View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by CURVE/open

An Effective Swarm Intelligence Optimization Algorithm for Flexible Ligand Docking

Li, C., Sun, J., Li, L-W., Wu, X. & Palade, V.

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:
Li, C, Sun, J, Li, L-W, Wu, X & Palade, V 2021, 'An Effective Swarm Intelligence Optimization Algorithm for Flexible Ligand Docking', IEEE/ACM Transactions on Computational Biology and Bioinformatics.
https://dx.doi.org/10.1109/TCBB.2021.3103777

DOI 10.1109/TCBB.2021.3103777
ISSN 1545-5963
ESSN 1557-9964

Publisher: IEEE

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

An Effective Swarm Intelligence Optimization Algorithm for Flexible Ligand Docking

Chao Li, Jun Sun, Li-Wei Li, Xiaojun Wu, Vasile Palade

Abstract—In general, flexible ligand docking is used for docking simulations under the premise that the position of the binding site is already known, and meanwhile, it can also be used without prior knowledge of the binding site. However, most of the optimization search algorithms used in popular docking software are far from being ideal in the first case, and they can hardly be directly utilized for the latter case due to the relatively large search area. In order to design an algorithm that can flexibly adapt to different sizes of the search area, we propose an effective swarm intelligence optimization algorithm in this paper, called diversity-controlled Lamarckian quantum particle swarm optimization (DCL-QPSO). The highlights of the algorithm are a diversity-controlled strategy and a modified local search method. Integrated with the docking environment of Autodock, the DCL-QPSO is compared with Autodock Vina, Glide and other two Autodock-based search algorithms for flexible ligand docking. Experimental results revealed that the proposed algorithm has a performance comparable to those of Autodock Vina and Glide for dockings within a certain area around the binding sites, and is a more effective solver than all the compared methods for dockings without prior knowledge of the binding sites.

Index Terms—Flexible ligand docking, Search algorithm, Quantum particle swarm optimization, Diversity-controlled strategy, Solis and Wets local search, Autodock

1 Introduction

Molecular docking methods are of utmost importance and have been widely used in drug discovery and other academic research projects [1], [2]. Generally, its aim is to predict the experimental binding modes and affinities of ligand molecules within the particular receptor targets. For the docking problems with small ligands and large protein receptors, flexible ligand docking is currently the most widely adopted method [3], because of its excellent balance between the computational efficiency and docking accuracy. This docking process can be simulated by numerous docking programs, and the quality of the simulation results depends on two factors, i.e. the search algorithm and the scoring function [4]. In flexible ligand docking, the search algorithm explores suitable translations, orientations and conformations of a ligand, with the protein considered as a rigid object. The scoring function is the objective function (or fitness function) that guides the search algorithm during the search process and is also used to evaluate the quality of the docking conformations. This work focuses on improving search algorithms for flexible ligand docking.

During the past few decades, many docking software packages were developed to solve the flexible docking problem, such as Autodock [5], [6], Autodock Vina (referred to as Vina) [7], GOLD [8], Surflex [9], DOCK [10], Gilde [11], to name a few. Among them, Autodock is widely used and attracts many researchers to make improvements on it, since it is open source and can be easily implemented. For the same reason, the work we undertake in this paper is based on the latest version of Autodock software (version 4.2.6). Autodock 4.2.6 adopts the Lamarckian genetic algorithm (LGA) [6] (i.e., a hybrid of the genetic algorithm (GA) and the Solis and Wets local search (SWLS) [12] method) as its default search algorithm, and a semi-empirical force field as its scoring function.

Generally, most of the released docking software packages adopt flexible ligand docking under the premise that the binding site is already known, so that the area where the possible binding sites are located should be determined in advance [13], and several approaches can be used to solve this problem [14], [15], [16], [17]. After the binding area is determined, the search algorithm is then used to find candidates for active compounds within this binding area. The binding area is definitely restricted in a small area rather than the entire surface of the protein, and thus the search algorithm is dedicated to solving a smaller-scale optimization problem.

However, sometimes we need to dock a ligand to the whole surface of a protein without prior knowledge of the binding site, that is, to find potential binding sites or directly reproduce the crystallographic docking pose. Such an operation is known as blind docking [18]. The significant difference between blind docking and the docking approaches mentioned above is that the search algorithms used for blind docking should find candidates by scanning the entire surface of protein targets [19]. It should be noted that in blind docking, both the ligands and the proteins can also be rigid and flexible, and therefore flexible ligand docking methods can also be easily applied in blind docking. Thus, in order to distinguish between the flexible ligand docking around the binding site and blind docking with flexible ligand, in this paper, the former one is called "normal flexible ligand docking" or "normal docking", and the latter one "blind flexible ligand docking" or "blind docking".

In terms of the search algorithms used for normal flexible ligand docking, many optimization algorithms have been

- *Chao Li, Jun Sun, Wei-Li and Xiaojun Wu are with the Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Wuxi, 214122, China. E-mail: sunjun_wx@hotmail.com.*
- *Vasile Palade is with Faculty of Engineering and Computing, Coventry University, Coventry, CV1 5FB, UK.*

proposed to improve the docking performance, and some of them have been used as search algorithms in Autodock. Simulated annealing was used as default algorithm in earlier version of Autodock [6], but it cannot perform so well as LGA when handling ligands with more degrees of freedom. In SODOCK [3], the PSO algorithm combined with the SWLS method was proposed for solving highly flexible ligand docking problems, and this search algorithm has already been integrated in Autodock 4.2.6. FIPSDock [20] is derived from the fully informed PSO and a newly developed energy function for improving the accuracy of docking. Both SODOCK and FIPSDock have better docking performance than LGA, but have relatively low docking efficiency due to the implementation of neighbourhood topology [3], [20]. A more recent variant of Autodock, known as FWADOCK [21], utilized an improved fireworks optimization method as its search algorithm, but it can only be comparable to some other variants of Autodock with other different optimization algorithms [21]. When it comes to blind flexible ligand docking, only a small amount of research focuses on applying optimization algorithms to sample the whole energy landscape surface [22], [23], [24], since it is very difficult for an optimization algorithm to reproduce the crystallographic complex directly by scanning a much larger search area. Therefore, the docking performance of most existing optimization algorithms can still be further improved on normal flexible ligand docking, and few of the algorithms can be suitable on both normal and blind dockings.

In order to provide an effective optimization search algorithm, which can adapt to different sizes of the search areas so that it can be used in the docking project no matter whether the prior knowledge of binding site is known or not, we propose in this paper a novel diversity-controlled hybrid optimization algorithm. The main body of this algorithm is the quantum particle swarm optimization (QPSO), combining with the attractor guided (AG) strategy based on a modified definition of diversity. A modified SWLS (MSWLS) method is also integrated into the hybrid algorithm to further improve the quality of the final search results. The diversity-controlled strategy used in QPSO and the modification in the local search method can make the algorithm search in an appropriate area in terms of the size of the entire search scope, which allows this algorithm to be able to be used for both normal and blind dockings. The implementation of the proposed hybrid algorithm adopts the environment and energy function of Autodock 4.2.6. Compared with LGA [6], SODOCK [3], Vina [7] and Gilde [11] for both normal and blind flexible ligand docking, the proposed hybrid algorithm is comparable to Vina and Glide for normal docking, and much better than all the other compared docking methods for blind docking, as shown by experimental results.

The rest of the paper is structured as follows. In Section 2, we present in detail a modified definition of swarm diversity, and based on this, the diversity-controlled strategy for QPSO and the modified local search method are proposed, which both contribute to the search performance of the hybrid search algorithm. Section 3 describes the dataset for testing and the experimental settings for all compared docking methods. Section 4 presents the docking experimental results and comparative analysis between the proposed method and other docking approaches. Finally, some concluding remarks are given in Section 5.

2 Diversity Guided Lamarckian Quantum Particle Swarm Optimization

2.1 Quantum Particle Swarm Optimization

Particle swarm optimization (PSO) is an important metaheuristic algorithm and is inspired by social behavior of bird flocks and was first proposed by Eberhart and Kennedy [25]. The PSO algorithm performs an optimization task by iteratively improving a swarm of candidate solutions with respect to an objective (fitness) function. In a PSO with M individuals to solve an N-dimensional real problem, each particle i $(1 \le i \le M)$ is a candidate solution, representing a foraging bird in a flock. At the n^{th} iteration, the current position vector and the velocity vector of particle i denote those of the bird, which can be expressed as $X_{i,n} = (X_{i,n}^1, X_{i,n}^2, \cdots, X_{i,n}^N)$ and $V_{i,n} = (V_{i,n}^1, V_{i,n}^2, \cdots, V_{i,n}^N)$, respectively. With respect to the PSO algorithm applied to a docking problem, each dimension in $X_{i,n}$ has its specific meaning: the first three are Cartesian coordinates for the ligand translation, the following four define a quaternion specifying the ligand orientation, and each of the remaining ones represents a ligand torsion. Thus, $X_{i,n}$ can express a ligand pose at n^{th} iteration during the docking process.

In the canonical PSO, the particle moves according to the following equations [26]:

$$V_{i,n+1}^j = wV_{i,n}^j + c_1 r_{i,n}^j\left(P_{i,n}^j - X_{i,n}^j\right) + c_2 R_{i,n}^j\left(G_n^j - X_{i,n}^j\right) \quad (1)$$

$$X_{i,n+1}^j = X_{i,n}^j + V_{i,n+1}^j \quad (2)$$

for $i = 1,2 \cdots, M; j = 1,2 \cdots, N,$ where c_1 and c_2 are known as the acceleration coefficients. The vector $P_{i,n} = (P_{i,n}^1, P_{i,n}^2, \cdots, P_{i,n}^N)$ is the personal best (pbest) position of particle i, records its previous best position. The vector $G_n = (G_n^1, G_n^2, \cdots, G_n^N)$ is the best one according to the objective function among all the pbest positions in the particle swarm and is called the global best (gbest) position. The movement of each particle according to these two equations is somewhat similar to the flight path of each bird determined by its own information and that shared by the whole bird flock. When the search process is terminated, the current gbest position is the solution we finally found. In equation (1), $r_{i,n}^j$ and $R_{i,n}^j$ are two different sequences of random numbers uniformly distriuted on the interval (0,1), and w is the inertia weight and is set as a linearly decreasing value in canonical PSO [27]. $V_{i,n}^j$ and $X_{i,n}^j$ should be restricted with $[V_{min}^j, V_{max}^j]$ and $[X_{min}^j, X_{max}^j]$, respectively during the search process. All these value ranges are set in advance according to the specific situation.

QPSO is a variant version of PSO, motivated by concepts from quantum mechanics [28] and the trajectory analysis of the canonical PSO in [26]. The trajectory analysis in [26] demonstrated that each particle i in canonical PSO moves in a trajectory with its local focus (i.e. local attractor) $p_{i,n} = (p_{i,n}^1, p_{i,n}^2, \cdots, p_{i,n}^N)$, and each dimension of $p_{i,n}$ is given by:

$$p_{i,n}^j = \frac{c_1 r_{i,n}^j P_{i,n}^j + c_2 R_{i,n}^j G_n^j}{c_1 r_{i,n}^j + c_2 R_{i,n}^j} \qquad (3)$$

In the canonical PSO, both acceleration coefficients are set to be equal, i.e. $c_1 = c_2$. Therefore, equation (3) is equivalent to

$$p_{i,n}^j = \gamma_{i,n}^j P_{i,n}^j + (1 - \gamma_{i,n}^j) G_n^j \quad \gamma_{i,n}^j \sim U(0,1) \qquad (4)$$

where $\gamma_{i,n}^j$ is a sequence of random numbers uniformly distribute on the interval (0,1).

Under the above premise, it is assumed that each particle i in QPSO is treated as a spin-less one moving in a quantum space and is attracted by its local focus, with a δ potential well centered at $p_{i,n}^j$ in the jth dimension $(1 \le j \le N)$. Thus, in each dimension, we can solve the Schrödinger equation and obtain the wave function for $X_{i,n+1}^j$ and then get the corresponding probability distribution function as

$$F(X_{i,n+1}^j) = 1 - \exp(-2|X_{i,n+1}^j - p_{i,n}^j| / L_{i,n}^j) \qquad (5)$$

where $L_{i,n}^j$ is the characteristic length of the wave function, which determines the search scope of each particle in each dimension. Employing the Monte Carlo method, we can get $X_{i,n+1}^j$ as

$$X_{i,n+1}^j = p_{i,n}^j \pm \frac{L_{i,n}^j}{2} \ln(1/u_{i,n+1}^j),\ u_{i,n+1}^j \sim U(0,1) \qquad (6)$$

where $u_{i,n+1}^j$ is the sequence of random numbers uniformly distributed on the interval (0,1). In the canonical QPSO, the $L_{i,n}^j$ is given by

$$L_{i,n}^j = 2\alpha |X_{i,n}^j - C_n^j| \qquad (7)$$

where $C_n = (C_n^1, C_n^2, \cdots, C_n^j)$ is known as the mean best (mbest) position, which is defined as the mean of the pbest positions of all the particles in the swarm, tht is, $C_n^j = \frac{1}{M}\sum_i^M P_{i,n}^j$, $(1 \le j \le N)$. Therefore, we can get the update equation of each particle's position as

$$X_{i,n+1}^j = p_{i,n}^j \pm \alpha |X_{i,n}^j - C_n^j| \ln(1/u_{i,n+1}^j) \qquad (8)$$

α in equations (7) and (8) is known as the contraction-expansion (CE) coefficient, which can be tuned to control the convergence speed of the particles in QPSO. As QPSO can get better performance than many other variants of PSO for most practical problems [29], in this paper, it is used as the core part of the search algorithm in docking. According to the discussion in [29], when α decreases linearly from 1.0 to 0.5, QPSO can obtain generally better performance than other parameter configurations. In the following sections, the QPSO algorithm and its improved version also adopt this parameter setting.

2.2 Definition of Normalized Swarm Diversity

Although QPSO has a good algorithmic performance, like other PSO variants and evolutionary algorithms, it still inevitably encounters premature convergence when solving complex optimization problems such as flexible ligand docking, as will be shown in experimental results in the next section. Therefore, a novel diversity-controlled strategy is proposed in this paper to further enhance the search ability of the QPSO algorithm used for flexible ligand docking.

For the PSO algorithms, a genotype diversity measure is widely used, which defined as the average distance to the mean position of the particle swarm [30], [31], [32]:

$$\begin{aligned} D(X_n) &= \frac{1}{M \cdot S} \sum_{i=1}^{M} [\sum_{j=1}^{N} [X_{i,n}^j - \overline{X_n^j}]^2]^{1/2} \\ &= \frac{1}{M \cdot S} \sum_{i=1}^{M} |X_{i,n} - \overline{X_n}| \end{aligned} \qquad (9)$$

where S is the diagonal length of the search space, represents the scope of the search area. $\overline{X_n^j}$ is the component of mean current position of all the particles in the j^{th} dimension. $D(X_n)$ represents the dispersion of the particle swarm over the whole search space at the n^{th} iteration. By controlling the diverisity, the particle swarm is able to maintain its search ability and avoid premature convergence as much as possible during the search process.

However, such a definition of diversity in equation (9) is probelmetic when it is used in practical applications, since the magnitudes are different in each dimension of the search space and the dimensions with larger magnitudes have gereater impact on the value of $D(X_n)$ than those with smaller magnitudes. For example, in the flexible ligand docking, as the decision variables are positions or oritentions so that their magnitudes are different from each other. Therfore, in this paper, we propose a modified diversity measure, in which the magnitude of each dimension is linearly normalized to ensure that each dimension has the same contribution to swarm diversity:

$$Z_{i,n}^j = (X_{i,n}^j - X_{min}^j) / (X_{max}^j - X_{min}^j) \qquad (10)$$

$$\begin{aligned} ND(X_n) &= \frac{1}{M \cdot \sqrt{N}} \sum_{i=1}^{M} [\sum_{j=1}^{N} [Z_{i,n}^j - \overline{Z_n^j}]^2]^{1/2} \\ &= \frac{1}{M \cdot \sqrt{N}} \sum_{i=1}^{M} |Z_{i,n} - \overline{Z_n}| \end{aligned} \qquad (11)$$

where $[X_{min}^j, X_{max}^j]$ is the range of the jth diemsnion of search space, and $Z_{i,n}^j$ is the normalized value of $X_{i,n}^j$. Replacing $X_{i,n}^j$ in equation (9) by $Z_{i,n}^j$, we get the normalized swarm diversity $ND(X_n)$ given by equation (11), in which, the diagonal length of the search space is $\sqrt{N}$, since the normalized range of each dimension is assumed to be $[0, 1]$.

2.3 Attractor-Guided Strategy in Quantum Particle Swarm Optimization

In order to use QPSO to solve docking problems effectively, the global search ability of QPSO should be enhanced to avoid premature convergence. A strong local search ability is also necessary to get a final result with higher quality. As swarm diversity measures the dispersion of the particles, the search directions of particles can be adjusted according to the current value of swarm diversity to enable the particle swarm to search more globally or more locally. In order to guide the particles' search directions without abandoning the search properties of QPSO, based on the definition of the normalized swarm diversity, we propose a novel diversity guided strategy for the QPSO algorithm as described below.

According to equations (4) and (8), the motion direction of each particle in QPSO is influenced by three other positions, namely, the particle's own pbest position, the gbest position, and the mbest position. In this paper, the three kinds of positions are defined as the "attractors" of the related particle, and the particles' search directions can be guided by adjusting the positions of attractors. Therefore, the novel update equation of each particle's position can be expressed by

$$A(p_{i,n}^{j}) = \gamma_{i,n}^{j} A(P_{i,n}^{j}) + (1-\gamma_{i,n}^{j}) G_{n}^{j}, \qquad \gamma_{i,n}^{j} \sim U(0,1) \quad (12)$$

$$X_{i,n+1}^{j} = A(p_{i,n}^{j}) \pm \alpha |X_{i,n}^{j} - A(C_{n}^{j})| \ln(1/u_{i,n+1}^{j}) \quad (13)$$

where $A(P_{i,n}^{j})$ and $A(C_{n}^{j})$ are the adjusted values of $P_{i,n}^{j}$ and C_{n}^{j}, respectively. The positions of these attractors should be set according to specific situation, which will be elaborated below. It should be noted that the gbest position is not adjusted in the proposed diversiy-controlled strategy, since this historical best solution found by the particle swarm attracts all the particles for convergence of the algorithm. The adjustment of the attractors is the most distinctive characteristics of the proposed strategy, and correspondingly this strategy is called "attractor-guided (AG) strategy".

In the AG strategy, two types of swarm diversities should be measured. One is the position diversity $ND(X_n)$, i.e., the normalized swarm diversity of all particles' current positions, and the other is the cognitive diversity $ND(P_n)$, defined as the normalized swarm diversity of all the pbest positions. Since all the attractors are related to pbest positions, both of the two diversities should not decrease rapidly, otherwise premature convergence can be resulted in. Considering that $ND(P_n)$ keeps declining in most of the search process, a linearly decreasing baseline B_n of cognitive diversity is set according to equation (14), indicating the desired decline rate of $ND(P_n)$, and then the appropriate distribution area of pbest positions can guide the particles to search within an appropriate scope.

$$B_n = (1 - n/n_{max}) * (B_0 - B_{n_{max}}) + B_{n_{max}} \quad (14)$$

where n_{max} is the maximum number of iterations, and B_0 is the initial value of B_n set to 0.25, which is the diversity value representing the uniform distribution of all the particles on the whole search space. $B_{n_{max}}$ is the end value calculated by:

$$B_{n_{max}} = \frac{1}{M \cdot \sqrt{N}} \sum\nolimits_{i=1}^{M} [\sum\nolimits_{j=1}^{N} [er \times 0.5]^2]^{1/2} = er/2 \quad (15)$$

where er is the ratio of the diagonal length of the desirable search scope at the end of the search process to that of the entire search space. er should be low enough to make sure QPSO is able to search in a small scope during the later stage of the search process, so as to find a final result with higher quality. By using "0.5" in equation (15), we mean it is assumed that the mean current position of all the particles is at the center of search space, such that the distance of the mean position to the edge of the normalized search space should be 0.5. In this paper, er is set empirically to be 1×10^{-4}, corresponding to $B_{n_{max}} = 5 \times 10^{-5}$.

With the set baseline, the whole search process of the QPSO algorithm can be divided into three phases according to the comparison of the baseline with the $ND(X_n)$ and $ND(P_n)$. In these phases, the positions of attractors are adjusted to control the search behavior of all particles, which are descried as follows.

A. Normal phase

When $ND(X_n)$ is larger than the baseline and $ND(P_n)$ lower than the baseline, $A(P_{i,n}^{j})$ and $A(C_{n}^{j})$ are just set to be the same as $P_{i,n}^{j}$ and C_{n}^{j}, which means in this situation the algorithm searches as in the general setting. The reason is that the large difference between $ND(X_n)$ and $ND(P_n)$ in this phase promotes the particles with the QPSO mechanism to search in a relatively large scope, which makes it hard for the $ND(P_n)$ to sharply decrease, and thus gives the particles a good balance between global search and local search.

B. Divergence phase

When both $ND(X_n)$ and $ND(P_n)$ are lower than the current value of the baseline, $ND(P_n)$ should stop decreasing or decrease much lower than before to wait for the value of the baseline to go down. However, a relatively small value of $ND(X_n)$ probably leads to constant convergence of the pbest positions. Hence, in this phase, all the particles should diverge to be distributed in a larger area, which can prevent $ND(P_n)$ from declining and in turn enhance the global search ability of the algorithm.

To guide particles to diverge, the attractors, $P_{i,n}^{j}$ and $C_{i,n}^{j}$, should be set far from G_{n}^{j} and $X_{i,n}^{j}$ so that each particle can be pulled away from the center of the swarm. $A(P_{i,n}^{j})$ and $A(C_{i,n}^{j})$ are set as the equations (16) and (17), respectively:

$$A(P_{i,n}^{j}) = \begin{cases} G_{n}^{j} \pm 2 * di * er * srad, & P_{i,n}^{j} = G_{n}^{j} \\ G_{n}^{j} + 2 * di * er * srad, & 0 < P_{i,n}^{j} - G_{n}^{j} < 2 * er * srad \\ G_{n}^{j} - 2 * di * er * srad, & -2 * er * srad < P_{i,n}^{j} - G_{n}^{j} < 0 \\ G_{n}^{j} + 2 * di * (P_{i,n}^{j} - G_{n}^{j}), & otherwise \end{cases} \quad (16)$$

$$A(C_{i,n}^{j}) = \begin{cases} X_{i,n}^{j} \pm di * er * srad, & C_{n}^{j} = X_{i,n}^{j} \\ X_{i,n}^{j} + di * er * srad, & 0 < C_{n}^{j} - X_{i,n}^{j} < er * srad \\ X_{i,n}^{j} - di * er * srad, & -er * srad < C_{n}^{j} - X_{i,n}^{j} < 0 \\ X_{i,n}^{j} + 2 * di * (C_{n}^{j} - X_{i,n}^{j}), & otherwise \end{cases} \quad (17)$$

where $srad$ is the half of the search range in the j^{th} dimension, i.e. $srad = (X_{max}^{j} - X_{min}^{j})/2$. $di > 1$ represents the divergence coefficient expressed as:

$$di = \begin{cases} B_0/ND(X_n), & B_{n_{max}} \le L(X_n) < B_n \\ B_0/B_{n_{max}}, & L(X_n) < B_{n_{max}} \end{cases} \quad (18)$$

During the divergence phase, there is little possibility for the algorithm to find a better solution, and thus the algorithm only stays for one iteration each time it comes into this phase. Therefore, di is set to increase with the decline of $ND(X_n)$ in a reciprocal form to force $A(P_{i,n}^{j})$ and $A(C_{i,n}^{j})$ to leave far enough away from G_{n}^{j} and $X_{i,n}^{j}$ according to equations (16) and (17), respectively, with the $ND(X_n)$ increasing to B_0. In equation (16), G_{n}^{j} is set as the reference point to $P_{i,n}^{j}$, since G_{n}^{j} is the best one among all $P_{i,n}^{j}$. In equation (17), $X_{i,n}^{j}$ is the reference point to $C_{i,n}^{j}$, since the second term on the right side of equation (8) is determined by the difference between C_{n}^{j} and $X_{i,n}^{j}$. As indicated by equation (17), $A(C_{i,n}^{j})$ is set according to the distance of C_{n}^{j} to $X_{i,n}^{j}$, otherwise if C_n^j is too close to $X_{i,n}^j$, it should be determined by the desirable least distance from the reference point, i.e. $er * srad$, without changing its motion direction. Similarly, $A(P_{i,n}^j)$ is set in the same way as $A(C_{i,n}^j)$ according to equation (16). It should be noted that there is a scaling coefficient "2" before er in equation (16), since $p_{i,n}^j$, the linear combination of $P_{i,n}^j$ and G_n^j, leads the expected position of $p_{i,n}^j$ to be the mean position of $P_{i,n}^j$ and G_n^j. $p_{i,n}^j$ in the first term on right side of equation (8) is the real local attractor, so multipling $er * srad$ by 2 in equation (16) can set $A(p_{i,n}^j)$ at the desirable position, making $ND(X_n)$ increase to the ideal value.

C. Acceleration phase

If $ND(P_n)$ is higher than the baseline, the algorithm should change into acceleration phase, making the $ND(X_n)$ rapidly decrease so that $ND(P_n)$ is able to have more opportunity to go down to catch up the decline rate of the baseline. In this phase, only $A(P_{i,n}^j)$ is set to be closer to G_n^j as shown by equation (19), while $A(C_n^j)$ leaves unchanged so that the particles can search in a relatively small area around the gbest position, resulting in enhanced local search ability of QPSO.

$$A(P_{i,n}^j) = G_n^j + ai * (P_{i,n}^j - G_n^j) \tag{19}$$

where $0 < ai < 1$ is the acceleration coefficient, which can be expressed as

$$ai = B_n / ND_s(X_n) \tag{20}$$

where $ND_s(X_n)$ is a constant value during a continuous acceleration phase, which equals to $ND(X_n)$ of the first iteration in such a continuous acceleration phase. When the algorithm stays in a continuous acceleration phase for relatively long time, this setting can make ai keep decreasing with the drop of the baseline, further narrowing the search scope.

2.4 Position Diversity-Controlled Solis and Wets Local Search Method

Like in LGA [6] and SODOCK [3], the SWLS method [12] is also employed in this study. The SWLS method is a stochastic heuristic for continuous search spaces, which introduces a probabilistic element. Its primal purpose is the local optimization of functions that do not provide gradient information [12]. Basically, the SWLS method always starts by a random search step with a normal distribution $N(b, q)$ in each dimension, and, during the evolving process, b and q are adjusted according to the current objective function value of the particle, then the search process should be ended when q is lower than 1% of the start step, or the algorithm meets the maximum number of local search iterations. The specific steps of the SWLS method can be accessed in [3].

Due to the different value ranges of each dimension in docking problems, in LGA and SODOCK, the SWLS method has different constant initial q value in each dimension. Similar to but not the same as these two algorithms, a modified SWLS (MSWLS) method is proposed in this paper, where the initial value of q in the j^{th} dimension should be referred to

$$q^j = ((X_{max}^j - X_{min}^j)/s) * (ND(X_n)/B_0) \tag{21}$$

In equation (21), q^j is related to the value range in dimension j, since for normal docking and blind docking, there are large differences in search ranges of ligands' coordinates. s is the ratio of the maximum initial search step size to the entire search range in each dimension. In this paper, we set empirically $s = 30$. $ND(X_n)/B_0$ measures the size of the current search scope of the QPSO algorithm. Multiplying by this value means the MSWLS method is set to match the QPSO's search behavior. If QPSO searches in a relatively large space, the initial search range of MSWLS method is also large, and vice versa. The termination criterion of this improved SWLS method is unchanged, that is, the lower bound of q should still be 1% of the initial value so that it should be calculated every time the MSWLS method is applied. With respect to other settings in MSWLS, the maximum number of consecutive successes or failures before doubling or halving the local search step size is set to 4, and the maximum number of iterations should be set according to Table 1.

2.5 The Hybrid Search Algorithm for Automated Docking

The proposed hybrid algorithm combining the QPSO algorithm, the AG strategy and the MSWLS method is named as the diversity-controlled Lamarckian QPSO (DCL-QPSO) algorithm, which is implemented in the Autodock 4.2.6. Similar to LGA, the MSWLS method is expected at the end of each iteration, and each particle has a probability of 0.06 to be subjected to local search [6]. With the implementation of the AG strategy based on the new definition of swarm diversity and the improved local search method, the hybrid algorithm has a good balance between global search ability and local search ability no matter how large the search range is. It means that the DCL-QPSO algorithm can be used for automated docking whether the binding site is known or not. The procedure of the hybrid algorithm is outlined below.

```
Algorithm DCL-QPSO ($M, N, X_{\min}, X_{\max}, neval_{max}$)
    $neval = 0$;
    $n = 0$;
    for $i = 1$ to $M$ do
        randomly initialize $X_{i,n}$;
        evaluate docked energy of $X_{i,n}$;
        $P_{i,n} = X_{i,n}$;
    end for
    compute $C_n$ and find $G_n$ among all $P_{i,n}$;
    compute $ND(X_n)$ and $ND(P_n)$ using equation (11);
    while $neval < neval_{max}$ do
        compute $B_n$ using equation (14);
        $A(P_n) = P_n$;
        $A(C_n) = C_n$;
        if $ND(P_n) < B_n$ and $ND(X_n) < B_n$ then
            compute $A(P_n)$ and $A(C_n)$ using equation (16). and (17);
        end if
        if $ND(P_n) > B_n$ then
            compute $A(P_n)$ using equation (19);
        end if
        for $i = 1$ to $M$ do
            for $j = 1$ to $N$ do
```

```
                $A(p_{i,n}^j) = \gamma_{i,n}^j A(P_{i,n}^j) + (1-\gamma_{i,n}^j)G_n^j;$
                $X_{i,n+1}^j = A(p_{i,n}^j) \pm \alpha|X_{i,n}^j - A(C_n^j)|\ln(1/u_{i,n+1}^j);$
                **if** $X_{i,n+1}^j > X_{max}^j$ **or** $X_{i,n+1}^j < X_{min}^j$ **then**
                    set $X_{i,n+1}^j$ a random value within its search range;
                **end if**
            **end for**
            evaluate docked energy of $X_{i,n+1}$;
            $neval = neval + 1;$
        **end for**
        compute $ND(X_n)$ using equation (11);
        Apply the MSWLS method with a probability of 0.06 to each particle in $X_{i,n+1}$;
        $neval = neval + neval_{ls};$
        update $P_n$, compute $C_n$ and find $G_n$;
        compute $ND(X_n)$ and $ND(P_n)$ using equation (11);
        $n = n + 1;$
    **end while**
```

In the DCL-QPSO algorithm, $neval$ is the number of objective function evaluations. $neval_{max}$ is the maximum number of the evaluations, and $neval_{ls}$ means the number of the objective function evaluations by the MSWLS method. In Appendix A, our preliminary experiments and the corresponding analysis have proved the effectiveness of the diversity-controlled strategies in improving the search ability of the QPSO algorithm from an algorithmic perspective.

3 Experimental Setups

3.1 Dataset

To evaluate the performance of the DCL-QPSO algorithm for docking applications, the EDock dataset was used for testing the proposed algorithm [33]. This dataset consists of 102 test cases from DUDE [34] and 433 test cases from COACH [35]. All the 102 test cases in DUDE were chosen from a diverse family, including 26 kinases, 15 proteases, 11 nuclear receptors, 5 GPCRs, 2 ion channels, 2 cytochrome P450s, 36 other enzymes, and 5 miscellaneous proteins. It should be pointed out that for each test case in DUDE, only the active compound is used in docking experiments, while the decoy compounds are skipped since they do not have a native pose that can be compared with the predicted conformation. The original COACH dataset contains 500 non-redundant proteins that harbor 812 ligands (410 natural ligand, 238 drug-like ligand and 164 metal ions) [35]. Eliminating the targets with metal ions and large ligands from the original COACH dataset, the remaining 433 test cases are suitable for protein-ligand docking experiments. Consequently, the EDock dataset totally contains 535 test cases. The number of torsions of the ligands in these test cases ranges from 0 to 24, representing that the search dimension varies from 7 to 31, so that the performance of the proposed algorithm can be evaluated comprehensively on the problems with different search dimensions.

3.2 Docking preparation and experimental settings

In this paper, the docking simulations comprise two parts, i.e., the normal flexible ligand docking and the blind flexible ligand docking. In each part, all the 535 test cases in EDock dataset were employed, and five docking methods, including DCL-QPSO, LGA [6], SODOCK [3], Vina (version 1.1.2) [7] and Glide (version 7.8, integrated in Schrodinger 2018-2) [11] were compared in terms of docking performance. The LGA was selected since it is still the default algorithm used in the latest version of Autodock. The SODOCK was chosen as a compared algorithm from many proposed optimization algorithms integrated with Autodock, since in [36] Guo et al. has proved that the SODOCK generally has better docking accuracy and robustness than many other Autodock-based algorithms. Vina was developed in order to improve the docking speed and accuracy of Autodock4, and thus this program was often used to compare with Autodock in many aspects [7], [37], [38] so that it was also employed in our experiments. As a commercial docking program with high robustness [37], Glide is another comparative docking program not based on Autodock used in our experiments. In this paper, all the docking experiments were run on a personal computer with an Intel® i7-6850 8-core 3.60GHz processor, a 12-GB RAM and an Ubuntu 16.04 Linux platform. The following paragraph in this subsection describes the specific parameter settings of each docking method in detail.

Three Autodock-based docking algorithms, including DCL-QPSO, LGA and SODOCK, employed the docking environment and scoring function of Autodock 4.2.6. The semi-empirical force field of Autodock 4.2.6 was used as the scoring function to evaluate conformations in docking simulations, with the total energy including the intermolecular and intramolecular interaction energies. Each term in the force field contains evaluations for dispersion/repulsion, hydrogen bonding, electrostatics, and desolvation. More detailed explanation of this force field can be found in [39].

With respect to the experiment settings in the Autodock environment, 50 particles, 5×10^5 energy function evaluations, 100 local search maximum iterations and 30 docking times were utilized for each normal docking test case; 100 particles, 3×10^6 energy function evaluations, 300 local search maximum iterations and 50 docking times were applied to each blind docking test case. The AutodockTools within MGLTools (version 1.5.6) was used to generate PDBQT format files of the receptor and ligand, and to add hydrogens to the receptor for each test case. The energy grid maps were calculated with AutoGrid. For normal flexible ligand docking, the grid size was set to 60×60×60 points with a spacing of 0.375Å (i.e., a cube with an edge length of 22.5 Å), and the center of the grid map was set as the center of the reference ligand for each test case. For blind flexible ligand docking, the grid size was set to a cube such that it can cover the whole protein for each test case and the spacing was also set to 0.375Å. Within the corresponding search range of each docking trial, the component in each dimension of each particle's position was randomly initialized. Besides, with respect to the specific setting in each Autodock-based algorithm, the DCL-QPSO algorithm utilized the parameter setting specified in Section 2, and those of LGA and SODOCK were just as recommended in [6] and [3], respectively.

For Vina, the PDBQT files and the grid maps were also prepared by using AutodockTools and AutoGrid, respectively.

For both normal and flexible ligand docking on each test case, the grid maps were set to be the same as those for Autodock-based programs. The maximum energy difference between the best binding mode and the worst one reported in the result files was set to 10 kcal/mol, in order to generate enough modes with different energy level. For each docking test, up to 20 (maximum number in Vina) binding modes were generated and the exhaustiveness was set to 8.

The programs in Schrodinger (version 2018-2) were used for the docking preparation of Glide [11]. For both normal and flexible ligand docking, the grid center and the box size were set to be the same as those applied to Autodock-based programs. Docking precision mode was set to SP. The maximum number of the poses reported for each docking experiment was set to 50. All the other parameters were set to default values in Gilde.

3.3 Performance metrics

In this paper, the comparisons in terms of energy-related results are only done among DCL-QPSO, LGA and SODOCK, since their energy values are all calculated by the same scoring function. However, in Autodock 4.2.6, the energy value calculated by the force field (i.e., the objective function value) is not used to rank the final docking conformations of multiple trials in a single docking problem, but the binding free energy [6] is used instead. Its value can be easily calculated since it is part of the scoring function value, which is the sum of the intermolecular energy and the torsional free energy. In Autodock-based algorithms, the binding free energy is only reported immediately when the search process is finished. After all the required trials are ended for a docking test case, the value of the binding free energy is used to rank all the final docking conformations, since it is more instructive for the selection and specific research of the final conformations than the energy value calculated by the force field. The binding free energy does not include the internal or intramolecular interaction energy of the ligand, since this part of energy cannot improve the accuracy of the binding free energy model but really affects the docking results during the search process. Therefore, the value of the binding free energy, rather than the scoring function value, is used as one performance metric for the Autodock-based algorithms in this paper.

For evaluating the similarity between the produced conformation and the crystallographic one, the root mean squared deviation (RMSD) between these two conformations was calculated. In this paper, the RMSD value was calculated by using all the heavy atoms of the ligand without considering the symmetry. This calculation of RMSD is stricter than the default one applied in Autodock, which considers the symmetry of the conformations [39]. In this paper, two kinds of widely used RMSD, i.e., the best-scored RMSD and the best-sampled RMSD, are evaluated to compare docking programs with different scoring functions [40], [41]. The former is the RMSD between the reference structure and the conformation with the lowest score or binding free energy, and the latter is the smallest RMSD among the RMSDs between all the produced conformations and the crystal one. A common threshold of RMSD is set to 2Å, which is used to determine whether the crystal structure was successfully reproduced.

Another specific performance metric employed in this paper is the highest cluster rank of all the successful docking conformations (referred to top-success cluster rank), which is only used to evaluate the results for blind flexible ligand docking. In the Autodock program, all the found conformations for a docking test are classified to multiple clusters. In each cluster, the RMSDs between the conformation with the lowest binding free energy and the lowest-energy conformations in all the other clusters are relatively large (>2Å), while the RMSDs between the lowest-energy conformation and all the other conformations in this cluster are smaller than 2Å. Generally, a certain area around the center of a cluster is considered as a potential binding site [22]. Therefore, the researchers can probably investigate the final docking conformations based on the cluster rank, and a higher top-success cluster rank can probably lead users to find the right binding site and binding mode more efficiently. For Vina and Glide, a reported conformation in the result file represents a single cluster, since the found conformations very close to any reported conformation have been eliminated during the docking process. Therefore, the top-success cluster rank of Vina or Glide docking results mentioned in section 4.2 is actually the highest rank of all the successful docking conformations found by the corresponding docking program.

In order to determine whether there is a significant difference between two sets of evaluation data obtained by DCL-QPSO and another compared docking method, we used the Wilcoxon signed ranked test [42] in this paper. Each value in a set of evaluation data represents a specific statistical result (e.g. mean binding free energy) obtained by a docking method after multiple trials for a test case. For evaluating the robustness in terms of the best or the lowest metrics (e.g. the best-sampled RMSD or the lowest binding free energy), the bootstrapping method [43] was used to compute the 95% bootstrap confidence interval for the average value of the corresponding metrics for all test cases. In this paper, the distances from the average value to the upper and lower limits of its confidence interval are called upper error and lower error, respectively. The number of bootstrap samples was set to 2000 in the computations.

4 Results and Discussion

4.1 Normal Flexible Ligand Docking

With respect to normal flexible ligand docking, we firstly compare the statistical results of binding free energies among Autodock-based algorithms. Table 1 and Table 2 illustrate some statistics about the mean binding free energy and lowest successful binding free energy, respectively. The lowest successful binding free energy is the lowest binding free energy among the energies obtained by all the successful docking conformations for each test case. It should be pointed out that the statistical results in Table 2 are calculated based on 416 test cases, since all the three Autodock-based algorithms cannot find successful docking results out of 30 runs for the remaining 119 test cases, whose names are listed in Appendix B. Additionally, if a docking algorithm failed to find a successful docking result for a test case, the lowest successful binding free energy was recorded as 0 in statistics.

According to the average energy and the *P*-values in Table

1, it is obvious that the DCL-QPSO can obtain significantly better results of the mean binding free energy than LGA and SODOCK. The average standard deviation by the DCL-QPSO is the lowest among three compared algorithms, indicating the high robustness of the proposed method. From the statistical results in Table 2, it can be concluded that the DCL-QPSO is also the best choice in finding a successful docking conformation with enough low binding free energy. One reason is that the average energy by the DCL-QPSO algorithm in Table 2 is much lower than those by LGA and SODOCK. Another reason is that both the lower and upper error values obtained by the DCL-QPSO are lower than those by the other two Autodock-based algorithms, which again demonstrates the superiority of the DCL-QPSO algorithm in robustness.

TABLE 1

COMPARISON OF THE MEAN BINDING FREE ENERGY RESULTS AMONG AUTODOCK-BASED ALGORITHMS FOR NORMAL FLEXIBLE LIGAND DOCKING

	Average energy[1]	Average standard deviation[2]	*P*-value[3]
DCL-QPSO	-9.125	0.833	
LGA	-8.291	1.017	3.19e-88
SODOCK	-5.756	0.855	2.11e-74

[1] Average value of the mean binding free energy of all test cases after 30 runs

[2] Average value of the standard deviation of all test cases after 30 runs

[3] *P*-value for the results obtained by the corresponding algorithm and DCL-QPSO

TABLE 2

COMPARISON OF THE LOWEST SUCCESSFUL BINDING FREE ENERGY RESULTS AMONG AUTODOCK-BASED ALGORITHMS FOR NORMAL FLEXIBLE LIGAND DOCKING[1]

	Average energy[2]	Upper error	Lower error	*P*-value[3]
DCL-QPSO	-9.175	0.375	0.400	
LGA	-5.532	0.401	0.435	4.29e-54
SODOCK	-7.231	0.480	0.503	1.55e-28

[1] All the statistics in this table were calculated based on the results for 416 test cases, not including the results for the 119 all-failed ones.

[2] Average value of the lowest successful binding free energy out of 30 runs for all test cases

[3] *P*-value for the results obtained by the corresponding algorithm and DCL-QPSO

Secondly, the docking accuracy obtained by the proposed algorithm was also evaluated for normal flexible ligand docking. Therefore, the statistical results of best-scored RMSD and best-sampled RMSD obtained by five docking methods, i.e. DCL-QPSO, LGA, SODOCK, Vina and Glide, are recorded in Table 3 and Table 4, respectively.

The statistics in Table 3 and Table 4 illustrate that the mean RMSD-related results obtained by the DCL-QPSO are the best ones among all the compared algorithms. For the upper and lower errors in both two tables, the DCL-QPSO is a little worse than LGA, but comparable to SODOCK, and better than Vina and Glide, indicating that the robustness of the DCL-QPSO is the second best among the five compared algorithms in terms of RMSD. However, the *P*-values in these two tables show that Vina is comparable to the DCL-QPSO for the results of best-scored RMSD, and Glide is competitive to the DCL-QPSO for both two kinds of RMSD-related results. This conclusion can also be drawn by the number of successful dockings in Table 3 and Table 4, and in Table 3 Vina is even better than the DCL-QPSO for this evaluation criterion. Therefore, in order to further analyze the docking accuracy of the predicted docking poses obtained by DCL-QPSO, Vina and Glide, we plotted in Fig. 1 and Fig. 2 the histograms of the best-scored RMSD and the best-sampled RMSD with RMSD thresholds being 0.5Å, 1Å, 2Å and 3Å, respectively. Fig. 1 illustrates that the DCL-QPSO can find the conformation very close to the crystallographic one (RMSD<0.5Å) for the least test cases among three compared docking methods, while for Fig. 2, the DCL-QPSO can find a result comparable to Vina and Glide. This probably demonstrates the mismatch between binding free energy and RMSD results for the DCL-QPSO, which will be further discussed in section 4.2. Fig. 1 also shows that although the DCL-QPSO cannot find the most successful docking test cases (less than 2Å) among three docking methods, it can find conformations for more test cases with RMSD between 2Å and 3Å than Vina and Glide, which may also be helpful in real docking applications.

TABLE 3

COMPARISON OF THE BEST-SCORED RMSD RESULTS AMONG FIVE DOCKING METHODS FOR NORMAL FLEXIBLE LIGAND DOCKING

	Average RMSD[1]	Upper error	Lower error	*P*-value[2]	Succ[3]
DCL-QPSO	3.580	0.254	0.256		231
LGA	4.133	0.248	0.247	1.90e-12	161
SODOCK	4.202	0.262	0.245	5.89e-06	178
Vina	3.761	0.292	0.283	0.730	248
Glide	3.815	0.297	0.270	0.699	225

[1] Average value of the best-scored RMSD out of 30 runs for all test cases

[2] *P*-value for the results obtained by the corresponding docking method and DCL-QPSO

[3] The number of the test cases for which the corresponding docking method can obtain a best-scored RMSD lower than 2Å

TABLE 4

COMPARISON OF THE BEST-SAMPLED RMSD RESULTS AMONG FIVE DOCKING METHODS FOR NORMAL FLEXIBLE LIGAND DOCKING

	Average RMSD[1]	Upper error	Lower error	*P*-value[2]	Succ[3]
DCL-QPSO	1.847	0.158	0.136		379
LGA	2.292	0.137	0.127	3.82e-28	288
SODOCK	2.232	0.149	0.136	1.16e-09	296
Vina	1.945	0.150	0.130	0.026	364
Glide	2.264	0.242	0.215	0.337	347

[1] Average value of the best-sampled RMSD out of 30 runs for all test cases

[2] *P*-value for the results obtained by the corresponding docking method and DCL-QPSO

[3] The number of the test cases for which the corresponding docking method can obtain a best-sampled RMSD lower than 2Å

With the above analysis, it can be concluded that for normal flexible ligand docking, the DCL-QPSO is much better than the other two Autodock-based algorithms in all the evaluation criteria, and can be a docking method comparable to Vina and Glide in terms of the RMSD-related results.

4.2 Blind Flexible Ligand Docking

In this subsection, the results of the top-success cluster rank (defined in section 3.3) and the best-sampled RMSD obtained by five docking methods are evaluated. The former metric specifies the probability of quickly finding the right binding site or even the correct docking pose. The latter metric denotes how similar the reproduced conformation can be to the crystallographic one if one evaluates all the generated conformations found by a docking method for a specific test case. What should be pointed here is that the overall statistical results in terms of the best-scored RMSD (see Appendix C) is not discussed in this section. The reason is that for the docking tests with RMSDs lower than 2Å, this performance metric is partially included in the top-success cluster rank (top-success cluster rank = 1), and for some of the other docking tests, the best-scored RMSD can be relatively large values (larger than 10Å), making its statistical results for all the test cases meaningless.

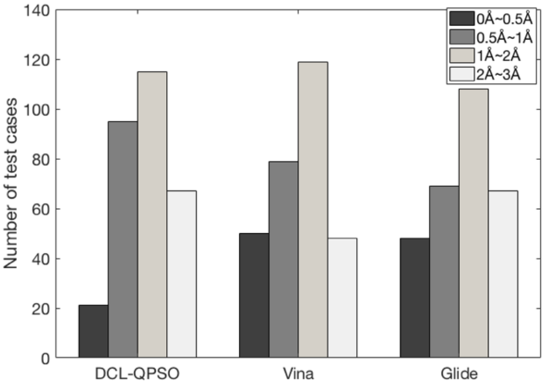

Fig. 1. Histograms of the best-scored RMSD for normal flexible ligand docking obtained by DCL-QPSO, Vina and Glide using RMSD thresholds of 0.5Å, 1Å, 2Å and 3Å.

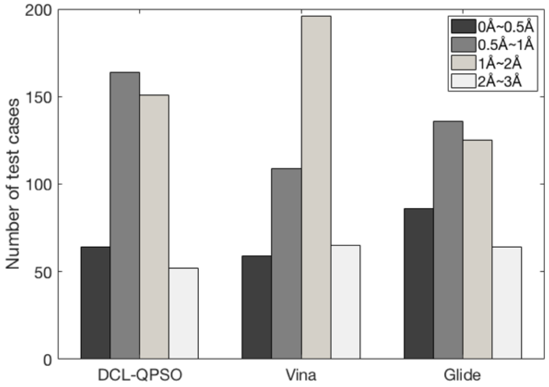

Fig. 2. Histograms of the best-sampled RMSD for normal flexible ligand docking obtained by DCL-QPSO, Vina and Glide using RMSD thresholds of 0.5Å, 1Å, 2Å and 3Å.

TABLE 5
COMPARISON OF THE TOP-SUCCESS CLUSTER RANK RESULTS AMONG FIVE DOCKING METHODS FOR BLIND FLEXIBLE LIGAND DOCKING[1]

	Rank = 1	Rank = 2	Rank > 2
DCL-QPSO	189	40	80
LGA	108	15	34
SODOCK	61	9	32
Vina	135	30	56
Glide	64	13	28

[1] The values in this table illustrates the number of the test case with specific top-success cluster rank

TABLE 6
COMPARISON OF THE BEST-SAMPLED RMSD RESULTS AMONG FIVE DOCKING METHODS FOR BLIND FLEXIBLE LIGAND DOCKING

	Average RMSD[1]	Upper error	Lower error	*P*-value[2]	Succ[3]
DCL-QPSO	4.058	0.485	0.437		309
LGA	5.006	0.443	0.354	8.94e-26	157
SODOCK	6.846	0.482	0.460	1.19e-37	102
Vina	5.074	0.533	0.472	1.36e-08	221
Glide	9.623	0.723	0.639	1.30e-43	105

[1] Average value of the best-sampled RMSD out of 50 runs for all test cases
[2] *P*-value for the results obtained by the corresponding docking method and DCL-QPSO
[3] The number of the test cases for which the corresponding docking method can obtain a best-sampled RMSD lower than 2Å

The statistical results in Table 5 illustrate that the DCL-QPSO can obtain the best result for all evaluation criteria among five compared docking programs, and for more than 74% of successful docking test cases, the DCL-QPSO can obtain the first or the second top-success cluster rank. Moreover, the “Succ” criteria in Table 6 shows that the DCL-QPSO can find at least one successful docking conformation on most of the test cases. This means that for the blind flexible ligand docking, the DCL-QPSO has the strongest ability to find right binding sites or binding modes, and if the DCL-QPSO can reproduce the crystallographic conformation successfully for a test case, there is a high probability of finding a right docking conformation efficiently.

According to Table 6, for almost all the evaluation criteria, the DCL-QPSO algorithm can obtain the best results among five docking methods only except that LGA has the better upper and lower errors than the DCL-QPSO, and the advantage of the DCL-QPSO over the other docking methods is obvious (see *P*-values). This confirms that over all the 535 test cases,

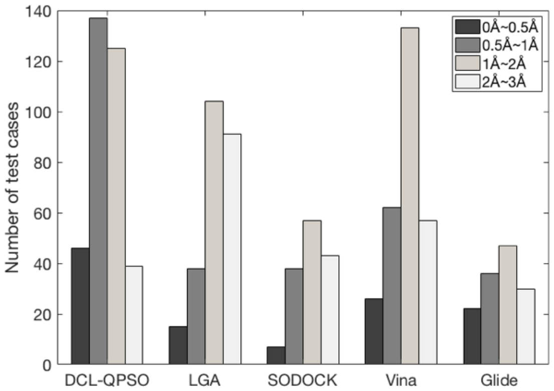

Fig. 3. Histograms of the best-sampled RMSD for blind flexible ligand docking obtained by DCL-QPSO, LGA, SODOCK, Vina and Glide using RMSD thresholds of 0.5Å, 1Å, 2Å and 3Å.

the DCL-QPSO has the best mean performance and the second-best robustness among all compared methods in terms of best-sampled RMSD. It should be pointed out that although Vina and Gilde can obtain comparable results to the DCL-QPSO for normal flexible ligand docking, their RMSD-related results, especially those of Glide, have a relatively large gap with those of the DCL-QPSO for blind docking. Furthermore, the histograms in Fig. 3 show some detailed statistics of the best-sampled RMSD. The results verify the superiority of the DCL-QPSO to all the other docking methods with the thresholds of 0.5Å, 1Å and 2Å, further showing the ability of the DCL-QPSO to find conformations as close as possible to the crystallographic one. Based on the aforementioned statistical results and analysis, we can conlude that unlike the performance in normal flexible ligand docking, the DCL-QPSO does a better job for blind flexible ligand docking than all the other compared docking methods due to the implementation of AG strategy and MSWLS method.

To further illustrate the effectiveness of the diversity-controlled strategies for different sizes of the search scopes, the scatter plot in terms of energy values for blind docking and normal docking obtained by the DCL-QPSO on each test case is illustrated in Fig. 4. Note that the energy evaluated in Fig. 4 is the lowest binding free energy out of multiple trials for a specific test case, regardless of whether the corresponding docking conformation is a successful one or not, since this criterion can represent the search ability of the DCL-QPSO to some extent. According to Fig. 4, only a few points in the upper half of the figure are distributed far from the straight line, which means that the lowest binding free energies found by the DCL-QPSO for blind flexible ligand docking are better than or comparable to those for normal flexible ligand docking on most of the test cases. This indicates the diversity-controlled strategies can definitely help the DCL-QPSO maintain its search ability when the search scope enlarges.

However, the comparison between the RMSDs of the conformations with the lowest binding free energy (i.e. the best-scored RMSDs) for normal docking and blind docking shows a completely different phenomenon. That is, the best-scored RMSDs for blind docking are much larger than those for normal docking on many test cases (see Fig. 5). It means that the lower binding free energy does not always match a smaller RMSD for DCL-QPSO. To illustrate it clearly, we plotted in Fig. 6 the energy versus RMSD of all the conformations found by the DCL-QPSO for blind flexible ligand docking on a specific test case rock1 (the PDB ID of the ligand) as an example. Fig. 6 shows that for some found conformations, the relatively low energies correspond to very large RMSDs (the points at upper left corner in Fig. 6), and the conformations with RMSDs lower or a little larger than 2Å has relatively high binding free energy (the points near the dotted line in Fig. 6). This demonstrates that not most of the low energy values can correspond to the docking poses with small RMSDs, verifying that the whole energy landscape surface of the protein can hardly be sampled by the scoring function used in Autodock 4.2.6. Sometimes this limitation of the scoring function may bring undesirable results: one search algorithm which can only find relatively high binding free energy successfully reproduces the crystallographic complex, while another search algorithm that can find much lower binding free energy fails to obtain a successful best-scored docking conformation (see the corresponding results of LGA and DCL-QPSO on the test case "nram" for normal docking and blind docking in Appendix B and Appendix C, respectively, as an example). As such, our future work is to improve the scoring function of Autodock to reduce the mismatch of energy and RMSD.

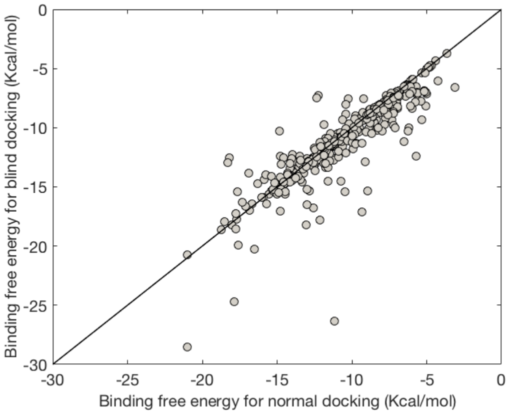

Fig. 4. The scatter plot in terms of the lowest binding free energy for blind docking and normal docking obtained by DCL-QPSO on each test case.

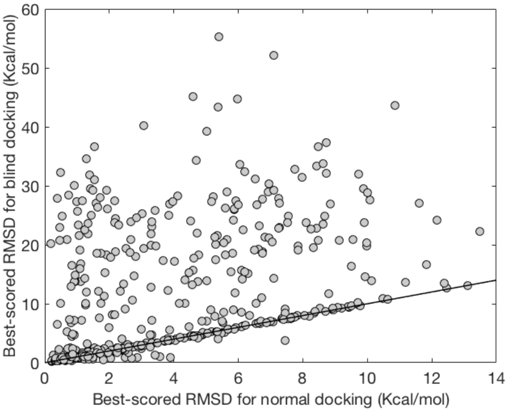

Fig. 5. The scatter plot in terms of best-scored RMSD for blind docking and normal docking obtained by DCL-QPSO on each test case

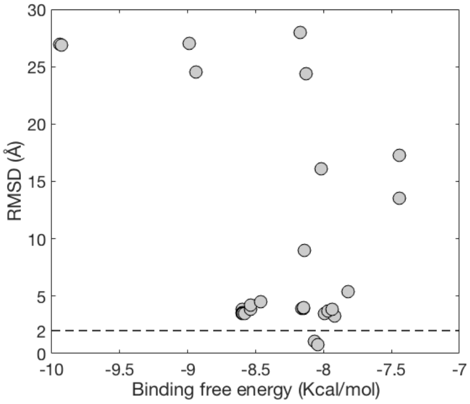

Fig. 6. Energy versus RMSD of all the 50 conformations found by DCL-QPSO for blind flexible ligand docking on rock1.

Although the scoring function in Autodock 4.2.6 has its limitation, the best-scored RMSDs found by the DCL-QPSO for blind docking have a relatively good correlation with those for normal docking. As shown in Table 7, for the test cases on which the docking method can find best-scored RMSD less than 2Å for normal docking, the DCL-QPSO has a much higher rate (66.23%) in finding successful docking conformations in terms of best-scored RMSD for blind docking than all the other docking methods. This shows that when the search scope enlarges, the probability for the algorithm to find a successful best-scored docking conformation will not drop dramatically, verifying the effectiveness of the diversity-controlled strategies again. This characteristic of DCL-QPSO is desirable in real docking applications, since it can provide a certain possibility to help users directly find the right docking pose without having to obtain the suitable binding site in advance.

TABLE 7
THE STATISTICAL RESULTS OF THE NUMBER OF THE SUCCESSFUL DOCKING TEST CASES IN TERMS OF THE BEST-SCORED RMSD FOR BOTH BLIND DOCKING AND NORMAL DOCKING OBTAINED BY DCL-QPSO

	Both success[1]	Normal only[2]	Blind only[3]	Correlation rate[4]
DCL-QPSO	153	78	11	66.23%
LGA	79	82	20	49.07%
SODOCK	48	130	10	26.97%
Vina	125	123	10	50.40%
Glide	49	176	17	21.78%

[1] The number of the test cases for which DCL-QPSO can find successful docking in terms of the best-scored RMSD for both normal and blind docking

[2] The number of the test cases for which DCL-QPSO can find successful docking in terms of the best-scored RMSD only for both normal docking

[3] The number of the test cases for which DCL-QPSO can find successful docking in terms of the best-scored RMSD only for blind docking

[4] Correlation rate = Both success / (Both success + Normal only)

4.3 Docking Time Comparison

TABLE 8
MEAN TIME FOR GENERATING PER DOCKING POSE TAKEN BY EACH COMPARED DOCKING METHOD

	Normal docking	Blind docking
DCL-QPSO	7.40s	57.05s
LGA	28.54s	179.60s
SODOCK	28.74s	198.27s
Vina	7.31s	109.21s
Glide	5.88s	46.95s

In order to evaluate the docking efficiency of all the compared docking methods, we list in Table 8 the mean time of generating per docking pose for normal docking and blind docking. According to Table 8, the DCL-QPSO is much less time-consuming than LGA and SODOCK for both normal and blind docking problems, since it can take much time to exchange elements from genotype to phenotype in LGA [6] and to utilize the neighborhood topology in SODOCK [3]. The mean docking time spent by the DCL-QPSO is equivalent to that by Vina for normal docking, but is about half of that by Vina for blind docking. The reason may be that the total number of steps for searching each docking pose in Vina is associated with the number of atoms, and thus Vina is forced to execute more search steps for blind docking problems. Glide consumes less time than the DCL-QPSO for both normal docking and blind docking, but the difference between them is much smaller than that between the DCL-QPSO and the other two Autodock-based docking methods. Therefore, it can be concluded that the DCL-QPSO is a competitive search algorithm among all the compared docking methods in terms of docking efficiency.

5 CONCLUSIONS

With respect to the optimization search algorithms used for popular docking software packages, when they are used for flexible ligand docking, most of them are far from ideal for normal docking and can be hardly used for real blind docking. Therefore, in this paper, the DCL-QPSO algorithm was proposed to address this problem. This algorithm is a hybrid search method, combining the QPSO algorithm with a novel AG strategy and the MSWLS method. The implementation of the DCL-QPSO adopts the environment and energy function of AutoDock 4.2.6. The source code of DCL-QPSO integrated with Autodock 4.2.6 (named DCLQdock) can be freely downloaded from https://codeocean.com/capsule/8601140/tree. Compared with other four docking methods for normal flexible ligand dockings, the DCL-QPSO algorithm has much better docking performance than LGA and SODOCK, and is comparable to Vina and Glide in terms of RMSD, as shown by the experimental results. For the blind flexible ligand docking problems, the DCL-QPSO is definitely the best one among all the compared methods according to the statistical results on top-success cluster rank and best-sampled RMSD. Moreover, the high correlation between the results obtained by the proposed algorithm for normal docking and those for blind docking illustrates the effectiveness of the AG strategy and MSWLS method for different sizes of search areas. Besides, the docking efficiency of the DCL-QPSO is also acceptable. It should also be pointed out that generally users need not to change the key parameters (i.e., *er* and *s*) in the AG strategy and MSWLS method in most cases. However, if researchers want to modify *er* and *s* for a specific docking test case, some recommendations can be given here. Specifically, increasing *er* and *s*, on one hand, can enhance the robustness but reduce the possibility of finding a conformation with relatively low binding free energy for the DCL-QPSO, which is more suitable for a test case with a very small number of torsions. On the other hand, decreasing *er* and *s* can provide an opportunity for the DCL-QPSO to find a conformation with enough low binding free energy, but can make the algorithm has a higher probability of generating a pose with poor energy value. Thus, users generally need to increase the number of trials for the specific test case when *er* and *s* both decline, in order to give the search algorithm a chance to find low-energy conformations.

Our future work is to modify the scoring function of Autodock to make it more suitable for sampling the energy landscape of the entire surface of a protein. Additionally, we will also apply the DCL-QPSO algorithm in some other docking software packages to test its performance with different scoring functions and different docking environments.

APPENDICES

Appendix A: The experimental results and corresponding analysis for verifying the effectiveness of the diversity-controlled strategies

Appendix B: The statistical results obtained by five compared algorithms for normal flexible ligand docking on each test case in the EDock dataset

Appendix C: The statistical results obtained by five compared algorithms for blind flexible ligand docking on each test case in the EDock dataset

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Projects Numbers: 61673194, 61672263, 61672265), and in part by the national first-class discipline program of Light Industry Technology and Engineering (Project Number: LITE2018-25).

REFERENCES

[1] E. Yuriev, J. Holien, P. A. Ramsland, "Improvements, trends, and new ideas in molecular docking: 2012–2013 in review," Journal of Molecular Recognition, vol. 28, no. 10, pp. 581-604, 2015.

[2] E. López-Camacho, M. J. García Godoy, J. García-Nieto, A.-J. Nebro, J.-F. Aldana-Montes, "Solving molecular flexible docking problems with metaheuristics: A comparative study," Applied Soft Computing, vol. 28, pp. 379-393, 2015.

[3] H. M. Chen, B. F. Liu, H. L. Huang, S. F. Hwang, S. Y. Ho, "SODOCK: Swarm optimization for highly flexible protein-ligand docking," Journal of computational chemistry, vol. 28, no. 2, pp. 612-623, 2008.

[4] S. D. Handoko, X. Ouyang, C. T. T. Su, C. K. Kwoh, Y. S. Ong, "Quick-Vina: accelerating AutoDock Vina using gradient-based heuristics for global optimization," IEEE/ACM transactions on computational biology and bioinformatics, vol. 9, no. 5, pp. 1266-1272, 2012.

[5] D. S. Goodsell, A. J. Olson, "Automated docking of substrates to proteins by simulated annealing," Proteins: Structure, Function, and Bioinformatics, vol. 8, no. 3, pp. 195-202, 1990.

[6] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, A. J. Olson, "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function," Journal of computational chemistry, vo. 19, no. 14, pp. 1639-1662, 1998.

[7] O. Trott, A. J. Olson, "AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," Journal of computational chemistry, vol. 31, no. 2, pp. 455-461, 2010.

[8] R. A. Tahir, S. A. Sehgal, "Pharmacoinformatics and molecular docking studies reveal potential novel compounds against schizophrenia by target SYN II," Combinatorial chemistry & high throughput screening, vol. 21, no. 3, pp. 175-181, 2018.

[9] R. Spitzer, A. N. Jain, "Surflex-Dock: Docking benchmarks and real-world application," Journal of computer-aided molecular design, vol. 26, no. 6, pp. 687-699, 2012.

[10] 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," Journal of computational chemistry, vol. 36, no. 15, pp. 1132-1156, 2015.

[11] H. Alogheli, G. Olanders, W. Schaal, P. BrandtOrcid, A. Karlén, "Docking of Macrocycles: Comparing Rigid and Flexible Docking in Glide," Journal of Chemical Information and Modeling, vol. 57, no. 2, pp. 190–202, 2017.

[12] F. J. Solis, R. J. B. Wets, "Minimization by random search techniques," Mathematics of operations research, vol. 6, no. 1, pp. 19-30, 1981.

[13] W. H. Shin, J. K. Kim, D. S. Kim, C. Seok, "GalaxyDock2: Protein–Ligand Docking Using Beta-Complex and Global Optimization," Journal of computational chemistry, vol. 34, no. 30, pp. 2647–2656, 2013.

[14] Y. Shan, E. T. Kim, M. P. Eastwood, R. O. Dror, M. A. Seeliger, D. E. Shaw, "How does a drug molecule find its target binding site?" Journal of the American Chemical Society, vol. 133, no. 24, pp. 9181-9183, 2011.

[15] R. J. Najmanovich, "Evolutionary studies of ligand binding sites in proteins," Current opinion in structural biology, vol. 45, pp. 85-90, 2017.

[16] G. Sciortino, E. Garribba, J. Rodríguez-Guerra Pedregal, J. D. Maréchal, "Simple Coordination Geometry Descriptors Allow to Accurately Predict Metal-Binding Sites in Proteins," ACS Omega, vol. 4, no. 2, pp. 3726-3731, 2019.

[17] Y. Liu, M. Grimm, W. T. Dai, M. C. Hou, Z. X. Xiao, Y. Cao, "CB-Dock: a web server for cavity detection-guided protein–ligand blind docking," Acta Pharmacologica Sinica, vol. 41, no. 1, pp. 138-144, 2020

[18] H. Pérez-Sánchez, D. T. Kumar, C. G. P. Doss, R. Rodríguez-Schmidt, J. P. Cerón-Carrasco, J. Peña-García, Z. W. Ye, S. Yuan, S. Günther, "Prediction and characterization of influenza virus polymerase inhibitors through blind docking and ligand based virtual screening," Journal of Molecular Liquids, vol. 321, pp. 114784, 2020.

[19] M. Bálint, I. Horváth, N. Mészáros, C. Hetényi, "Towards unraveling the histone code by fragment blind docking," International Journal of Molecular Sciences, vol. 20, no. 2, pp. 422, 2019.

[20] Y. Liu, L. Zhao, W. Li, D. Zhao, M. Song, Y. Yang, "FIPSDock: a new molecular docking technique driven by fully informed swarm optimization algorithm," Journal of computational chemistry, vol. 34, no. 1, pp. 67-75, 2013.

[21] Z. Liu, D. Jiang, C. Zhang, H. Zhao, Q. Zhao, B. Zhang, "A Novel Fireworks Algorithm for the Protein-Ligand Docking on the AutoDock," Mobile Networks & Applications, pp. 1-12, 2019.

[22] C. Hetényi, D. Van Der Spoel, "Efficient docking of peptides to proteins without prior knowledge of the binding site," Protein science, vol. 11, no. 7, pp. 1729-1737, 2002.

[23] A. Grosdidier, V. Zoete, O. Michielin, "Blind docking of 260 protein–ligand complexes with EADock 2.0," Journal of computational chemistry, vol. 30, no. 13, pp. 2021-2030, 2009.

[24] N. M. Hassan, A. A. Alhossary, Y. Mu, C. K. Kwoh, "Protein-ligand blind docking using QuickVina-W with inter-process spatio-temporal integration," Scientific reports, vol. 7, no.1, pp. 1-13, 2017.

[25] J. Kennedy, R. C. Eberhart, "Particle Swarm Optimization," In Proceedings of ICNN'95-International Conference on Neural Networks, IEEE, vol. 4, pp. 1942-1948, 1995.

[26] M. Clerc, J. Kennedy, "The particle swarm-explosion, stability and convergence in a multidimensional complex space," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 58–73, 2002.

[27] E. Assareh, M. A. Behrang, M. R. Assari, A. Ghanbarzadeh, "Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran," Energy, vol. 35, no. 12, pp. 5223-5229, 2010.

[28] J. Sun, B. Feng, W. Xu. "Particle swarm optimization with particles having quantum behavior" In Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), IEEE, vol. 1, pp. 325-331, 2004.

[29] J. Sun, W. Fang, X. Wu, V. Palade, W. Xu, "Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection," Evolutionary computation, vol. 20, no. 3, pp. 349-393, 2012.

[30] R. K. Ursem, "Diversity-guided evolutionary algorithms," In International Conference on Parallel Problem Solving from Nature, Springer, Berlin, Heidelberg, pp. 462-471, 2002.

[31] R. Santos, G. Borges, A. Santos, M. Silva, C. Sales, J. C. Costa, "A semi-autonomous particle swarm optimizer based on gradient information and diversity control for global optimization," Applied Soft Computing, vol. 69, pp. 330-343, 2018.

[32] D. Tian, X. Zhao and Z. Shi, "DMPSO: Diversity-Guided Multi-Mutation Particle Swarm Optimizer," IEEE Access, vol. 7, pp. 124008-124025, 2019.

[33] W. Zhang, E. W. Bell, M. Yin, Y. Zhang, "EDock: blind protein–ligand docking by replica-exchange monte carlo simulation," Journal of Cheminformatics, vol. 12, no.1, pp. 1-17, 2020.

[34] M. M. Mysinger, M. Carchia, J. J. Irwin, B. K. Shoichet, "Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking," Journal of medicinal chemistry, vol. 55, no.14, pp. 6582-6594, 2012.

[35] J. Yang, A. Roy, Y. Zhang, "Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment," Bioinformatics, vol. 29, no. 20, pp. 2588-2595, 2013.

[36] L. Guo, Z. Yan, X. Zheng, L. Hu, Y. Yang, J. Wang, "A comparison of various optimization algorithms of protein–ligand docking programs by fitness accuracy," Journal of Molecular Modeling, vol. 20, no. 7, pp. 2251, 2014.

[37] Z. Wang, H. Sun, X. Yao, D. Li, L. Xu, Y. Li, S. Tian, T. Hou, "Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power," Physical Chemistry Chemical Physics, vol. 18, no. 18, pp. 12964-12975, 2016.

[38] N. T. Nguyen, T. H. Nguyen, T. N. H. Pham, N. T. Huy, M. V. Bay, M. Q. Pham, P. C. Nam, V. V. Vu, S. T. Ngo, "Autodock vina adopts more accurate binding poses but autodock4 forms better binding affinity," Journal of Chemical Information and Modeling, vol. 60, no. 1, pp. 204-211, 2019.

[39] W. Forli, S. Halliday, R. Belew, A. J. Olson, "AutoDock Version 4.2," 2012.

[40] A. S. Hauser, B. Windshügel, "LEADS-PEP: a benchmark data set for assessment of peptide docking performance," Journal of Chemical Information and Modeling, vol. 56, no. 1, pp. 188-200, 2016.

[41] D. Devaurs, D. A. Antunes, S. Hall-Swan, N. Mitchell, M. Moll, G. Lizée, L. E. Kavraki, "Using parallelized incremental meta-docking can solve the conformational sampling issue when docking large ligands to proteins," BMC molecular and cell biology, vol. 20, no. 1, pp. 42, 2019.

[42] S. M. Taheri, G. Hesamian, "A generalization of the Wilcoxon signed-rank test and its applications," Statistical Papers, vol. 54, no. 2, pp. 457-470, 2013.

[43] P. C. D. Hawkins, G. L. Warren, A. G. Skillman, A. Nicholls, "How to do an evaluation: pitfalls and traps," Journal of computer-aided molecular design, vol. 22, no. 3-4, pp. 179–190, 2008.

Chao Li received his Bachelor Degree in Information Engineering from Nanjing University of Information Science and Technology, Nanjing, China, in 2012 and a Master Degree in Computer Science and Technology from Jiangnan University, Wuxi, China, in 2017. He is currently working as a PHD student in Control Science and Engineering in Jiangnan University, Wuxi, China. His research areas are related to computational intelligence, bioinformatics and computer vision. He has published several papers in journals and conferences in the above areas.

Jun Sun received his PhD in control theory and engineering, and an MSc in Computer Science and Technology from Jiangnan University, China, in 2009 and 2003, respectively. He is currently working as a full Professor with the Department of Computer Science and Technology, Jiangnan University, China. He is also Vice-Director of Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangsu Province. His major research areas and work are related to computational intelligence, machine learning, bioinformatics, among others. He published more than 150 papers in journals, conference proceedings and several books in the above areas.

Li-Wei Li received her Bachelor's Degree and Master Degree in Management from Oxford Brookes University and University of London, UK, in 2013 and 2014, respectively. She is currently working as a PHD student in Control Science and Engineering in Jiangnan University, China. Her major research areas are related to intelligent optimization, data mining among others. She has published several papers in the above areas.

Xiaojun Wu received his B.Sc. degree in mathematics from Nanjing Normal University, Nanjing, China, in 1991. He received the M.S. degree in 1996, and the Ph.D. degree in pattern recognition and intelligent systems in 2002, both from Nanjing University of Science and Technology, Nanjing, China. He joined Jiangnan University in 2006, where he is currently a Professor. He has published more than 200 papers in his fields of research. He was a visiting researcher in the Centre for Vision, Speech, and Signal Processing (CVSSP), University of Surrey, U.K., from 2003 to 2004. His current research interests include pattern recognition, computer vision, fuzzy systems, neural networks, and intelligent systems.

Vasile Palade (M'02–SM'04) received his Ph.D. degree from the University of Galati, Romania, in 1999. He worked as a Lecturer with the Department of Computer Science, University of Oxford, U.K., and he is currently a Professor in Artificial Intelligence and Data Science in the Centre for Data Science at Coventry University, UK. His research interests include machine learning with application to computer vision, bioinformatics, fault diagnosis, autonomous cars, web usage mining, and health. He published more than 200 papers in journals and conference proceedings as well as several books in the above areas.