Applied Soft Computing Journal 161 (2024) 111732

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Improving the artificial bee colony algorithm with a proprietary estimation of distribution mechanism for protein–ligand docking

Shuangbao Song [a], Cheng Tang [b], Zhenyu Song [c], Jia Qu [a], Xingqian Chen [d,*]

[a] School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213164, China
[b] Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
[c] College of Information Engineering, Taizhou University, Taizhou 225300, China
[d] School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China

ARTICLE INFO

Keywords:
Protein–ligand docking
Artificial bee colony
Estimation of distribution algorithm
Structure-based drug design
Optimization

ABSTRACT

The protein–ligand docking problem plays an essential role in structure-based drug design. The challenge for a protein–ligand docking method is how to execute an efficient conformational search to explore a well-designed scoring function. In this study, we improved the artificial bee colony (ABC) algorithm and proposed an approach called ABC-EDM to solve the protein–ligand docking problem. ABC-EDM employs the scoring function of the classical AutoDock Vina to evaluate a solution during docking simulation. ABC-EDM adopts the search framework of the canonical ABC algorithm to execute conformational search. By further investigating the characteristics of the protein–ligand docking problem, a proprietary search mechanism inspired by estimation of distribution algorithm, i.e., estimation of distribution mechanism (EDM), is designed to enhance the performance of ABC-EDM. To verify the effectiveness of the proposed ABC-EDM, we compare it with three variants of the ABC algorithm, three evolutionary computation algorithms, and AutoDock Vina. The experimental results show that ABC-EDM can effectively solve the protein–ligand docking problem, and it can achieve a success rate 5% higher than AutoDock Vina on the GOLD dataset. This study reveals that taking advantage of problem-specific information about the protein–ligand docking problem to enhance a docking method contributes to solving this problem.

1. Introduction

Proteins are large macromolecules and perform many fundamental biological functions in organisms. Since many diseases are associated with specific proteins, these proteins can serve as therapeutic targets in structure-based drug design. Usually, small molecules (e.g., ligands) that activate or inhibit the function of a target protein and produce a therapeutic benefit can be seen as drugs. Determining how the target proteins interact with small molecules is an important issue in structure-based drug design [1]. Owing to the rapid development of computer science, computational methods have greatly advanced the research of structure-based drug design in recent years [2,3]. Specifically, the protein–ligand docking method has emerged as one of the most typical techniques in drug design. This technique is commonly used in the process of lead discovery and medicinal chemistry optimization, and consequently, it improves the efficiency of drug design and discovery [4].

Protein–ligand docking is based on the assumption that a small molecule (ligand) exerts its biological activity by specific binding to the protein receptor. The principles of protein–ligand docking are usually explained as the 'lock and key' mechanism and 'hand and glove' concept [5]. The protein–ligand docking problem is conceptually defined as follows: given the three-dimensional structure of a protein receptor, its prespecified binding site, and a ligand, predict the pose of the ligand bound to the binding site. The aim of docking methods is to predict the protein–ligand binding mode and estimate the corresponding binding affinity. Usually, docking methods are based on single-objective optimization techniques. A docking method reaches the native-like protein–ligand binding mode by optimizing a well-designed scoring function. Thus, an accurate scoring function and an efficient search strategy constitute the core components of a successful protein–ligand docking method.

The scoring functions are used to evaluate the protein–ligand interactions during a docking simulation. A sophisticated scoring function can achieve great performance regarding accuracy and speed. Numerous scoring functions have been proposed for molecule docking, and

* Corresponding author.
E-mail addresses: leadingsong@outlook.com, leadingsong@cczu.edu.cn (S. Song), tang@ait.kyushu-u.ac.jp (C. Tang), songzhenyu@tzu.edu.cn (Z. Song), qj199232@cczu.edu.cn (J. Qu), xingzai@jsut.edu.cn (X. Chen).

https://doi.org/10.1016/j.asoc.2024.111732
Received 3 December 2023; Received in revised form 13 April 2024; Accepted 2 May 2024
Available online 9 May 2024

these functions can be roughly classified into four groups [6]: force-field-based, empirical, knowledge-based, and machine-learning-based scoring functions. Force-field-based scoring functions are derived from the laws of physics and calculate the noncovalent energy terms within protein–ligand interactions [7]. Empirical scoring functions, e.g., the scoring function of AutoDock Vina [8], accumulate the contributions of energetic factors in protein–ligand binding. Knowledge-based scoring functions sum up pairwise statistical potentials between protein and ligand [9]. Machine-learning-based scoring functions employ machine learning techniques to construct statistical models of scoring functions [10]. Overall, it is well accepted that the performance of all of these groups of scoring functions is not very satisfactory [6]. A specific scoring function has a unique theoretical basis and unique applicable condition.

Evolutionary computation (EC) [11] is a family of population-based metaheuristic optimization algorithms that are inspired by natural evolution and mimic biological phenomena. The classic EC algorithms include the genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE). In addition, in recent years, considerable achievements have been witnessed in two typical EC algorithms, i.e., ABC and estimation of distribution algorithm (EDA). ABC is a swarm-based stochastic optimization algorithm and is mainly inspired by the foraging behavior of bee colonies. Since ABC was originally proposed by Karaboga in 2005 [12], unremitting efforts have been made to improve the performance of ABC, and numerous ABC variants have been proposed in the literature [13–16]. Owing to the powerful performance and ease of ABC, ABC and its variants have been applied to solve many real-world problems, such as medical-image processing [17] and robot path planning [18]. On the other hand, EDA [19] is a special class of EC algorithms. In contrast to conventional EC algorithms, EDA explores the search space by sampling explicit probabilistic models, which are built from the promising candidate solutions in the current population. In recent years, significant improvements [20] have been made in EDA research, and the effectiveness of EDA is verified on many complex problems [21,22]. In general, ABC and EDA have emerged as important options among EC algorithms in the face of complex problems.

Due to the importance of the protein–ligand docking problem in structure-based drug design, several classic docking approaches have been developed in the literature, such as Glide [23], GOLD [24], DOCK [25], and AutoDock [26]. Among these docking approaches, AutoDock Vina (Vina for short) [8,27] is arguably one of the most widely used docking approaches owing to its excellent performance and open source. Since the search space of the protein–ligand docking problem is very large, an exhaustive search strategy is impractical. There are two main categories of optimization techniques to execute conformational search in docking methods: the Monte Carlo method and EC. Vina employs a modified Monte Carlo simulation method coupled with a local search. Zhang et al. proposed a blind protein–ligand docking method called EDock [28], where a replica-exchange Monte Carlo simulation is used to perform rigid-body docking. On the other hand, considerable success in applying the EC technique to bioinformatics [29,30] has been achieved. Many protein–ligand docking methods have also employed the EC technique as the search strategy. GOLD and AutoDock use the canonical GA to constitute the core of their search strategies. Leonhart et al. proposed a BRKGA-DOCK method based on a biased random key GA for the protein–ligand docking problem [31]. Prentis et al. improved DOCK by employing a new 3D GA as the search strategy [32]. Ng et al. attempted to adopt PSO to enhance the performance of Vina, and a modified method called PSOVina was proposed [33]. PSOVina combines PSO with BFGS local search to perform conformational search. Later, they improved PSOVina by incorporating chaos-embedded local search into the search strategy [34]. In addition, a random drift PSO was proven effective for protein–ligand docking [35,36]. Song et al. used an adaptive DE algorithm to improve the search efficiency of a docking method [37]. Ji et al. adopted a gradient boosting DE as the search strategy [38]. Moreover, several docking methods based on ABCs have also been proposed in the literature. Uehara et al. employed a special ABC, i.e., a fitness learning-based ABC with proximity stimuli, to perform protein–ligand docking [39]. Guan et al. attempted to integrate ABC and DE as a hybrid algorithm for protein–ligand docking [40].

These aforementioned works have shown the advantage of the employed EC technique when solving the protein–ligand docking problem. However, the researchers in these works have attempted to improve the EC-based search strategy from the perspective of optimization. Although numerous search mechanisms to enhance docking performance have been proposed, little attention has been paid to the characteristics of the protein–ligand docking problem. As suggested in the work [41], taking full advantage of problem-specific information and incorporating a proprietary search mechanism into EC to improve docking performance is considered a promising research direction. This motivates us to develop a problem-specific EC technique for solving the protein–ligand docking problem.

In this study, considering the superior performance of ABCs, we propose an approach called ABC-EDM to solve the protein–ligand docking problem. The scoring function of Vina is incorporated into ABC-EDM to evaluate a protein–ligand complex during docking simulation. ABC-EDM adopts the general framework of the canonical ABC to execute conformational search. By further investigating the characteristics of the protein–ligand docking problem, a proprietary search mechanism inspired by EDA is proposed to enhance the performance of ABC-EDM. Finally, a set of fifty benchmark docking instances and the GOLD dataset are used to evaluate ABC-EDM. The experimental results show the superiority of ABC-EDM in comparison with the other seven methods. The contribution of this paper is fourfold. First, we propose an improved ABC algorithm called ABC-EDM for protein–ligand docking. Second, we further investigate the characteristics of the protein–ligand docking problem and propose a proprietary search mechanism inspired by EDA to enhance the performance of ABC-EDM. Third, we conduct integral experiments to verify the performance of ABC-EDM. Fourth, a new perspective for solving the protein–ligand docking problem by means of designing a problem-specific search strategy is provided in this paper.

The remainder of this paper is organized as follows. Section 2 presents the description of three important concepts used in this study. Section 3 presents the details of the proposed ABC-EDM approach. The experimental studies are provided in Section 4. Finally, Section 5 draws the conclusion of this paper.

2. Materials

In this section, we introduce three important concepts used in this study: the protein–ligand docking problem, ABC, and EDA.

2.1. The formulation of the protein–ligand docking problem

Given a protein receptor and its binding site, the goal of protein–ligand docking methods is to predict the pose of the ligand at the binding site and to give a score estimating the binding affinity. Normally, the protein receptor is regarded as a rigid object, while the ligand has flexibility and is seen as an articulated object. Given a ligand with n active rotatable bonds, a solution is represented as a real-value vector with $n + 7$ variables in common docking methods. This vector provides the geometric description of the ligand bound to the binding site, including the conformation, orientation, and position of the ligand.

Fig. 1 exhibits the strategy of solution encoding in the proposed approach. The first n variables of the vector $\mathbf{s}$ represent the torsion angles $(\tau_1, \tau_2, \ldots, \tau_n)$, which describe the flexibility of the ligand and uniquely determine the conformation of the ligand. The middle four variables (v_x, v_y, v_z, θ) of the vector $\mathbf{s}$ form a unit vector $\mathbf{u} = (u_x, u_y, u_z)$ and a rotation angle θ that determine the orientation of the ligand.

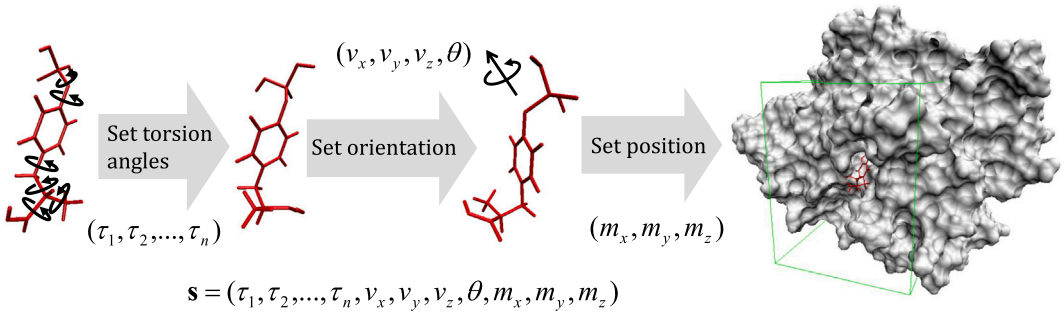


Fig. 1. The strategy of solution encoding in the proposed approach. The designated binding site is marked as the green cube.

Given a rotation around the unit vector $\mathbf{u}$ through the angle θ on any point $\mathbf{p} = (p_x, p_y, p_z)$ in the ligand, the new point $\mathbf{p}' = (p'_x, p'_y, p'_z)$ is calculated using the Hamilton product:

$$\begin{aligned} \mathbf{p}' &= \mathbf{q}\mathbf{p}\mathbf{q}^{-1} \\ \mathbf{q} &= \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2} \\ \mathbf{q}^{-1} &= \cos\frac{\theta}{2} - (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2} \end{aligned} \tag{1}$$

where $\mathbf{q}$ and $\mathbf{q}^{-1}$ are two conjugate unit quaternions formed by $\mathbf{u}$ and θ. $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the unit quaternions along three spatial axes. The last three variables of the vector $\mathbf{s}$ compose a translation vector $\mathbf{m} = (m_x, m_y, m_z)$ to move the ligand into the binding site. In addition, all of the torsion angles $(\tau_1, \tau_2, \ldots, \tau_n)$ and the rotation angle θ range in $[-\pi, \pi]$. The variables (v_x, v_y, v_z) representing the orientation of the ligand are limited to $[-1, 1]$. Then, they are normalized as the unit vector $\mathbf{u}$. The variables (m_x, m_y, m_z) dominating the position of the ligand are restricted to suitable scopes, ensuring that the ligand is within the three-dimensional search space of the binding site.

2.2. Canonical ABC

The artificial bee colony algorithm is a powerful stochastic optimization algorithm and is mainly inspired by the foraging behavior of bee colonies [12]. In the ABC algorithm, a candidate solution of an optimization problem is represented as the position of a food source. The bee colony in ABC is composed of three types of bees: employed bee, onlooker bee and scout bee. ABC solves the optimization problem by driving the bee colony to continually seek a better food source within the search space.

A candidate solution (the position of a food source) in ABC is represented as a vector $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_D^{(i)})$, where D is the number of dimensions of the optimization problem. $\mathbf{x}^{(i)}$ represents the ith individual in the SN-size food sources. The minimum and maximum bounds of these candidate solutions are set to $\mathbf{x}^{min} = (x_1^{min}, x_2^{min}, \ldots, x_D^{min})$ and $\mathbf{x}^{max} = (x_1^{max}, x_2^{max}, \ldots, x_D^{max})$, respectively. The framework of the canonical ABC consists of four main phases: the initialization phase, employed bee phase, onlooker bee phase, and scout bee phase. The details of the four phases are described as follows.

Initialization phase: The positions of the food sources in ABC are uniformly initialized in the problem-specific search space as:

$$x_j^{(i)} = x_j^{min} + r_1(x_j^{max} - x_j^{min}) \quad , \; j \in \{1, 2, \ldots, D\} \tag{2}$$

where $x_j^{(i)}$ is the jth component of $\mathbf{x}^{(i)}$. r_1 is a random number that is uniformly distributed in $[0, 1]$. For a minimization problem, the fitness value of each individual $\mathbf{x}^{(i)}$ is calculated as follows:

$$fit(\mathbf{x}^{(i)}) = \begin{cases} 1/(1 + f(\mathbf{x}^{(i)})) & \text{if } f(\mathbf{x}^{(i)}) \geq 0 \\ 1 + |f(\mathbf{x}^{(i)})| & \text{otherwise} \end{cases} \tag{3}$$

where $fit(\mathbf{x}^{(i)})$ and $f(\mathbf{x}^{(i)})$ represent the fitness value and objective function value of $\mathbf{x}^{(i)}$, respectively. Then, ABC starts its main loop and executes the following three phases iteratively until the stopping criterion is met.

Employed bee phase: There are SN employed bees in ABC, and each employed bee is associated with a food source. The ith employed bee searches the surrounding area of its associated food source. The position of the candidate food source $\mathbf{v}^{(i)} = (v_1^{(i)}, v_2^{(i)}, \ldots, v_D^{(i)})$ inherits from $\mathbf{x}^{(i)}$ except the jth component. This component is calculated as follows:

$$v_j^{(i)} = x_j^{(i)} + r_2(x_j^{(i)} - x_j^{(k)}) \tag{4}$$

where j is randomly selected from $\{1, 2, \ldots, D\}$ and $\mathbf{x}^{(k)}$ donates another food source in the population. r_2 is a random number and ranges in $[-1, 1]$. The ith employed bee will compare the fitness values of $\mathbf{x}^{(i)}$ and $\mathbf{v}^{(i)}$. If $\mathbf{v}^{(i)}$ is better than $\mathbf{x}^{(i)}$, $\mathbf{v}^{(i)}$ replaces $\mathbf{x}^{(i)}$ and is memorized as the new food source; otherwise, $\mathbf{x}^{(i)}$ is retained in the next generation. Then, the employed bees go back to share the information about the food sources with the onlooker bees.

Onlooker bee phase: There are also SN onlooker bees in ABC. Every onlooker bee will first determine a food source in the population and then search its surrounding area. Based on the information brought by the employed bees, an onlooker bee tends to select a food source with a larger fitness value. The selection probability of a food source is calculated as follows:

$$p_i = fit(\mathbf{x}^{(i)}) / \sum_{i=1}^{SN} fit(\mathbf{x}^{(i)}) \tag{5}$$

After selecting a food source, the onlooker bee adapts the same method used by the employed bees to seek a better food source.

Scout bee phase: Since the nectar amount of a food source is limited, a food source will be abandoned if it cannot be improved during predetermined trials (denoted as the parameter '*limit*'). The associated employed bee becomes a scout bee and relocates to a new food source within the search space by using Eq. (2).

2.3. Estimation of distribution algorithm

EDA solves a problem by evolving a set of candidate solutions through a cycle of computational steps. Compared with traditional EC algorithms, EDA generates offspring by sampling an explicit probabilistic model that is built from promising candidate solutions. The flowchart of a typical EDA is exhibited in Algorithm 1.

3. Methodology

This section presents the details of the proposed ABC-EDM approach, including the scoring function, EDM, and the overview of ABC-EDM.

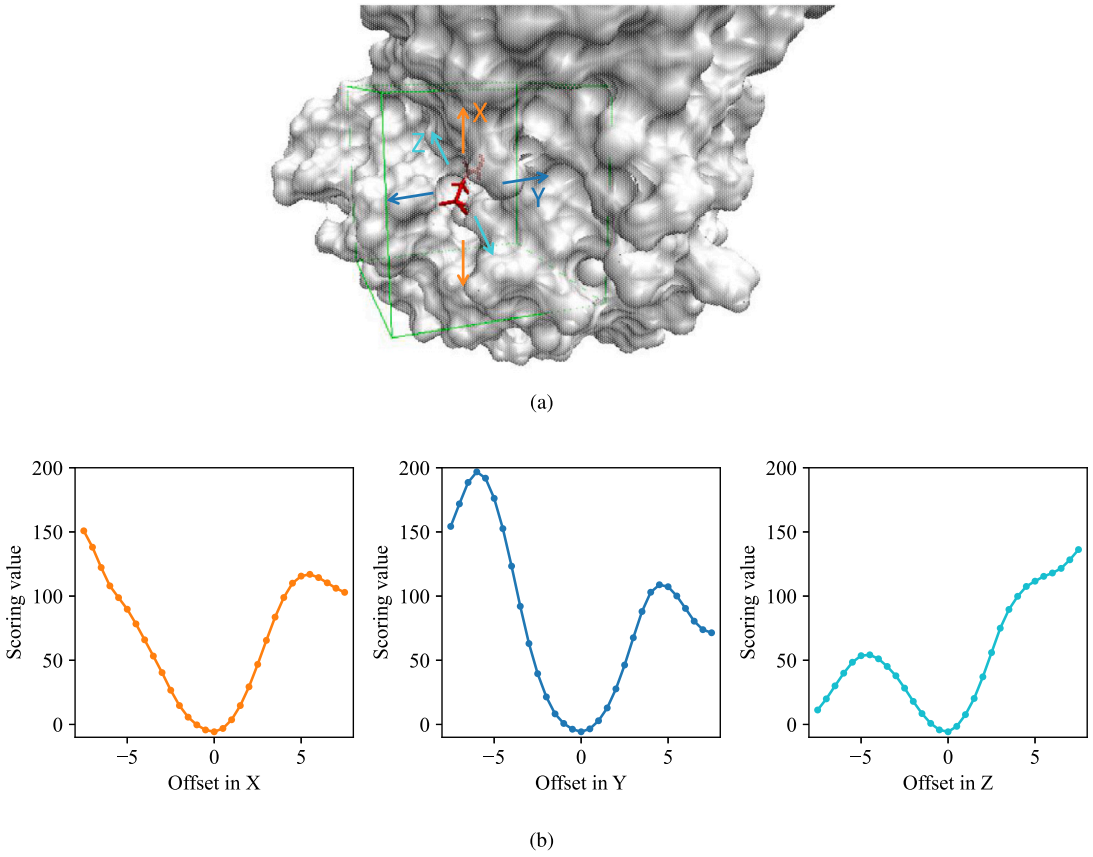


Fig. 2. Figure (a) shows how to offset the native ligand uniformly along the x-axis, y-axis, and z-axis. Correspondingly, the variations of the scoring function are exhibited in Figure (b).

Algorithm 1: The pseudocode of a typical EDA.

Input: Objective function, algorithm parameters.
Output: The best solution visited by the optimization algorithm.
begin
  Initialize the N-size population $\mathcal{P} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$.
  /* A solution is encoded as $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_D^{(i)})$. */
  Calculate the fitness value of each individual by using the objective function.
  while *Stopping criterion is not met* **do**
    Select M $(0 < M < N)$ promising solutions from $\mathcal{P}$.
    Build probabilistic model $p(x_1, x_2, ..., x_D)$ according to the M promising solutions.
    Sample $p(x_1, x_2, ..., x_D)$ to generate a set of new candidate solutions $\mathcal{P}'$.
    Calculate the fitness value of each individual in $\mathcal{P}'$ by using the objective function.
    Merge $\mathcal{P}$ and $\mathcal{P}'$, then create new $\mathcal{P}$ based on survival of the fittest.
  Output result.

3.1. Scoring function

Protein–ligand docking methods are usually based on single-objective optimization techniques, where the conformational search is performed under the guidance of the single-objective scoring function. The Vina function is a classic scoring function, and its effectiveness has been widely verified in the literature [33,35,37,42]. In this study, the Vina function is employed to evaluate the conformation of a protein–ligand complex during docking simulation. A lower value of the Vina function corresponds to a better conformation when two protein–ligand complexes are compared.

The conformation-dependent part of the Vina function is taken into consideration in this study, and the conformation-independent part, i.e., the penalty term about ligand flexibility, is omitted. The conformation-dependent part is a sum consisting of intermolecular and intramolecular contributions. The total scoring function c can be calculated as follows:

$$c = \sum_{i<j} c_{ij} \tag{6}$$

where c_{ij} is the scoring value of the pair of atoms i and j in the protein–ligand complex. c_{ij} can be calculated by the following equation:

$$\begin{aligned} c_{ij} = {} & w_1 Gauss_1(d_{ij}) + w_2 Gauss_2(d_{ij}) + w_3 Repulsion(d_{ij}) \\ & + w_4 Hydrophobic(d_{ij}) + w_5 HBonding(d_{ij}) \end{aligned} \tag{7}$$

where d_{ij} is the surface distance between the atoms i and j. $w_1 \sim w_5$ are five weighting parameters of different items that are optimized based on the PDBbind dataset [43]. The first three items in Eq. (7) describe the steric interaction between atom i and atom j. The fourth item reflects the hydrophobic interaction, and the fifth item reflects the effect of hydrogen bonding. These two items are calculated only when

the two types of interaction are identified. In detail, the five items in Eq. (7) can be calculated as follows:

$$
\begin{aligned}
Gauss_1(d_{ij}) &= e^{-(d_{ij}/0.5)^2} \\
Gauss_2(d_{ij}) &= e^{-((d_{ij}-3)/2)^2} \\
Repulsion(d_{ij}) &= \begin{cases} d_{ij}^2 & \text{if } d_{ij} < 0 \\ 0 & \text{if } d_{ij} \geq 0 \end{cases} \\
Hydrophobic(d_{ij}) &= \begin{cases} 1 & \text{if } d_{ij} \leq 0.5 \\ 1.5 - d_{ij} & \text{if } 0.5 < d_{ij} < 1.5 \\ 0 & \text{if } d_{ij} \geq 1.5 \end{cases} \\
HBonding(d_{ij}) &= \begin{cases} 1 & \text{if } d_{ij} \leq -0.7 \\ d_{ij}/(-0.7) & \text{if } -0.7 < d_{ij} < 0 \\ 0 & \text{if } d_{ij} \geq 0 \end{cases}
\end{aligned} \tag{8}
$$

3.2. Estimation of distribution mechanism

As mentioned above, the scoring function used in this study consists of intermolecular and intramolecular contributions. The intermolecular contributions score the interactions of the heavy atom pairs between the receptor and the ligand, and the intramolecular contributions score the non-1-4 heavy atom pairs in the ligand structure. From Eq. (8), it is easy to conclude that the Euclidean distances of the heavy atom pairs between the receptor and the ligand are essential for computing the intermolecular contributions. Compared with the conformation and orientation of a ligand, the position of the ligand can have a macro effect on the calculation of the scoring function. Taking advantage of this problem-specific information and then designing proprietary search mechanisms are considered to be helpful to improving the performance of docking methods.

In the scout bee phase of the canonical ABC, when a food source is abandoned, the associated employed bee will become a scout bee and relocate to a new food source within the search space. However, this scout bee hardly uses information about the current status of the food sources and randomly locates a solution in the search space. In this study, inspired by EDA, a mechanism called EDM is designed to assign the scout bee to a promising food source (candidate solution).

Differing from a common EDA that estimates the probability distribution on the whole variables, the proposed EDM estimates the probability distribution only on the variables m_x, m_y, and m_z that are related to the position of the ligand. This is because these three variables are considered to play a more important role in the docking process, as discussed above, and are delimited to the other variables in solution coding. Moreover, the probabilistic model is a key component of an EDA. Since the variables m_x, m_y, and m_z are the three components of a translation vector, it is obvious that they are independent from each other. Thus, the type of univariate probabilistic models [19] can be used in the proposed EDM. We empirically investigate the relation between the scoring function and the variables m_x, m_y, and m_z. Fig. 2(a) exhibits a typical protein–ligand complex where a ligand is bound to the binding site in the native state. We offset the ligand uniformly along the x-axis (i.e., modify the variable m_x), y-axis, and z-axis. The variations of the scoring function are exhibited in Fig. 2(b). From Fig. 2(b), we can see that the function graphs of the scoring function have funnel-shaped landscapes. This implies that the normal distribution is suitable for building probabilistic models for the variables m_x, m_y, and m_z. This is because the samples of a variable obeying a normal distribution can center around a single point. On the other hand, it is not easy to determine probabilistic models for the rest variables $(\tau_1, \tau_2, \ldots, \tau_n, v_x, v_y, v_z, \theta)$. Intuitively, the normal distribution is not suitable for building probabilistic models for these variables. It is considered arbitrary to casually select probabilistic models for these variables. Since these variables are related to the conformation and orientation of the ligand, a promising solution is considered to have suitable conformation and orientation. As a result, it is a wise method to set these variables of a candidate solution by directly inheriting from a promising solution.

In the scout bee phase of the proposed ABC-EDM approach, EDM works in three steps to relocate a new food source. First, M promising solutions that have better fitness values are selected from the current population of food sources. Second, for each variable of m_x, m_y, and m_z, the means $(\mu_{m_x}, \mu_{m_y}, \mu_{m_z})$ and the standard deviations $(\delta_{m_x}, \delta_{m_y}, \delta_{m_z})$ are calculated based on these M promising solutions. Consequently, three normal distribution models, i.e., $N(\mu_{m_x}, \delta_{m_x}^2)$, $N(\mu_{m_y}, \delta_{m_y}^2)$, and $N(\mu_{m_z}, \delta_{m_z}^2)$, are built. Finally, a new food source is initialized as follows:

$$
x_j^{(i)} = \begin{cases} x_j^{(best)} & \text{if } j \in \{1, 2, \ldots, n+4\} \\ \text{sampled from } N(\mu_{m_x}, \delta_{m_x}^2) & \text{if } j = n+5 \\ \text{sampled from } N(\mu_{m_y}, \delta_{m_y}^2) & \text{if } j = n+6 \\ \text{sampled from } N(\mu_{m_z}, \delta_{m_z}^2) & \text{if } j = n+7 \end{cases} \tag{9}
$$

where n is the number of active rotatable bonds. $x_j^{(best)}$ is the jth component of the best individual in the current population. In this way, a new candidate solution with a promising conformation and orientation can be relocated to a promising region in the binding site.

3.3. Overview of the proposed ABC-EDM approach

To implement the proposed ABC-EDM approach to solve the protein–ligand docking problem, the scoring function is incorporated as the objective function of ABC-EDM. A solution $\mathbf{s}$ for the protein–ligand docking problem is encoded as the individual $\mathbf{x}^{(i)}$ in ABC-EDM. Consequently, the number of dimensions of $\mathbf{x}^{(i)}$, i.e., D, is equal to $n+7$. In addition, the minimum and maximum bounds of each variable, i.e., $\mathbf{x}^{min}$ and $\mathbf{x}^{max}$, are set to appropriate values, as described in Section 2.1.

Dealing with an infeasible solution in a docking simulation is an important issue in the implementation of ABC-EDM. The unsuitable values of the variables in a solution can cause the ligand to exceed the boundary of the binding site and lead to steric clashing between atoms. To avoid this issue, two simple strategies are proposed in ABC-EDM. In the initialization phase and the scout bee phase of ABC-EDM, the initialization of a new solution is repeated until this solution is checked as a feasible solution. In the employed bee phase and the onlooker bee phase of ABC-EDM, a candidate solution $\mathbf{v}^{(i)}$ is generated based on $\mathbf{x}^{(i)}$, as described in Section 2.2. Since the solution $\mathbf{v}^{(i)}$ may represent an infeasible solution, $\mathbf{v}^{(i)}$ replaces $\mathbf{x}^{(i)}$ and is memorized as the new food source only when $\mathbf{v}^{(i)}$ is checked as a feasible solution and its fitness value is better. Otherwise, $\mathbf{x}^{(i)}$ is retained to the next generation.

The pseudocode of the proposed ABC-EDM is presented in Algorithm 2. ABC-EDM follows the general framework of the canonical ABC but has a proprietary EDM. Before optimization, the docking environment is initialized according to the ligand and the protein receptor. Then, the ABC-EDM algorithm is employed to execute conformational search. Finally, the best solution visited by the bee colony is outputted as the docking result.

4. Experimental study

In this section, we present the experiments to evaluate the performance of the proposed ABC-EDM approach. We compare ABC-EDM with three variants of the ABC algorithm, three EC algorithms, and AutoDock Vina. Moreover, some algorithm analysis of ABC-EDM is also provided.

4.1. Experimental setup

All algorithms in this study are implemented in C++ and Python language. They are executed on the Linux 64-bit system with an Intel Core i5 CPU and 16 G memory.

Algorithm 2: The pseudocode of the proposed ABC-EDM.

Input: A protein receptor and its binding site, a ligand, algorithm parameters.
Output: The best solution visited by ABC-EDM.
begin
  /* Initialization phase. */
  Initialize the docking environment.
  Obtain the minimum and maximum bounds, i.e., $\mathbf{x}^{min}$ and $\mathbf{x}^{max}$.
  Initialize the population $\{\mathbf{x}^{(i)}|i \in \{1,2,...,SN\}\}$ and ensure that they are feasible.
  Calculate the fitness value of each $\mathbf{x}^{(i)}$, i.e., $fit(\mathbf{x}^{(i)})$.
  while *Stopping criterion is not met* **do**
    /* Employed bee phase. */
    for i *in* $\{1,2,...,SN\}$ **do**
      The ith employed bee searches the neighbor food source $\mathbf{v}^{(i)}$ of $\mathbf{x}^{(i)}$ using Eq. (4).
      Check whether $\mathbf{v}^{(i)}$ is feasible; if so, calculate $fit(\mathbf{v}^{(i)})$.
      if $\mathbf{v}^{(i)}$ *is feasible and* $fit(\mathbf{v}^{(i)}) > fit(\mathbf{x}^{(i)})$ **then**
        $\mathbf{x}^{(i)} \leftarrow \mathbf{v}^{(i)}$
      else
        $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)}$
    /* Onlooker bee phase. */
    Calculate the selection probability of each food source p_i using Eq. (5).
    for k *in* $\{1,2,...,SN\}$ **do**
      The kth onlooker bee selects a food source $\mathbf{x}^{(i)}$ according to the selection probability.
      The kth onlooker bee searches the neighbor food source $\mathbf{v}^{(i)}$ of $\mathbf{x}^{(i)}$ using Eq. (4).
      Check whether $\mathbf{v}^{(i)}$ is feasible; if so, calculate $fit(\mathbf{v}^{(i)})$.
      if $\mathbf{v}^{(i)}$ *is feasible and* $fit(\mathbf{v}^{(i)}) > fit(\mathbf{x}^{(i)})$ **then**
        $\mathbf{x}^{(i)} \leftarrow \mathbf{v}^{(i)}$
      else
        $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)}$
    /* Scout bee phase. */
    for i *in* $\{1,2,...,SN\}$ **do**
      if $\mathbf{x}^{(i)}$ *cannot be improved during predetermined trials* $'limit'$ **then**
        /* Estimation of distribution mechanism. */
        Select M promising solutions from the current population.
        Build normal distribution $N(\mu_{mx}, \delta_{mx}^2)$, $N(\mu_{my}, \delta_{my}^2)$, and $N(\mu_{mz}, \delta_{mz}^2)$.
        Repeat locating a new food source $\mathbf{x}^{(i)}$ by Eq. (9) until it is feasible.
        Calculate $fit(\mathbf{x}^{(i)})$.
  Output result.

A set of 50 protein–ligand complexes with a wide range of ligand sizes is used as the test docking instances, as shown in Table 1. The number of active rotatable bonds in these ligands varies from 1 to 21. The corresponding native structure of each complex was downloaded from the PDB database [44] and was properly prepared before docking simulation. In detail, for the structure of a complex, all water molecules are removed from it, and all missing hydrogens are appended into it. Then, the processed complex is separated into a ligand structure and a receptor structure. Next, AutoDockTools [26,27] is used to assign partial charges to the corresponding atoms in the ligand structure and the receptor structure. Finally, the information about the binding site is configured as a supplementary file.

Table 1
Fifty protein–ligand complexes are selected as the test docking instances.

PDB ID	Size[a]	PDB ID	Size	PDB ID	Size	PDB ID	Size	PDB ID	Size
1AHA	1	1HSL	4	1LAH	6	6RNT	7	1EAP	12
1MRG	1	1SRJ	4	1ROB	6	2CGR	8	1GLP	12
3HVT	1	2MCP	4	2ADA	6	1ETR	9	1SLT	12
1TNG	2	3GCH	4	2CMD	6	2SIM	9	4DFR	14
2PHH	2	7TIM	4	2GBP	6	3TPI	9	1TMN	15
3PTB	2	1COM	5	2PK4	6	1ATL	10	1IDA	16
1ACK	3	1DR1	5	1ETA	7	1FKG	10	1LIC	16
1MDR	3	1HYT	5	1LST	7	1LNA	10	2PLV	19
2CHT	3	1TPH	5	2AK3	7	1NCO	11	1AAQ	20
4CTS	3	1HDC	6	3CPA	7	1BMA	12	5P2P	21

[a] Size: Number of active rotatable bonds in the ligand.

To measure the performance of different docking methods, three metrics are employed to evaluate the docking result.

(1) Scoring value: This metric evaluates the optimization performance of a docking method. In addition, it estimates the binding affinity of a docked protein–ligand complex. A lower scoring value corresponds to a better docking result.

(2) Root mean square deviation (RMSD): RMSD measures the geometric similarity between the predicted ligand pose and the native ligand pose. Given a ligand with n heavy atoms, RMSD is used to calculate the average distance between the matched heavy atoms of two poses:

$$\mathrm{RMSD}_{(a,b)} = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \tag{10}$$

where a, b are the two poses of a ligand. d_i is the distance between two corresponding heavy atoms. A smaller RMSD value indicates higher similarity between two poses of the ligand.

(3) Success rate: The docking process is considered successful when the RMSD value of the predicted ligand pose is less than a threshold (e.g. 2 Å). The success rate is defined as the percentage of the successful docking instances.

To verify the performance of the proposed ABC-EDM approach, we compare ABC-EDM with three types of methods, including three ABC variants, three common EC algorithms, and a classic docking program. Usually, a docking method is executed multiple times for an instance, e.g., 30 times, and the ligand pose with the lowest scoring value is outputted as the final docking result. In the experiment, we follow this practice and compare the docking result with the lowest scoring value. To make a fair comparison, the common parameters of these algorithms are set as follows. The population size (or colony size) is set to 100, the maximum number of fitness function evaluations is set to 100000, and the number of docking simulations for a docking instance is set to 30.

4.2. Comparing ABC-EDM with ABC variants

To evaluate the performance of the proposed ABC-EDM approach, we compare it with three classic ABC variants, including the canonical ABC [12], GABC [13], and qABC [14]. The user-defined parameters of these ABC algorithms are set as listed in Table 2. We execute the four ABC algorithms on the 50 test docking instances. As mentioned above, for each instance, the ligand pose with the lowest scoring value during 30 docking simulations is outputted as the final docking result.

Table 3 reports the optimization results of ABC-EDM versus the three ABC variants on the 50 docking instances. From Table 3, we can observe that ABC-EDM demonstrates the best performance on 40 docking instances. ABC-EDM is considered to have better optimization performance than the other three algorithms because ABC, GABC, and qABC yield the best performance on 1, 6, and 3 docking instances, respectively. Moreover, to determine the significant differences between ABC-EDM and the three algorithms, we perform the Friedman test as

Table 2
The user-defined parameters of the seven algorithms.

Algorithm	Parameter settings
ABC	Colony size $CS = 100$; $SN = CS/2$; $limit = 100$
ABC-EDM	$CS = 100$; $SN = CS/2$; $limit = 100$; $M = \lfloor SN/2 \rfloor$
GABC	$CS = 100$; $SN = CS/2$; $limit = 100$; Gbest parameter $C = 1.5$
qABC	$CS = 100$; $SN = CS/2$; $limit = 100$; Neighborhood radius $r = 1$
PSO	$NP = 100$; Inertia weight $w = 0.36$; Cognitive weight $c_1 = 0.99$; Social weight $c_2 = 0.99$
DE	$NP = 100$; Crossover rate $Cr = 0.9$; Scaling factor $F = 0.5$
GA	$NP = 100$; Crossover rate $p_c = 0.9$; Mutation rate $p_m = 0.1$; Number of elites $N_e = 10$

Table 3
The optimization results of ABC-EDM versus the three ABC variants on the 50 docking instances. The scoring value of the ligand pose with the lowest scoring value (among 30 independent docking simulations) is reported for each instance.

PDB ID	ABC-EDM	ABC	GABC	qABC
1AAQ	**−17.13**	−14.01	−15.41	−14.89
1ACK	**−7.70**	−7.62	−7.64	−7.64
1AHA	−6.75	−6.77	**−6.83**	−6.80
1ATL	−10.40	−10.33	−10.32	**−10.68**
1BMA	**−15.11**	−13.78	−14.35	−13.43
1COM	**−8.06**	−7.87	−7.91	−7.90
1DR1	**−9.62**	−9.41	−9.50	−9.44
1EAP	**−9.29**	−8.55	−8.85	−9.13
1ETA	**−6.68**	−6.25	−6.04	−6.17
1ETR	**−17.58**	−14.29	−16.38	−16.35
1FKG	−13.56	−12.98	**−13.86**	−12.75
1GLP	−9.96	**−10.10**	−9.85	−9.71
1HDC	**−13.44**	−13.29	−13.39	−13.10
1HSL	**−8.11**	−7.93	−7.94	−7.88
1HYT	**−8.72**	−8.09	−8.54	−8.47
1IDA	**−21.23**	−19.20	−19.77	−18.96
1LAH	**−8.35**	−8.28	−8.30	−8.25
1LIC	**−11.85**	−10.57	−10.67	−10.93
1LNA	−8.97	−8.85	**−9.15**	−9.00
1LST	**−8.87**	−8.73	−8.66	−8.75
1MDR	**−8.39**	−8.29	−8.35	−8.29
1MRG	−5.97	−5.87	**−5.98**	−5.98
1NCO	**−19.57**	−18.73	−18.65	−17.90
1ROB	**−9.37**	−9.26	−9.23	−9.15
1SLT	**−9.59**	−9.32	−9.46	−9.37
1SRJ	**−10.92**	−10.31	−10.68	−10.61
1TMN	−13.44	−13.60	**−14.94**	−13.42
1TNG	**−5.23**	−5.03	−5.09	−5.07
1TPH	−6.91	−7.04	**−7.35**	−7.10
2ADA	**−11.96**	−11.80	−11.85	−11.83
2AK3	**−11.12**	−10.89	−10.67	−10.50
2CGR	**−14.85**	−13.66	−14.80	−14.50
2CHT	−5.73	−6.13	−6.28	**−6.38**
2CMD	**−9.83**	−9.72	−9.76	−9.74
2GBP	**−9.35**	−9.22	−9.25	−9.27
2MCP	**−5.02**	−4.73	−5.02	−4.76
2PHH	**−7.47**	−7.44	−7.47	−7.46
2PK4	**−5.60**	−5.28	−5.38	−5.55
2PLV	**−11.29**	−10.49	−10.58	−10.38
2SIM	**−11.80**	−11.57	−11.73	−11.54
3CPA	**−10.85**	−10.38	−10.51	−10.33
3GCH	**−7.38**	−7.04	−6.93	−6.98
3HVT	**−10.77**	−10.61	−10.69	−10.66
3PTB	**−6.65**	−6.62	−6.64	−6.58
3TPI	**−11.42**	−11.20	−11.25	−11.34
4CTS	**−5.97**	−5.72	−5.89	−5.85
4DFR	**−13.67**	−13.11	−13.29	−12.39
5P2P	**−13.84**	−13.47	−13.62	−13.49
6RNT	**−8.52**	−8.22	−8.11	−8.10
7TIM	−6.66	−6.56	−6.55	**−6.68**
Number of best	40	1	6	3

shown in Table 4. The Friedman test ranks the four ABC variants, and a smaller ranking value indicates a better performance. We can see that ABC-EDM achieves the smallest ranking of 1.38. To avoid a Type I error [45], the Holm's method is employed to adjust the p values to p_{Holm} values. The p_{Holm} values of ABC, GABC, and qABC are smaller

Table 4
Statistical results obtained by the Friedman test on the optimization results (in Table 3).

Algorithm	Ranking	Unadjusted p value	p_{Holm} value
ABC-EDM	1.38	–	–
GABC	2.30	0.000366	0.000366
qABC	3.00	0	0
ABC	3.32	0	0

than the significance level of 0.05. This indicates that ABC-EDM is significantly better than the three ABC variants in terms of optimization performance.

We compare the convergence performance of ABC-EDM and ABC. The average (30 times) convergence curves of ABC-EDM and ABC for six typical instances are illustrated in Fig. 3. These convergence curves reflect the exploration and exploitation ability of ABC-EDM and ABC [46]. Obviously, ABC converges close to ABC-EDM in the early search process. This finding indicates that ABC and ABC-EDM have similar exploration abilities. Moreover, in the late search process, ABC-EDM can more effectively improve the quality of the visited best solution than ABC. This observation implies that ABC-EDM have a stronger exploration ability. Since ABC-EDM is mainly different from ABC in that it has a proprietary EDM, the stronger exploration ability of ABC-EDM is considered to benefit from this mechanism. Overall, the effectiveness of the proposed EDM is empirically demonstrated.

We also compare the docking accuracy of the four ABC algorithms for the 50 docking instances. The RMSD values of the predicted ligand poses (reported in Table 3) are summarized in Table 5. From Table 5, we can see that ABC-EDM achieves the best RMSD values on 19 docking instances among the four algorithms. ABC, GABC, and qABC achieve the best RMSD values on 7, 11, and 13 docking instances, respectively. It is obvious that ABC-EDM achieves the best docking performance in terms of RMSD values. Moreover, the success rates of the four ABC algorithms are also reported in Table 5. ABC-EDM achieves the best performance among the four algorithms because ABC-EDM obtains the highest success rate. Overall, we can conclude that the proposed ABC-EDM can provide a competitive result compared with the three ABC variants in terms of docking accuracy.

4.3. Comparing ABC-EDM with EC algorithms

To further evaluate the performance of the proposed ABC-EDM, three classical EC algorithms are used as the test baseline to show the performance of ABC-EDM. The three EC algorithms are PSO, DE, and GA. To make a fair comparison, their common parameters are set as mentioned above. In addition, they share the same initialization method, and their user-defined parameters are properly set based on the works in the literature [33,47], as listed in Table 2. We perform the four algorithms 30 times on the 50 test docking instances. For each instance, the ligand pose with the lowest scoring value is outputted as the final docking result.

The optimization results of the four algorithms on the 50 docking instances are summarized in Table 6. From Table 6, we can observe that ABC-EDM and DE achieve the best result on 25 and 24 docking

Table 5
The docking results of ABC-EDM versus the three ABC variants on the 50 docking instances. The RMSD (Å) of the predicted ligand pose is reported for each instance.

PDB ID	ABC-EDM	ABC	GABC	qABC
1AAQ	**3.65**	6.05	5.53	5.69
1ACK	**0.67**	0.71	0.75	0.79
1AHA	0.21	0.28	0.30	**0.20**
1ATL	2.71	4.30	2.70	**2.68**
1BMA	**1.30**	3.48	3.71	6.14
1COM	3.10	3.14	**2.34**	4.65
1DR1	1.43	0.49	1.44	**0.47**
1EAP	**2.61**	4.64	4.21	3.04
1ETA	**8.39**	10.39	10.44	10.52
1ETR	**0.82**	1.40	1.42	1.27
1FKG	7.18	**1.21**	1.38	7.27
1GLP	**2.39**	11.43	2.43	6.88
1HDC	**0.93**	1.09	1.06	1.01
1HSL	0.26	**0.25**	1.11	0.94
1HYT	**1.35**	1.86	1.40	1.45
1IDA	1.80	2.12	**1.61**	4.08
1LAH	**0.24**	1.03	1.02	0.28
1LIC	6.40	2.45	**2.22**	5.89
1LNA	1.68	1.48	2.30	**1.40**
1LST	**0.17**	0.23	0.98	0.22
1MDR	1.03	1.03	**1.03**	1.04
1MRG	0.38	**0.36**	0.38	0.49
1NCO	**0.27**	0.56	1.09	1.24
1ROB	**0.45**	0.65	0.51	0.68
1SLT	4.83	**2.05**	4.73	8.15
1SRJ	1.27	1.43	1.28	**1.23**
1TMN	7.00	**1.44**	1.59	6.87
1TNG	1.82	1.85	**1.80**	1.81
1TPH	0.79	1.52	1.71	**0.34**
2ADA	0.31	0.29	**0.28**	0.49
2AK3	2.56	2.54	0.99	**0.96**
2CGR	**0.63**	3.06	0.92	1.03
2CHT	**11.03**	11.45	11.38	11.40
2CMD	1.35	1.40	1.38	**1.27**
2GBP	**0.53**	0.58	0.59	0.57
2MCP	4.58	4.66	**4.42**	4.50
2PHH	1.07	1.05	**0.65**	1.03
2PK4	1.45	1.63	**1.27**	1.32
2PLV	10.80	12.65	10.71	**8.28**
2SIM	**0.44**	0.56	0.91	0.57
3CPA	1.68	**1.65**	1.82	1.88
3GCH	**1.99**	2.49	2.72	2.26
3HVT	0.44	0.43	0.31	**0.30**
3PTB	**1.60**	1.61	1.60	1.93
3TPI	0.58	**0.43**	0.88	0.52
4CTS	3.98	3.75	3.99	**3.49**
4DFR	6.28	10.98	11.07	**6.07**
5P2P	5.21	5.46	**3.60**	5.09
6RNT	0.60	0.72	**0.56**	1.25
7TIM	1.97	2.00	2.06	**1.94**
Number of best	19	7	11	13
Success rate (RMSD<2 Å)	66.00%	60.00%	64.00%	62.00%

Table 6
The optimization results of ABC-EDM versus the three EC algorithms on the 50 docking instances. The scoring value of the ligand pose with the lowest scoring value (among 30 independent docking simulations) is reported for each instance.

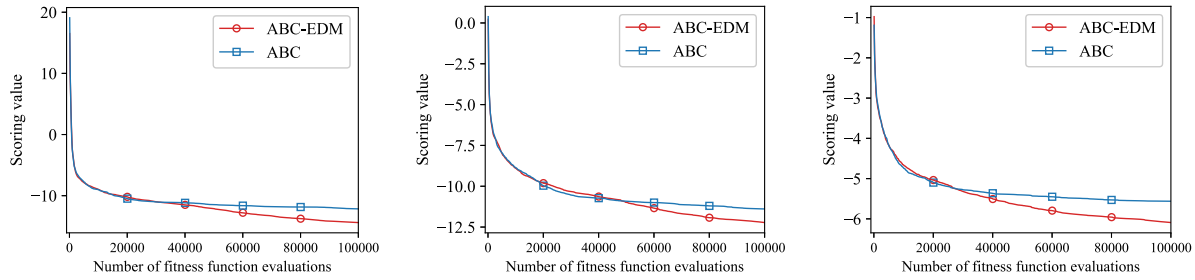
PDB ID	ABC-EDM	PSO	DE	GA
1AAQ	**−17.13**	−11.49	−13.69	−10.50
1ACK	**−7.70**	−7.65	−7.67	−7.55
1AHA	−6.75	−6.83	**−6.84**	−6.82
1ATL	−10.40	−8.87	**−10.70**	−9.43
1BMA	**−15.11**	−12.94	−14.60	−12.52
1COM	**−8.06**	−7.69	−7.95	−7.73
1DR1	**−9.62**	−9.44	−9.59	−9.20
1EAP	−9.29	−9.21	**−10.57**	−8.68
1ETA	**−6.68**	−6.18	−6.28	−6.36
1ETR	**−17.58**	−11.29	−14.08	−12.67
1FKG	−13.56	−10.75	**−15.31**	−10.89
1GLP	−9.96	−8.90	**−12.02**	−9.09
1HDC	−13.44	−12.29	**−13.63**	−12.89
1HSL	**−8.11**	−7.99	−8.07	−7.89
1HYT	**−8.72**	−7.04	−8.20	−6.86
1IDA	**−21.23**	−13.11	−13.89	−12.01
1LAH	−8.35	−8.27	**−8.42**	−8.22
1LIC	**−11.85**	−10.42	−11.48	−10.36
1LNA	**−8.97**	−7.16	−7.37	−7.08
1LST	−8.87	−8.79	**−8.91**	−8.67
1MDR	−8.39	−8.25	**−8.39**	−8.36
1MRG	**−5.97**	−5.95	−5.96	−5.87
1NCO	**−19.57**	−13.98	−14.11	−12.55
1ROB	−9.37	−9.17	**−9.41**	−9.09
1SLT	−9.59	−8.59	**−9.72**	−8.95
1SRJ	−10.92	−9.67	**−11.00**	−8.14
1TMN	−13.44	−12.61	**−15.23**	−12.00
1TNG	**−5.23**	−5.13	−5.09	−5.03
1TPH	**−6.91**	−5.41	−5.79	−5.30
2ADA	−11.96	−11.69	**−11.97**	−11.77
2AK3	**−11.12**	−10.69	−10.97	−10.42
2CGR	−14.85	−10.20	**−15.08**	−10.96
2CHT	−5.73	−5.05	−5.07	**−6.45**
2CMD	**−9.83**	−9.74	−9.82	−9.75
2GBP	−9.35	−9.32	**−9.37**	−9.30
2MCP	−5.02	−4.34	**−5.19**	−4.65
2PHH	−7.47	−7.38	**−7.47**	−7.08
2PK4	−5.60	−5.23	**−5.89**	−4.97
2PLV	**−11.29**	−8.46	−10.57	−8.56
2SIM	−11.80	−10.95	**−11.87**	−11.63
3CPA	**−10.85**	−6.55	−8.06	−6.92
3GCH	−7.38	−6.89	**−7.39**	−6.46
3HVT	−10.77	−10.73	**−10.77**	−10.57
3PTB	**−6.65**	−6.59	−6.64	−6.64
3TPI	**−11.42**	−10.68	−6.40	−10.91
4CTS	−5.97	−5.75	**−6.01**	−5.71
4DFR	**−13.67**	−11.29	−13.04	−11.42
5P2P	**−13.84**	−10.06	−13.12	−11.28
6RNT	−8.52	−8.46	**−8.59**	−7.68
7TIM	**−6.66**	−5.83	−6.00	−6.53
Number of best	25	0	24	1

instances, respectively. ABC-EDM and DE are considered to have better optimization performance than the other two algorithms because PSO and GA obtain the best optimization performance on 0 and 1 docking instances, respectively. The Friedman test is employed to determine the significance, and Table 7 reports the statistical results. ABC-EDM achieves the smallest ranking of 1.52, and ABC-EDM is considered to be significantly better than GA and PSO because the p_{Holm} values are less than 0.05. However, ABC-EDM is not significantly better than DE because the p_{Holm} value is larger than 0.05. Overall, ABC-EDM can provide a better or a competitive result compared with three classical EC algorithms in terms of optimization performance.

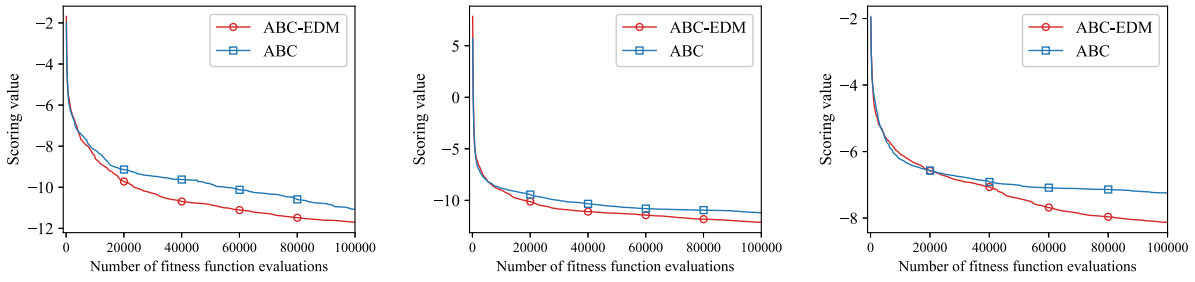
The RMSD values of the predicted ligand poses (reported in Table 6) are summarized in Table 8. For the 50 docking instances, we find that ABC-EDM, PSO, DE, and GA achieve the best result on 25, 12, 6, and 7 docking instances, respectively. This indicates that ABC-EDM can achieve the most accurate docking results on half of the 50 docking instances. The success rates of the four algorithms are reported in Table 8. ABC-EDM achieves the highest success rate of 66%, implying that ABC-EDM has the best performance compared with the other three EC algorithms. Therefore, we can conclude that, compared with three classical EC algorithms, ABC-EDM provides the most powerful performance for the protein–ligand docking problem in terms of docking accuracy.

Table 7
Statistical results obtained by the Friedman test on the optimization results (in Table 6).

Algorithm	Ranking	unadjusted p value	p_{Holm} value
ABC-EDM	1.52	–	–
DE	1.66	0.587669	0.587669
PSO	3.38	0	0
GA	3.44	0	0



(a) 1AAQ (b) 1BMA (c) 1ETA



(d) 2CGR (e) 5P2P (f) 6RNT

Fig. 3. The convergence curves of ABC-EDM and ABC for six typical instances.

Table 8
The docking results of ABC-EDM versus the three EC algorithms on the 50 docking instances. The RMSD (Å) of the predicted ligand pose is reported for each instance.

PDB ID	ABC-EDM	PSO	DE	GA
1AAQ	**3.65**	7.88	11.00	14.16
1ACK	**0.67**	0.77	0.68	0.75
1AHA	**0.21**	0.30	0.26	0.31
1ATL	**2.71**	4.10	4.71	3.72
1BMA	**1.30**	3.20	3.35	5.54
1COM	3.10	**2.37**	2.43	2.91
1DR1	1.43	0.48	1.42	**0.37**
1EAP	**2.61**	3.17	3.33	6.83
1ETA	8.39	9.20	**6.64**	10.00
1ETR	**0.82**	6.31	6.47	6.32
1FKG	7.18	3.95	**1.49**	5.38
1GLP	2.39	5.78	**0.69**	4.83
1HDC	0.93	1.06	**0.77**	0.99
1HSL	0.26	**0.24**	0.28	1.43
1HYT	**1.35**	8.76	8.57	9.02
1IDA	**1.80**	12.89	12.80	13.64
1LAH	0.24	**0.19**	1.02	1.03
1LIC	6.40	**4.87**	7.05	7.30
1LNA	**1.68**	6.07	5.47	8.40
1LST	**0.17**	0.19	0.97	1.00
1MDR	1.03	**0.94**	1.03	0.98
1MRG	**0.38**	0.44	0.39	0.48
1NCO	**0.27**	9.00	7.50	10.20
1ROB	**0.45**	0.91	0.46	0.50
1SLT	4.83	2.08	4.72	**1.86**
1SRJ	**1.27**	2.55	1.29	7.28
1TMN	**7.00**	7.21	9.57	8.11
1TNG	1.82	1.89	**0.51**	0.56
1TPH	**0.79**	6.21	6.28	5.45
2ADA	0.31	0.30	0.31	**0.27**
2AK3	2.56	**0.56**	1.07	1.10
2CGR	**0.63**	3.11	1.07	3.50
2CHT	11.03	**0.34**	1.12	11.46
2CMD	**1.35**	1.61	1.39	1.64
2GBP	**0.53**	0.59	0.55	0.56
2MCP	4.58	**2.30**	4.57	2.42
2PHH	1.07	**0.64**	1.07	1.00
2PK4	1.45	**0.78**	1.28	2.42
2PLV	**10.80**	11.57	11.83	11.94
2SIM	**0.44**	0.71	1.07	0.45
3CPA	**1.68**	8.60	8.91	8.27
3GCH	1.99	2.49	**1.98**	2.01
3HVT	0.44	0.48	0.44	**0.30**
3PTB	1.60	0.43	0.42	**0.39**
3TPI	0.58	0.91	10.01	**0.45**
4CTS	3.98	**0.75**	3.97	1.19
4DFR	**6.28**	6.57	6.75	6.99
5P2P	**5.21**	7.52	10.45	10.29
6RNT	0.60	**0.51**	0.54	6.52
7TIM	1.97	5.98	6.23	**1.93**
Number of best	25	12	6	7
Success rate (RMSD<2 Å)	66.00%	48.00%	54.00%	46.00%

4.4. Analysis of docking results

An accurate scoring function is essential for solving the protein–ligand docking problem because it guides the conformational search of a docking method to obtain a native-like ligand pose. The proposed ABC-EDM incorporates the Vina function as the scoring function. The relationship between the docking accuracy (RMSD) and the scoring function is investigated in this section. Figs. 4 and 5 display the relationship between the scoring value and the RMSD value for the 50 docking instances. For each instance, a total of 210 docking results (each one of the seven algorithms performs 30 independent runs) are scatter plotted. In addition, the Spearman's correlation coefficient (r) between the scoring value and the RMSD value is also calculated, as shown in Figs. 4 and 5. It can be observed that the scoring value has a positive correlation with the RMSD value, except 1ETA, 2CHT and 2GBP. This finding meets our expectation that a ligand pose with a lower scoring value generally has a higher docking accuracy (i.e., a smaller RMSD value).

Figs. 4 and 5 also show that the scoring function is not very precise, because the positive correlation between the scoring value and

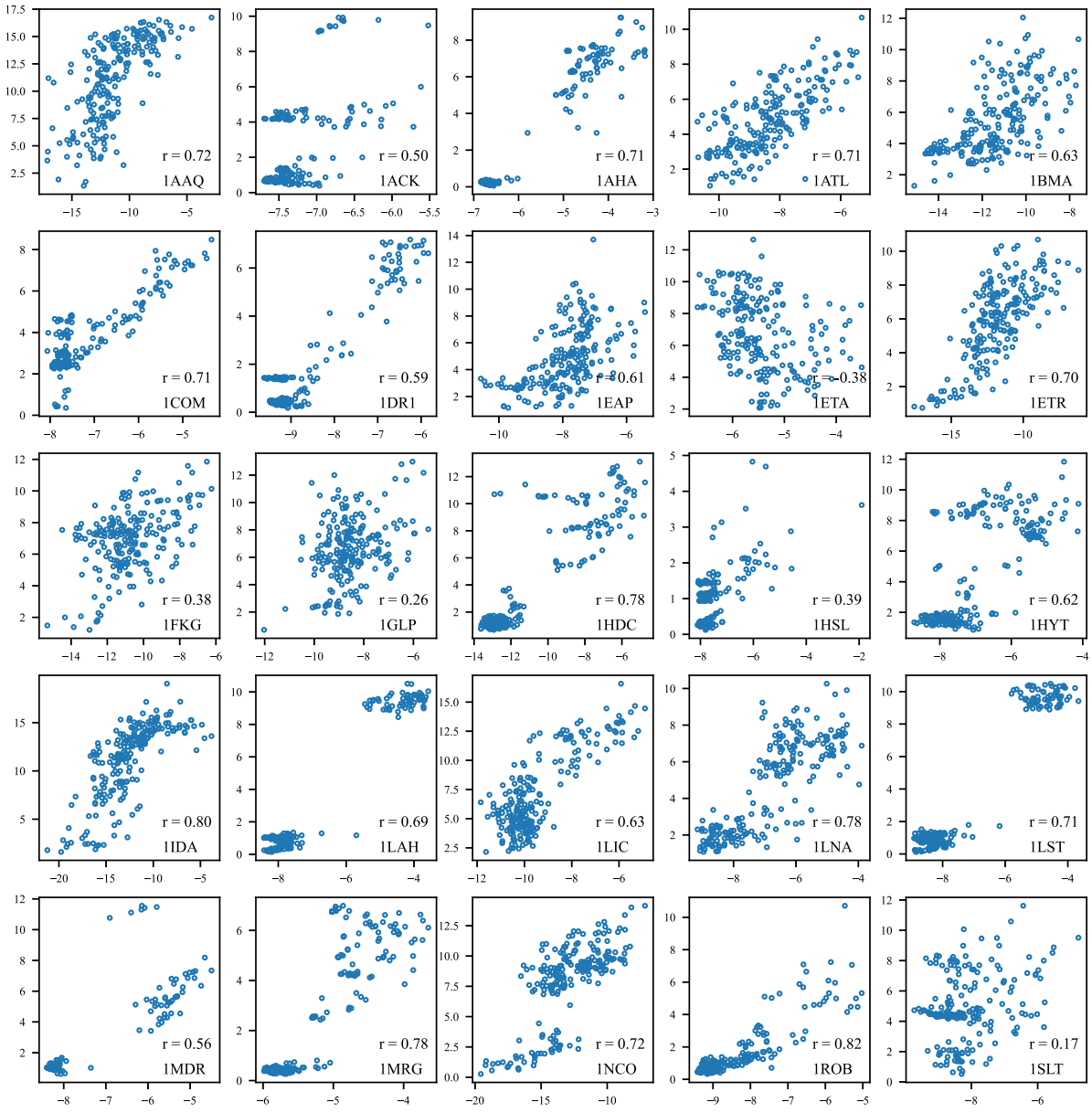


Fig. 4. The scoring value (horizontal axis) versus RMSD value (vertical axis) of 210 docking results are scatter plotted for the 25 docking instances. The Spearman's correlation coefficient (r) is also calculated.

the RMSD value is not considered strong in some cases, e.g., 1FKG, 1GLP, 1HSL, 1SLT, 1TMN, and 2MCP. This exhibition suggests that the landscape of the scoring function contains many local minima. A docking algorithm can be easily trapped in local minima during the search process. This is a main reason why the solutions in Figs. 4 and 5 are grouped into different numbers of clusters, especially for 1HYT, 1LAH, 1LST, 2ADA, 3CPA, and 3TPI. This finding implies that a successful docking method should not only pay attention to improving its optimization ability but also take advantage of the characteristics of the protein–ligand docking problem during the docking process. In fact, as shown in Table 6, DE produces a competitive optimization result compared with ABC-EDM. However, as exhibited in Table 8, DE produces worse docking results, and ABC-EDM is significantly better than DE in terms of docking accuracy. This finding indicates that a docking algorithm should not merely pursue optimization performance during the conformational search. Taking advantage of the characteristics of the protein–ligand docking problem contributes to solving this problem.

Moreover, we provide a three-dimensional visualization of the docking results to show how the predicted ligands are bound to the protein receptors. The docking results (the ligand poses with the lowest scoring values) of three typical instances, i.e., 1BMA, 1HYT, and 2CGR, are plotted in Fig. 6. Obviously, ABC-EDM provides more suitable solutions compared with the other algorithms because the ligand poses predicted by ABC-EDM are more similar to the native ligand poses. ABC-EDM can locate the ligand in the closer position of the native ligand than the other algorithms. Specifically, the ligand of 1BMA has a larger size (12 active rotatable bonds). ABC-EDM has an overwhelming advantage over the other algorithms for 1BMA, as shown in Fig. 6(a). In general, it can be concluded that ABC-EDM can provide a suitable solution for the protein–ligand docking problem.

4.5. Comparing ABC-EDM with AutoDock vina

We compare the docking results obtained by the proposed ABC-EDM approach with the state-of-the-art AutoDock Vina [8]. Vina is a classic molecular docking program and has been recognized as one of the most

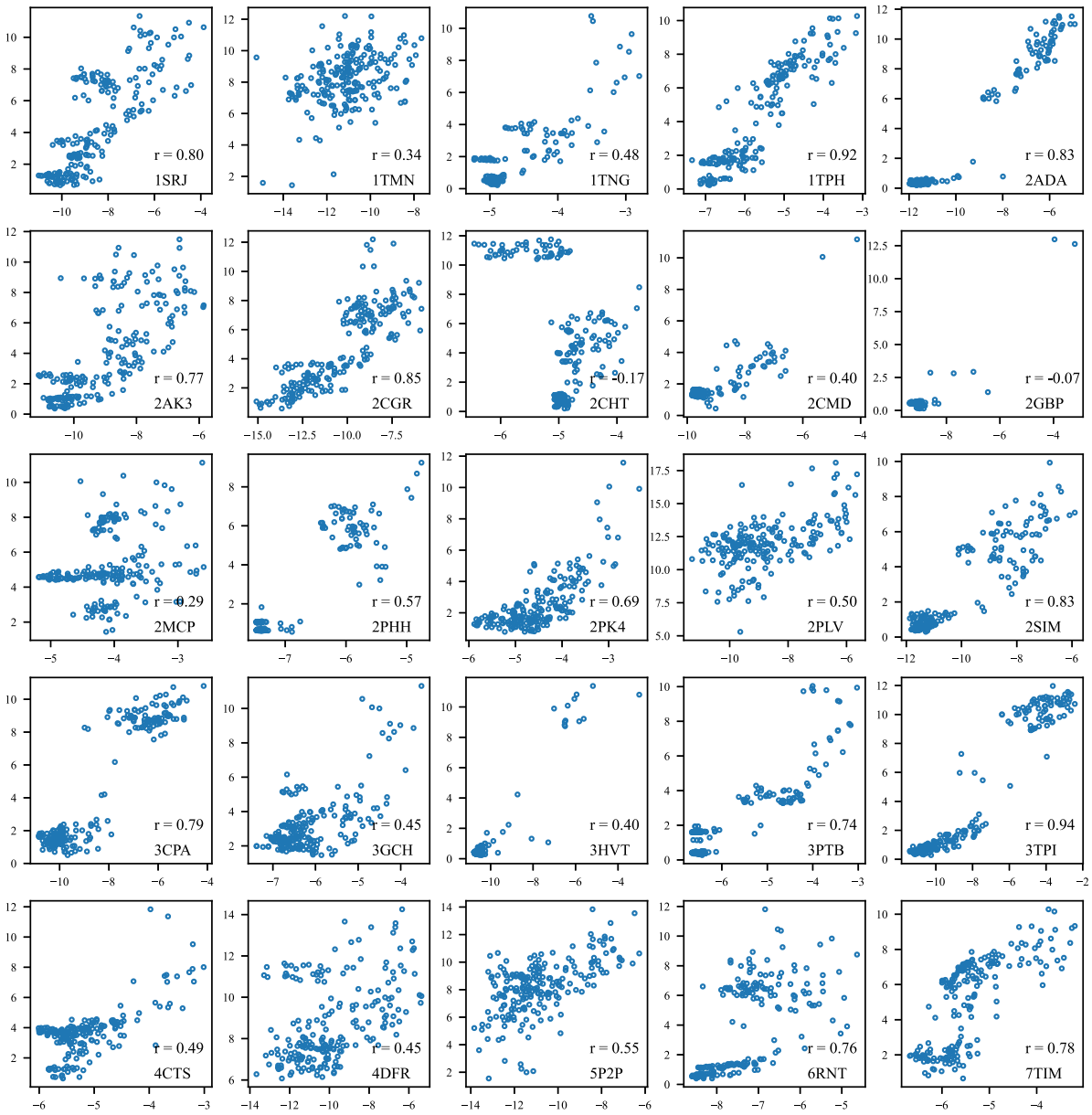


Fig. 5. The scoring value (horizontal axis) versus RMSD value (vertical axis) of 210 docking results are scatter plotted for the other 25 docking instances. The Spearman's correlation coefficient (r) is also calculated.

Table 9
Comparison of the success rates of ABC-EDM and Vina on the GOLD dataset (containing 113 docking instances) at three typical RMSD thresholds.

RMSD threshold	ABC-EDM	Vina
1.00 Å	30.09%	21.24%
2.00 Å	61.06%	55.75%
5.00 Å	84.07%	79.65%

popular protein–ligand docking approaches in the literature [27,48]. Vina shares the same scoring function as ABC-EDM but has a different search method. In Vina, a modified Monte Carlo simulation method coupled with a local search based on BFGS algorithm is employed as the search method. The GOLD dataset is used to evaluate the docking performance of ABC-EDM and Vina. The protein–ligand complexes with structural errors have been removed, and a set of 113 protein–ligand complexes are retained as the benchmark dataset.

The parameters of Vina are set according to the recommendation [8], and the experimental setup is set to be same as ABC-EDM. We execute Vina on the 113 docking instances. The docking simulation of Vina is performed 30 times on an instance, and the ligand pose with the lowest scoring value is outputted as the final docking result. Table 9 summarizes the success rates of ABC-EDM and Vina on the benchmark dataset. ABC-EDM achieves the better success rates of 30.09% (RMSD<1 Å), 61.06% (RMSD<2 Å), and 86.73% (RMSD<5 Å) in comparison with Vina. Fig. 7 gives the detailed comparison of the success rates at different RMSD thresholds. This comparison result illustrates that the proposed ABC-EDM approach can provide better docking results in comparison with Vina.

Finally, we note that the BFGS-based local search in Vina is a quasi-Newton optimization method. This means that the gradient of the scoring function needs to be calculated. Consequently, more computational resources are required during docking simulation. The success of Vina is partly due to the sophisticated gradient calculation method. Compared with Vina, ABC-EDM does not require gradient calculation but takes advantage of the characteristics of the protein–ligand docking problem. A proprietary search mechanism based on EDA is designed to

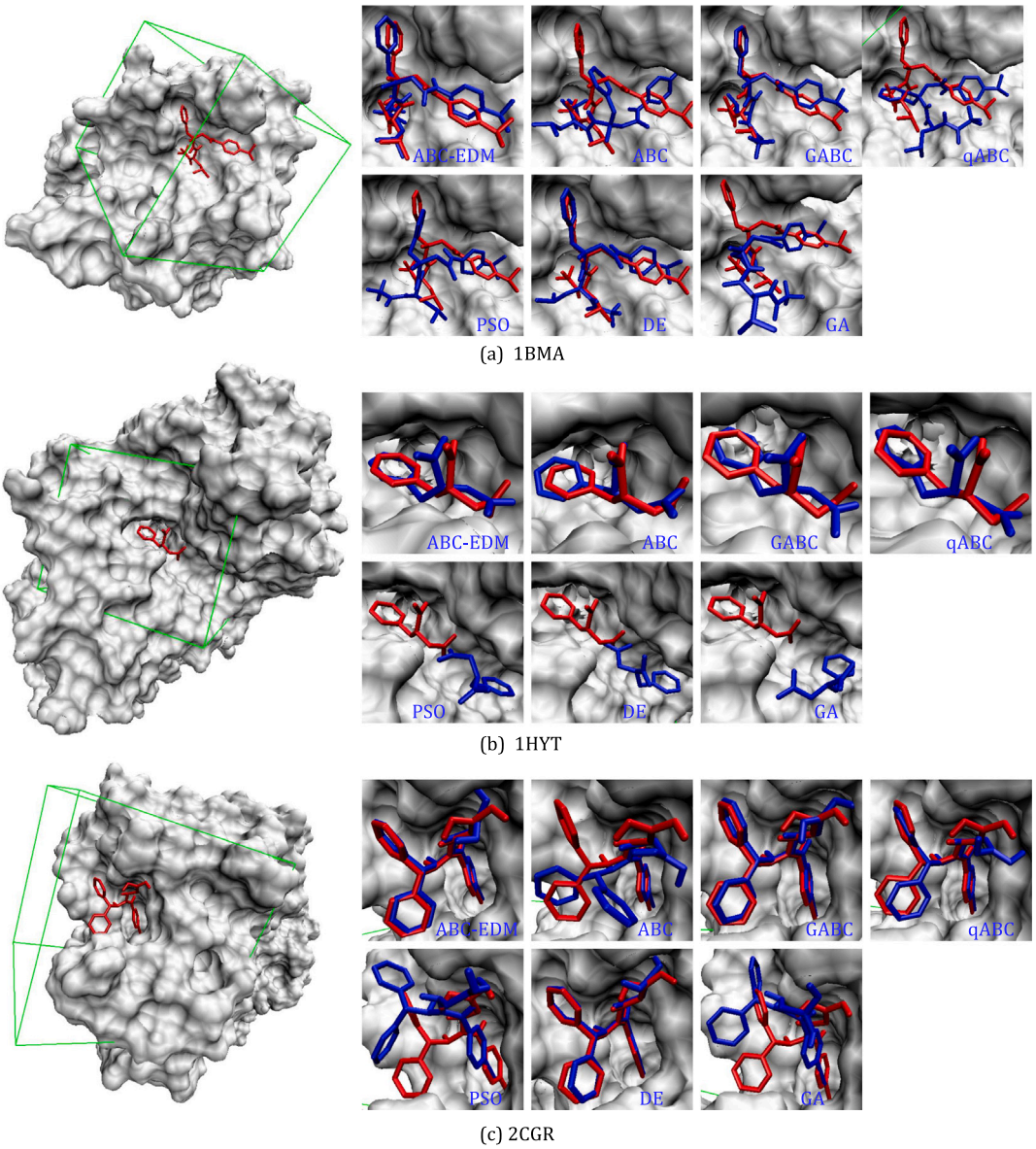


Fig. 6. Three-dimensional visualizations of the docking results for the three docking instances. For each instance, the rugged surface of the protein receptor is displayed in the left figure, and the green cuboid exhibits the three-dimensional search space of the binding sites. The native ligand poses are labeled in red, and the predicted ligand poses obtained by the different algorithms are labeled in blue.

improve the performance of ABC-EDM. Therefore, the proposed search strategy in ABC-EDM is considered more straightforward and concise.

5. Conclusions and future work

In this study, we proposed an approach called ABC-EDM to solve the protein–ligand docking problem. ABC-EDM employed the empirical scoring function of AutoDock Vina as the scoring function. Considering the powerful performance of ABC algorithms, we used a modified ABC to constitute the core search strategy of ABC-EDM. By further analyzing the characteristics of the protein–ligand docking problem, we proposed a proprietary search mechanism inspired by EDA to enhance the search performance. Later, the effectiveness of ABC-EDM was verified on many docking instances. The experimental results demonstrated the superiority of ABC-EDM in comparison with the other seven methods, and ABC-EDM achieved a success rate 5% higher than AutoDock Vina on the GOLD dataset. Different from previous works that developed a new search strategy starting from the view of optimization, we took full advantage of problem-specific information and designed a proprietary search mechanism to enhance the search performance. The experimental results suggested that incorporating the prior knowledge of the protein–ligand docking problem contributes to solving this problem. In fact, the analysis of experimental results indicated that merely pursuing optimization performance while overlooking the characteristics of the protein–ligand docking problem could lead to the degradation of the docking accuracy. The development of proprietary search strategies in protein–ligand docking methods deserves the efforts of researchers.

In the future, we intend to pay continuous attention to the search strategy and the scoring function to develop more effective protein–ligand docking methods. The adoption of more powerful evolutionary computation techniques in protein–ligand docking is worth investigating. Specifically, taking advantage of more prior knowledge and designing problem-specific mechanisms in docking methods deserve future efforts. Moreover, it is reasonable to incorporate other categories of scoring functions to enhance docking methods, such as knowledge-based scoring functions and machine-learning-based scoring functions.

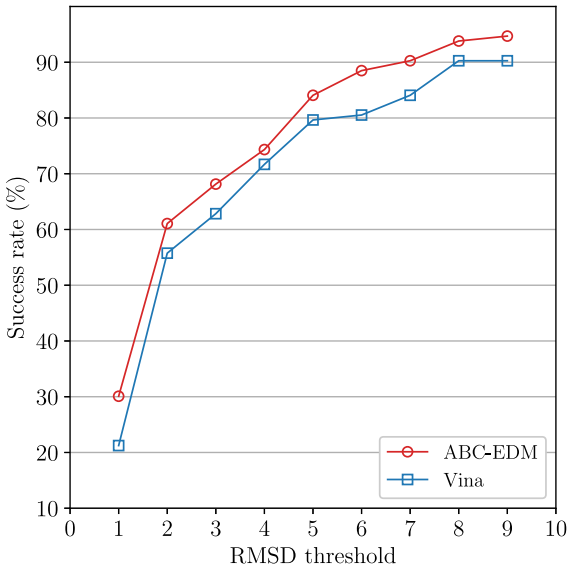


Fig. 7. Comparison of the success rates of ABC-EDM and Vina on the GOLD dataset (containing 113 docking instances).

CRediT authorship contribution statement

Shuangbao Song: Writing – original draft, Visualization, Software, Methodology, Funding acquisition, Conceptualization. **Cheng Tang:** Writing – review & editing, Visualization, Validation, Methodology, Investigation, Formal analysis. **Zhenyu Song:** Writing – review & editing, Visualization, Validation, Resources, Investigation, Formal analysis. **Jia Qu:** Writing – review & editing, Supervision, Software, Project administration, Investigation, Funding acquisition. **Xingqian Chen:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20220619), and the National Natural Science Foundation of China (Grant No. 62203069).

References

[1] K. Wu, E. Karapetyan, J. Schloss, J. Vadgama, Y. Wu, Advancements in small molecule drug design: A structural perspective, Drug Discov. Today (2023) 103730.
[2] W. Dai, B. Zhang, X.-M. Jiang, H. Su, J. Li, Y. Zhao, X. Xie, Z. Jin, J. Peng, F. Liu, et al., Structure-based design of antiviral drug candidates targeting the SARS-CoV-2 main protease, Science 368 (6497) (2020) 1331–1335.
[3] C. Isert, K. Atz, G. Schneider, Structure-based drug design with geometric deep learning, Curr. Opin. Struct. Biol. 79 (2023) 102548.
[4] C. Gorgulla, A. Boeszoermenyi, Z.-F. Wang, P.D. Fischer, P.W. Coote, K.M.P. Das, Y.S. Malets, D.S. Radchenko, Y.S. Moroz, D.A. Scott, et al., An open-source drug discovery platform enables ultra-large virtual screens, Nature 580 (7805) (2020) 663–668.
[5] P. Śledź, A. Caflisch, Protein structure-based drug design: from docking to molecular dynamics, Curr. Opin. Struct. Biol. 48 (2018) 93–102.
[6] C. Shen, J. Ding, Z. Wang, D. Cao, X. Ding, T. Hou, From machine learning to deep learning: Advances in scoring functions for protein–ligand docking, Wiley Interdiscip. Rev. Comput. Mol. Sci. 10 (1) (2020) e1429.
[7] S. Yin, L. Biedermannova, J. Vondrasek, N.V. Dokholyan, MedusaScore: an accurate force field-based scoring function for virtual drug screening, J. Chem. Inf. Model. 48 (8) (2008) 1656–1662.
[8] O. Trott, A.J. Olson, AutoDock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, J. Comput. Chem. 31 (2) (2010) 455–461.
[9] M. Kadukova, K.d.S. Machado, P. Chacón, S. Grudinin, KORP-PL: a coarse-grained knowledge-based scoring function for protein–ligand interactions, Bioinformatics 37 (7) (2021) 943–950.
[10] D.D. Wang, M. Zhu, H. Yan, Computationally predicting binding affinity in protein–ligand complexes: free energy-based simulations and machine learning-based scoring functions, Brief. Bioinform. 22 (3) (2021) bbaa107.
[11] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P.N. Suganthan, C.A. Coello Coello, F. Herrera, Bio-inspired computation: Where we stand and what's next, Swarm Evol. Comput. 48 (2019) 220–250.
[12] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471.
[13] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (7) (2010) 3166–3173.
[14] D. Karaboga, B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, Appl. Soft Comput. 23 (2014) 227–238.
[15] J. Ji, S. Song, C. Tang, S. Gao, Z. Tang, Y. Todo, An artificial bee colony algorithm search guided by scale-free networks, Inform. Sci. 473 (2019) 142–165.
[16] T. Ye, W. Wang, H. Wang, Z. Cui, Y. Wang, J. Zhao, M. Hu, Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure, Knowl.-Based Syst. 241 (2022) 108306.
[17] Ş. Öztürk, R. Ahmad, N. Akhtar, Variants of Artificial Bee Colony algorithm and its applications in medical image processing, Appl. Soft Comput. (2020) 106799.
[18] Y. Cui, W. Hu, A. Rahmani, A reinforcement learning based artificial bee colony algorithm with application in robot path planning, Expert Syst. Appl. 203 (2022) 117389.
[19] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm Evol. Comput. 1 (3) (2011) 111–128.
[20] B. Doerr, M.S. Krejca, Significance-based estimation-of-distribution algorithms, IEEE Trans. Evol. Comput. 24 (6) (2019) 1025–1034.
[21] Z.-Q. Zhang, R. Hu, B. Qian, H.-P. Jin, L. Wang, J.-B. Yang, A matrix cube-based estimation of distribution algorithm for the energy-efficient distributed assembly permutation flow-shop scheduling problem, Expert Syst. Appl. 194 (2022) 116484.
[22] A. Shirazi, Robust estimation of distribution algorithms via fitness landscape analysis for optimal low-thrust orbital maneuvers, Appl. Soft Comput. 144 (2023) 110473.
[23] I. Tubert-Brohman, W. Sherman, M. Repasky, T. Beuming, Improved docking of polypeptides with Glide, J. Chem. Inf. Model. 53 (7) (2013) 1689–1699.
[24] M.L. Verdonk, G. Chessari, J.C. Cole, M.J. Hartshorn, C.W. Murray, J.W.M. Nissink, R.D. Taylor, R. Taylor, Modeling water molecules in protein- ligand docking using GOLD, J. Med. Chem. 48 (20) (2005) 6504–6515.
[25] W.J. Allen, T.E. Balius, S. Mukherjee, S.R. Brozell, D.T. Moustakas, P.T. Lang, D.A. Case, I.D. Kuntz, R.C. Rizzo, DOCK 6: Impact of new features and current docking performance, J. Comput. Chem. 36 (15) (2015) 1132–1156.
[26] G.M. Morris, R. Huey, W. Lindstrom, M.F. Sanner, R.K. Belew, D.S. Goodsell, A.J. Olson, AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility, J. Comput. Chem. 30 (16) (2009) 2785–2791.
[27] S. Forli, R. Huey, M.E. Pique, M.F. Sanner, D.S. Goodsell, A.J. Olson, Computational protein–ligand docking and virtual drug screening with the AutoDock suite, Nat. Protoc. 11 (5) (2016) 905–919.
[28] W. Zhang, E.W. Bell, M. Yin, Y. Zhang, EDock: blind protein–ligand docking by replica-exchange monte carlo simulation, J. Cheminform. 12 (2020) 1–17.
[29] S. Song, J. Ji, X. Chen, S. Gao, Z. Tang, Y. Todo, Adoption of an improved PSO to explore a compound multi-objective energy function in protein structure prediction, Appl. Soft Comput. 72 (2018) 539–551.
[30] X. Chen, S. Song, J. Ji, Z. Tang, Y. Todo, Incorporating a multiobjective knowledge-based energy function into differential evolution for protein structure prediction, Inform. Sci. 540 (2020) 69–88.
[31] P.F. Leonhart, E. Spieler, R. Ligabue-Braun, M. Dorn, A biased random key genetic algorithm for the protein-ligand docking problem, Soft Comput. 23 (12) (2019) 4155–4176.
[32] L.E. Prentis, C.D. Singleton, J.D. Bickel, W.J. Allen, R.C. Rizzo, A molecular evolution algorithm for ligand design in DOCK, J. Comput. Chem. 43 (29) (2022) 1942–1963.
[33] M.C. Ng, S. Fong, S.W. Siu, PSOVina: The hybrid particle swarm optimization algorithm for protein-ligand docking, J. Bioinform. Comput. Biol. 13 (03) (2015) 1541007.

[34] H.K. Tai, S.A. Jusoh, S.W. Siu, Chaos-embedded particle swarm optimization approach for protein-ligand docking and virtual screening, J. Cheminform. 10 (1) (2018) 1–13.
[35] Y. Fu, Z. Chen, J. Sun, Random drift particle swarm optimisation algorithm for highly flexible protein-ligand docking, J. Theoret. Biol. 457 (2018) 180–189.
[36] J. Li, C. Li, J. Sun, V. Palade, Rdpsovina: the random drift particle swarm optimization for protein–ligand docking, J. Comput. Aided Mol. Des. 36 (6) (2022) 415–425.
[37] S. Song, X. Chen, Y. Zhang, Z. Tang, Y. Todo, Protein-ligand docking using differential evolution with an adaptive mechanism, Knowl.-Based Syst. 231 (2021) 107433.
[38] J. Ji, J. Zhou, Z. Yang, Q. Lin, C.A.C. Coello, AutoDock koto: A gradient boosting differential evolution for molecular docking, IEEE Trans. Evol. Comput. (2022) 1, http://dx.doi.org/10.1109/TEVC.2022.3225632.
[39] S. Uehara, K.J. Fujimoto, S. Tanaka, Protein-ligand docking using fitness learning-based artificial bee colony with proximity stimuli, Phys. Chem. Chem. Phys. 17 (25) (2015) 16412–16417.
[40] B. Guan, C. Zhang, Y. Zhao, An efficient ABC_DE_based hybrid algorithm for protein–ligand docking, Int. J. Mol. Sci. 19 (4) (2018) 1181.
[41] M.J. García-Godoy, E. López-Camacho, J. García-Nieto, J. Del Ser, A.J. Nebro, J.F. Aldana-Montes, Bio-inspired optimization for the molecular docking problem: state of the art, recent results and perspectives, Appl. Soft Comput. 79 (2019) 30–45.
[42] H. Li, K.-S. Leung, P.J. Ballester, M.-H. Wong, istar: A web platform for large-scale protein-ligand docking, PLoS One 9 (1) (2014) e85678.
[43] Z. Liu, Y. Li, L. Han, J. Li, J. Liu, Z. Zhao, W. Nie, Y. Liu, R. Wang, PDB-wide collection of binding data: current status of the pdbbind database, Bioinformatics 31 (3) (2015) 405–412.
[44] S.K. Burley, H.M. Berman, C. Bhikadiya, C. Bi, L. Chen, L. Di Costanzo, C. Christie, K. Dalenberg, J.M. Duarte, S. Dutta, et al., RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy, Nucleic Acids Res. 47 (D1) (2019) D464–D474.
[45] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064.
[46] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Comput. Surv. 45 (3) (2013) 1–33.
[47] E. López-Camacho, M.J.G. Godoy, J. Garcia-Nieto, A.J. Nebro, J.F. Aldana-Montes, Solving molecular flexible docking problems with metaheuristics: A comparative study, Appl. Soft Comput. 28 (2015) 379–393.
[48] J. Eberhardt, D. Santos-Martins, A.F. Tillack, S. Forli, AutoDock vina 1.2.0: New docking methods, expanded force field, and python bindings, J. Chem. Inf. Model. 61 (8) (2021) 3891–3898.