the size of the SB2012 set is more than ten times larger than the Astex diverse set, the results on the two sets are quite similar. The good agreement between different benchmark data sets suggests that the physics-based scoring function used in CDOCKER is potentially transferable among a large variety of protein ligand complexes.

We note that the above FFT-based rigid docking approach could be generalized to permit ligand conformational space to be sampled. This would involve first sampling a suitable ensemble of ligand conformations[36] and then carrying out the rotational/translational sampling to identify the lowest energy conformation using GPU-accelerated FFTs. This protocol can readily be implemented using CHARMM scripting language.[10] However, we instead pursue in the following integration of ligand (and possibly receptor side chain) sampling into an MD simulated annealing scheme as employed in CDOCKER[4] and flexible CDOCKER.[1]

3.3. Parallel MD-Based Simulated Annealing with GPUs.

3.3.1. GPU Accelerated Parallel Simulated Annealing Significantly Accelerates CDOCKER. Compared with the original CDOCKER running serially on CPUs, the speedup of the parallel MD-based simulated annealing with GPUs is shown in Table 4. For the protein−ligand pairs in the Astex diverse set, when 100 and 500 docking trials are used, the average wall time used by the original CDOCKER with CPUs are 338.4 and 1692.0 s, respectively. In contrast, the average wall time used by the parallel MD-based simulated annealing with GPUs are 30.8 and 85.5 s, respectively, which is about 10-fold and 20-fold faster. The speedup becomes even larger when the number of trials increases, because the wall time used by the original CDOCKER on CPUs is proportional to the number of trials.

Table 4. Acceleration of Parallel MD-Based Simulated Annealing with GPUs Compared with the Original CDOCKER with CPUs on the Astex Diverse Set

	CDOCKER with CPUs	CDOCKER with parallel MD-based simulated annealing with GPUs
accuracy[a]	0.623 ± 0.023	0.631 ± 0.029
wall time[b] (seconds)	338.4	30.8
wall time[c] (seconds)	1692.0	85.5

[a] The accuracy when 100 trials are used; The ligand native conformation is used as the starting conformation; The uncertainty is estimated using 10 independent repeats. [b] The wall time used when 100 trials are used. [c] The wall time used when 500 trials are used.

3.3.2. Comparison with Other Protein−Ligand Docking Programs for Rigid Receptor Docking. The accelerated CDOCKER is compared with three other widely used protein−ligand docking programs: Autodock, Autodock Vina, and DOCK. The computational details for setting up docking in all of the docking programs presented here are included in the Supporting Information. To make a fair comparison between different protein−ligand docking programs, the same settings are used for all docking programs whenever it is possible. For instance, for a protein−ligand complex, a docking grid box with the same position and size is used in all programs. The docking accuracy is calculated as the percentage of protein−ligand complexes in benchmark data sets for which the RMSD of the lowest energy docked pose is less than 2.0 Å of the native pose.

The redocking results on the Astex diverse set and the SB2012 set are shown in Tables 5 and 6, respectively. With the acceleration achieved by the parallel MD-based simulated annealing with GPUs in CDOCKER, the average wall time required by CDOCKER for docking one protein−ligand complex is either faster than or on par with other programs. For CDOCKER, Autodock, and Autodock Vina, the docking accuracy one obtains depends on whether the ligands' native conformation or a random conformation is used as the starting conformation: all of these approaches perform better when the ligand native conformation is used as the starting conformation. Starting with ligands' native conformation makes the conformational search easier and the docking accuracies higher than those corresponding to using random starting conformations. Because the DOCK program uses the "anchor and grow" search method,[7] its accuracy does not depend on the starting conformation.

Based on the results from the Astex diverse set, when ligand random conformations are used as starting conformations, DOCK and Autodock Vina have similar and higher docking accuracy. Autodock has the lowest docking accuracy and CDOCKER is in between. Increasing the parameter that controls the searching exhaustiveness in Autodock Vina from 8 to 20 proportionally increases the run time, but it does not change its docking accuracy significantly. Compared with the results on the Astex diverse set (Table 5), the relative performance of the protein−ligand docking programs on the SB2012 set is the same in terms of docking accuracy (Table 6). However, for all of the programs, the docking accuracy is lower on the SB2012 set (Table 6). Although the Astex diverse set contains a diverse set of protein−ligand complexes, the number of protein−ligand complexes in the set is relatively small. Because the SB2012 data set contains more than 1 order of magnitude more protein−ligand complexes, the perform-

Table 5. Docking Accuracy of Multiple Protein-Ligand Docking Programs on the Astex Diverse Set

	CDOCKER[d]	Autodock v4.2.6	Autodock Vina[e]	Autodock Vina[f]	DOCK v6.7
accuracy (native[a])	0.664 (±0.022[g])	0.600 (±0.020)	0.701 (±0.019)	0.710 (±0.009)	0.639 (±0.016)
accuracy (random[b])	0.537 (±0.021)	0.530 (±0.029)	0.633 (±0.014)	0.623 (±0.011)	
wall time (sec)[c]	85.5	279.6	82.3	202.9	50.0

[a] Ligand native conformations are used as starting conformations. [b] Ligand random conformations are used as starting conformations. [c] CDOCKER is run on a GPU (NVIDIA GeForce GTX 980). All of the other docking programs use one CPU (Intel Xeon Processor E5645 2.4 GHz). [d] 500 trials are used in CDOCKER. [e] Exhaustiveness = 8. [f] Exhaustiveness = 20. [g] All uncertainties are estimated using 10 independent repeats.

Table 6. Docking Accuracy of Multiple Protein-Ligand Docking Programs on the SB2012 Set

	CDOCKER[c]	Autodock v4.2.6	Autodock Vina[d]	Autodock Vina[e]	DOCK v6.7
accuracy (native[a])	0.569 (±0.006[f])	0.477 (±0.009)	0.631 (±0.004)	0.642 (±0.005)	0.553 (±0.005)
accuracy (random[b])	0.429 (±0.007)	0.418 (±0.004)	0.532 (±0.004)	0.547 (±0.004)	

[a]Ligand native conformations are used as starting conformations. [b]Ligand random conformations are used as starting conformations. [c]500 trials are used in CDOCKER. [d]Exhaustiveness = 8. [e]Exhaustiveness = 20. [f]All uncertainties are estimated using 10 independent repeats.

ance on the SB2012 set should be a more objective measure of the protein–ligand docking programs' docking accuracy. The lower docking accuracy on the SB2012 set for all of the tested protein–ligand docking programs can be attributed to either search algorithms or scoring functions or both. This suggests that more efforts are required to further improve search algorithms and scoring functions, including both physics based scoring functions used in CDOCKER and DOCK and the empirical scoring functions used in Autodock and Autodock Vina. We note that the docking accuracies of both Autodock Vina and DOCK reported in this study are quite different from those reported in previous studies.[6,7,37] This is because of the fact, as shown in this study, that the docking accuracy of a protein–ligand docking program can vary significantly depending on ligand starting conformation and benchmark data set.

3.3.3. Benchmarking Flexible Receptor Docking. The implementation of parallel simulated annealing on GPUs also significantly accelerates flexible receptor docking in CDOCKER. To test both the docking accuracy and speed of flexible receptor docking in CDOCKER, we used the SEQ17 data set which contains 17 pairs of apo-holo structures and represents a subset of the protein–ligand complexes in the SEQ data set that were able to be successfully docked using AutoDock (RMSD < 2.0 Å) with rigid redocking.[34] These 17 systems were selected to represent a wide range of receptors. For each of the 17 apoprotein structures there is at least one amino acid side chain around the binding pocket whose conformation in the holo structure is different from that in the apo structure by at least 2.5 Å (RMSD). Therefore, there would be at least one severe clash between ligand atoms and the receptor side chain if the ligand native conformation from the protein–ligand complex is directly fit onto the apo structure. In what follows, we evaluate the overall performance (accuracy and speed) of flexible CDOCKER for flexible receptor docking using this data set and compare it with AutodockFR.[34]

In preparing each system for flexible receptor docking, the apo structure is superposed on the corresponding holo structure. The criteria used for selecting flexible side chains is the same as that used in AutoDockFR, except that in flexible CDOCKER the heavy atoms beyond Cα are considered to be flexible, whereas the heavy atoms beyond Cβ are set to be flexible in AutoDockFR.[34] The RMSD cutoff to identify native-like poses is set to be 2.5 Å to be consistent with the evaluation criteria used by AutoDockFR.[34] We performed 10 repeated calculations, each of which consists of 500 trials of both flexible CDOCKER and rigid CDOCKER.

The docking results on the SEQ17 set are shown in Table 7. In each repeat, we first computed the searching accuracy. A searching success is positive when at least one native-like docking pose is identified among the docked poses from 500 trials of docking. The searching accuracy for rigid CDOCKER is 36.47% ± 5.41% when the apo structure is used as the receptor and 95.88% ± 2.84% when the holo structure is used. In contrast, for flexible CDOCKER, the search accuracies are 95.29% ± 2.48% and 80.00% ± 6.90% when the holo structures and the apo structure are used as receptors, respectively (Table 7). Therefore, flexible CDOCKER has a higher searching accuracy than rigid CDOCKER when the receptor binding pocket has a conformational change upon binding with ligands.

Table 7. Docking Accuracy of SEQ17 Dataset Using Flexible CDOCKER

receptor structure	searching accuracy	ranking accuracy
holo	95.29% ± 2.48%	78.82% ± 4.96%
apo	80.00% ± 6.90%	57.06% ± 3.97%

To further assess the accuracy of flexible CDOCKER for this set, we considered the following scoring protocol. The docked poses from 500 trials of docking are clustered using a K-means clustering algorithm from the MMTSB toolset (cluster.pl)[38] based on ligand heavy atoms and an RMSD cutoff of 2.0 Å. If the number of docked poses in a cluster is less than 10 (2%), then the cluster is discarded. This allows us to remove those less populated docking poses that are frequently away from the binding pocket. Then the minimum energy pose of each cluster is selected and ranked based on the scoring function. A scoring success is defined as at least one of the cluster representatives presents as a native-like pose.[34] The scoring accuracy of flexible docking in CDOCKER is about 57.06% when apo structures are used as receptors (Table 7) and the corresponding scoring accuracy of AutodockFR is 70.6%[34] which is higher than CDOCKER. However, as mentioned before, the SEQ17 data set is intentionally designed to only include protein–ligand complexes that can be correctly redocked (RMSD < 2 Å) using the Autodock. If we do the same filtering process for CDOCKER, i.e., exclude protein–ligand complexes that can not be redocked using CDOCKER, which include 3PTE and 2A78, the scoring accuracy of flexible docking in CDOCKER becomes 68.82% ± 3.97%, which is similar to the accuracy achieved by the AutodockFR.

In high-throughput virtual screening, where a large number of ligands need to be docked, the major barrier for the use of flexible receptor docking is its speed. For flexible CDOCKER with parallel simulated annealing running on a GPU (NVIDIA GEFORCE GTX 1080), the average wall time required for one protein–ligand complex is about 1 h. In contrast, AutodockFR running on a CPU (Intel Xeon Processor E5520) requires on average 50 × 8.5 = 425 h for one protein–ligand complex:[34] it takes 8.5 h for AutodockFR to run one round of genetic algorithm evolutioin and 50 rounds of evolution are required. Therefore, flexible CDOCKER based on parallel MD-based simulated annealing on GPUs greatly reduces the amount of time required while maintaining similar docking accuracy for flexible receptor docking.

4. CONCLUSIONS AND DISCUSSIONS

Two new features, fast Fourier transform (FFT) docking and parallel MD-based simulated annealing, are implemented and added to the protein–ligand docking program CDOCKER in

CHARMM. The FFT docking not only utilizes the acceleration provided by FFTs but also employs the parallel computing power of GPUs. Overall, FFT docking with GPUs accelerates the search of ligand positions and orientations by as much as 15,000 fold. With the significant speedup achieved by FFT docking on GPUs, it becomes practical to exhaustively search the translation and rotation space of ligands when docking a rigid ligand into a binding pocket. Although FFT docking alone can not solve the protein−ligand docking problem in which ligands are flexible, the FFT docking can be used to quickly calculate an upper bound of the docking accuracy that can be achieved by a scoring function. This in turn can provide insights into the problems of current scoring functions and help improve the scoring function. In addition, because FFT docking with GPUs can efficiently calculate protein ligand interaction energies for an almost exhaustive list of positions and orientations given a ligand conformation, FFT docking could also be used to explicitly calculate the partition function corresponding to a ligand's translational and rotational space. This can be combined with existing scoring functions in protein−ligand docking to more accurately estimate protein−ligand binding affinities. A similar idea has been investigated by Nguyen et al.[39]

The parallel MD-based simulated annealing with GPUs enables CDOCKER to run about 20 times faster when 500 trials of simulated annealing are used. The speedup becomes even larger when more trials of simulated annealing are employed. With this acceleration, the speed of CDOCKER is on par with or faster than several other popular protein−ligand docking programs tested in this study. In addition, the parallel MD-based simulated annealing on GPUs also enables flexible CDOCKER to achieve significant speed advantages while maintaining similar accuracy for flexible receptor docking. We note that these additions to the CDOCKER and flexible CDOCKER modules described in this paper are part of the c43a1 development version of CHARMM and will be released for general use in the near feature.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.0c00145.

> S1: Details of calculating protein grid potentials in CDOCKER; S2: accelerated calculation of soft-core grid potentials using graphical processing units (GPUs); S3: computational details for rigid receptor docking using CDOCKER, DOCK, Autodock, and Autodock Vina; S4: computational details for flexible receptor docking using CDOCKER; S5: examples of CHARMM script using the four new functionalities introduced in this study (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Charles L. Brooks, III − *Department of Chemistry, Department of Computational Medicine & Bioinformatics, and Biophysics Program, University of Michigan, Ann Arbor, Michigan 48109, United States;* orcid.org/0000-0002-8149-5417;
Email: brookscl@umich.edu

Authors

Xinqiang Ding − *Department of Computational Medicine & Bioinformatics, University of Michigan, Ann Arbor, Michigan 48109, United States;* orcid.org/0000-0002-4598-8732

Yujin Wu − *Department of Chemistry, University of Michigan, Ann Arbor, Michigan 48109, United States*

Yanming Wang − *Department of Chemistry, University of Michigan, Ann Arbor, Michigan 48109, United States;* orcid.org/0000-0002-2383-2470

Jonah Z. Vilseck − *Department of Chemistry, University of Michigan, Ann Arbor, Michigan 48109, United States*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.0c00145

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work is supported by grants from the NIH (GM037554 and GM107233).

■ REFERENCES

(1) Gagnon, J. K.; Law, S. M.; Brooks, C. L., III Flexible CDOCKER: Development and application of a pseudo-explicit structure-based docking method within CHARMM. *J. Comput. Chem.* **2016**, *37*, 753−762.

(2) Jorgensen, W. L. The Many Roles of Computation in Drug Discovery. *Science* **2004**, *303*, 1813−1818.

(3) Sousa, S. F.; Fernandes, P. A.; Ramos, M. J. Protein−ligand docking: current status and future challenges. *Proteins: Struct., Funct., Bioinf.* **2006**, *65*, 15−26.

(4) Wu, G.; Robertson, D. H.; Brooks, C. L., III; Vieth, M. Detailed analysis of grid-based molecular docking: A case study of CDOCKER?A CHARMm-based MD docking algorithm. *J. Comput. Chem.* **2003**, *24*, 1549−1562.

(5) Goodsell, D. S.; Morris, G. M.; Olson, A. J. Automated docking of flexible ligands: applications of AutoDock. *J. Mol. Recognit.* **1996**, *9*, 1−5.

(6) Trott, O.; Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2009**, *31*, 455−461.

(7) Allen, W. J.; Balius, T. E.; Mukherjee, S.; Brozell, S. R.; Moustakas, D. T.; Lang, P. T.; Case, D. A.; Kuntz, I. D.; Rizzo, R. C. DOCK 6: impact of new features and current docking performance. *J. Comput. Chem.* **2015**, *36*, 1132−1156.

(8) Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K.; Shaw, D. E.; Francis, P.; Shenkin, P. S. Glide:? A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **2004**, *47*, 1739−1749.

(9) Halgren, T. A.; Murphy, R. B.; Friesner, R. A.; Beard, H. S.; Frye, L. L.; Pollard, W. T.; Banks, J. L. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J. Med. Chem.* **2004**, *47*, 1750−1759.

(10) Brooks, B. R.; Brooks, C. L., III; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M. CHARMM: The biomolecular simulation program. *J. Comput. Chem.* **2009**, *30*, 1545−1614.

(11) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.;

Mackerell, A. D. CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.* **2009**, *31*, 671−690.

(12) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminf.* **2011**, *3*, 33.

(13) Brigham, E. O.; Brigham, E. O. *The fast Fourier transform and its applications*; Prentice Hall: Englewood Cliffs, NJ, 1988; Vol. *448*.

(14) Katchalski-Katzir, E.; Shariv, I.; Eisenstein, M.; Friesem, A. A.; Aflalo, C.; Vakser, I. A. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl. Acad. Sci. U. S. A.* **1992**, *89*, 2195−2199.

(15) Bracewell, R. N.; Bracewell, R. N. *The Fourier transform and its applications*; McGraw-Hill: New York, 1986; Vol. *31999*.

(16) Gabb, H. A.; Jackson, R. M.; Sternberg, M. J. Modelling protein docking using shape complementarity, electrostatics and biochemical information1. *J. Mol. Biol.* **1997**, *272*, 106−120.

(17) Mandell, J. G.; Roberts, V. A.; Pique, M. E.; Kotlovyi, V.; Mitchell, J. C.; Nelson, E.; Tsigelny, I.; Ten Eyck, L. F. Protein docking using continuum electrostatics and geometric fit. *Protein Eng., Des. Sel.* **2001**, *14*, 105−113.

(18) Chen, R.; Li, L.; Weng, Z. ZDOCK: an initial-stage protein-docking algorithm. *Proteins: Struct., Funct., Genet.* **2003**, *52*, 80−87.

(19) Ritchie, D. W.; Kozakov, D.; Vajda, S. Accelerating and focusing protein−protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics* **2008**, *24*, 1865−1873.

(20) Garzon, J. I.; Lopéz-Blanco, J. R.; Pons, C.; Kovacs, J.; Abagyan, R.; Fernandez-Recio, J.; Chacon, P. FRODOCK: a new approach for fast rotational protein−protein docking. *Bioinformatics* **2009**, *25*, 2544−2551.

(21) Padhorny, D.; Kazennov, A.; Zerbe, B. S.; Porter, K. A.; Xia, B.; Mottarella, S. E.; Kholodov, Y.; Ritchie, D. W.; Vajda, S.; Kozakov, D. Protein−protein docking by fast generalized Fourier transforms on 5D rotational manifolds. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113*, E4286−E4293.

(22) Ritchie, D. W.; Venkatraman, V. Ultra-fast FFT protein docking on graphics processors. *Bioinformatics* **2010**, *26*, 2398−2405.

(23) Sukhwani, B.; Herbordt, M. C. GPU acceleration of a production molecular docking code. *Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, 2009; pp 19−27.

(24) Kozakov, D.; Brenke, R.; Comeau, S. R.; Vajda, S. PIPER: an FFT-based protein docking program with pairwise potentials. *Proteins: Struct., Funct., Bioinf.* **2006**, *65*, 392−406.

(25) Padhorny, D.; Hall, D. R.; Mirzaei, H.; Mamonov, A. B.; Moghadasi, M.; Alekseenko, A.; Beglov, D.; Kozakov, D. Protein−ligand docking using FFT based sampling: D3R case study. *J. Comput.-Aided Mol. Des.* **2018**, *32*, 225−230.

(26) Nvidia, C. *CUFFT library*, 2010.

(27) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine microsecond molecular dynamics simulations with AMBER on GPUs. 2. Explicit solvent particle mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9*, 3878−3888.

(28) Stone, J. E.; Hardy, D. J.; Ufimtsev, I. S.; Schulten, K. GPU-accelerated molecular modeling coming of age. *J. Mol. Graphics Modell.* **2010**, *29*, 116−125.

(29) Hynninen, A.-P.; Crowley, M. F. New faster CHARMM molecular dynamics engine. *J. Comput. Chem.* **2014**, *35*, 406−413.

(30) Eastman, P.; Friedrichs, M. S.; Chodera, J. D.; Radmer, R. J.; Bruns, C. M.; Ku, J. P.; Beauchamp, K. A.; Lane, T. J.; Wang, L.-P.; Shukla, D.; Tye, T.; Houston, M.; Stich, T.; Klein, C.; Shirts, M. R.; Pande, V. S. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **2013**, *9*, 461−469.

(31) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L. P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.* **2017**, *13*, e1005659.

(32) Hartshorn, M. J.; Verdonk, M. L.; Chessari, G.; Brewerton, S. C.; Mooij, W. T.; Mortenson, P. N.; Murray, C. W. Diverse, high-quality test set for the validation of protein- ligand docking performance. *J. Med. Chem.* **2007**, *50*, 726−741.

(33) Mukherjee, S.; Balius, T. E.; Rizzo, R. C. Docking validation resources: protein family and ligand flexibility experiments. *J. Chem. Inf. Model.* **2010**, *50*, 1986−2000.

(34) Ravindranath, P. A.; Forli, S.; Goodsell, D. S.; Olson, A. J.; Sanner, M. F. AutoDockFR: advances in protein-ligand docking with explicitly specified binding site flexibility. *PLoS Comput. Biol.* **2015**, *11*, No. e1004586.

(35) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; Mackerell, A. D. CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.* **2009**, *31*, 671−690.

(36) Vieth, M.; Hirst, J. D.; Brooks, C. L., III Do active site conformations of small ligands correspond to low free-energy solution structures? *J. Comput.-Aided Mol. Des.* **1998**, *12*, 563−572.

(37) Gaillard, T. Evaluation of AutoDock and AutoDock Vina on the CASF-2013 benchmark. *J. Chem. Inf. Model.* **2018**, *58*, 1697−1706.

(38) Feig, M.; Karanicolas, J.; Brooks, C. L., III MMTSB Tool Set: enhanced sampling and multiscale modeling methods for applications in structural biology. *J. Mol. Graphics Modell.* **2004**, *22*, 377−395.

(39) Nguyen, T. H.; Zhou, H.-X.; Minh, D. D. Using the fast fourier transform in binding free energy calculations. *J. Comput. Chem.* **2018**, *39*, 621−636.

https://dx.doi.org/10.1021/acs.jctc.0c00145