Journal of Bioinformatics and Computational Biology
Vol. 13, No. 3 (2015) 1541007 (18 pages)

DOI: 10.1142/S0219720015410073

Imperial College Press
www.icpress.co.uk

PSOVina: The hybrid particle swarm optimization algorithm for protein–ligand docking

Marcus C. K. Ng*, Simon Fong† and Shirley W. I. Siu‡

Department of Computer and Information Science, University of Macau
Avenida da Universidade, Taipa Macau S.A.R, P. R. China
**marcus.ckng@gmail.com*
†*ccfong@umac.mo*
‡*shirleysiu@umac.mo*

Received 1 September 2014
Revised 17 November 2014
Accepted 7 February 2015
Published 23 March 2015

Protein–ligand docking is an essential step in modern drug discovery process. The challenge here is to accurately predict and efficiently optimize the position and orientation of ligands in the binding pocket of a target protein. In this paper, we present a new method called PSOVina which combined the particle swarm optimization (PSO) algorithm with the efficient Broyden–Fletcher–Goldfarb–Shannon (BFGS) local search method adopted in AutoDock Vina to tackle the conformational search problem in docking. Using a diverse data set of 201 protein–ligand complexes from the PDBbind database and a full set of ligands and decoys for four representative targets from the directory of useful decoys (DUD) virtual screening data set, we assessed the docking performance of PSOVina in comparison to the original Vina program. Our results showed that PSOVina achieves a remarkable execution time reduction of 51–60% without compromising the prediction accuracies in the docking and virtual screening experiments. This improvement in time efficiency makes PSOVina a better choice of a docking tool in large-scale protein–ligand docking applications. Our work lays the foundation for the future development of swarm-based algorithms in molecular docking programs. PSOVina is freely available to non-commercial users at http://cbbio.cis.umac.mo.

Keywords: Particle swarm optimization; protein–ligand docking; flexible docking; conformational search; AutoDock; drug design.

1. Introduction

Protein–ligand docking has become an important step in modern structure-based drug design.[1,2] Given a biological target related to the disease of interest, the docking program helps to decide if a small molecule (the ligand) can bind to the target protein with a desirable level of affinity. High quality docking predictions can reduce the time and cost for experimental tests remarkably and thus both academic and industrial

researches have been focused intensely on improving the accuracy and efficiency in docking algorithms.

Technically speaking, a protein–ligand docking algorithm consists of two main steps: Conformation generation and scoring.[3] The former uses sampling techniques to generate different ligand orientations at different positions inside the protein binding pocket. Each of these conformations will be evaluated by a scoring function and the top scoring ligand conformations will be reported in a ranked list as a result.

In flexible ligand docking, the size of the conformational space or the search space depends on the volume of the protein binding pocket and the number of rotatable bonds of the ligand of interest, while the energy landscape of the search space is determined by the energetic properties of protein–ligand binding which is known to be complex and rugged in shape.[4] To be able to search quickly and intelligently over the huge conformational space, heuristic or metaheuristic algorithms which find near-optimal solutions instead of the global optimum would be a method of choice for initial docking studies or high-throughput virtual screening, from which the potential ligand conformations can be further optimized by expensive but more accurate modeling methods.

In recent years, swarm intelligence algorithms have emerged as a fast and reasonably accurate technique in solving complex search problems in computer science.[5] In particular, the particle swarm optimization (PSO) has gathered much wider interests and has been applied to almost every area in optimization due to its simplicity and flexibility.[6] PSO is a nature-inspired population-based search algorithm which simulates the social behavior of bird flocking or fish school in looking for food. The birds called *particles* are initially dispersed in the search space. In each iteration, a particle moves based on the knowledge of other particles and its own experience to speculate about the promising region to explore. One global best solution is kept updated by all particles and each individual particle also keeps record of its own best solution. Finally, the search terminates until the global best solution is converged or the maximum number of iterations is reached. A study by Dong *et al.*[7] revealed that PSO is suitable for multiple-dimensional, nonlinear and complex problems, however, its applicability on bioinformatics problems, in particular, the protein–ligand docking has not been adequately investigated.

To date, there exists only handful of swarm-based docking methods; they are SODOCK — a hybrid of PSO and Solis and Wets local search method,[8] PLANTS — an ant colony optimization method,[9] PSO@AUTODOCK — a velocity adaptive and regenerative constricted PSO method,[10] ParaDockS — a parallel docking suite having PSO as the optimization algorithm,[11] and FIPSDOCK — the fully informed PSO method.[12] Three of the programs were modifications of the popular open-source docking program AutoDock, albeit different versions, all of them showed better predictive performance when compared to the original AutoDock implementation. Recently, a newly designed and implemented version of the AutoDock program called AutoDock Vina has been released.[13] This version abandoned the former empirical scoring function and GA-based optimizer, but adopted a new knowledge-based

scoring function with Monte Carlo sampling technique and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method for local optimization. Their simulation results showed that a significant improvement was obtained in both prediction accuracy and docking time. It is therefore interesting to see whether swarm intelligence methods such as PSO would further enhance the performance of Vina and how much it can achieve.

In this paper, we present PSOVina — the first PSO protein–ligand docking algorithm in the framework of AutoDock Vina. Through careful integration of Vina's efficient local optimizer into the canonical PSO procedure and proper tuning of parameters, PSOVina achieved a remarkable execution time reduction of 51–60% without compromising the docking accuracies. Our work demonstrates the power of PSO in solving the conformational search problem of protein–ligand docking and provides the foundation for future development of swarm-based algorithms in AutoDock Vina.

The organization of this article is as follows: In Sec. 2, we introduce the data set used in this study, the canonical PSO algorithm and its variants, details of algorithm implementation in Vina and docking simulations performed. Then, in Sec. 3, we present the performance comparisons of PSOVina and Vina in both prediction accuracy and time efficiency. Furthermore, we analyzed the prediction ability of PSOVina based on the ligand size and discussed a few example docking cases.

2. Methods

2.1. *Data set*

To assess the performance of PSOVina in docking and virtual screening, we employed two widely used data sets. The first data set was obtained from the PDBbind database.[14] This is a manually curated database which provides a collection of known three-dimensional structures of protein–ligand complexes with experimentally measured binding affinity data. In each version of the database, it provides three data sets: The *general set* contains all valid complexes; the *refined set* contains the good quality complexes (with resolution ≤ 2.5 Å) which has only standard amino acid residues in the protein and common organic elements (i.e., C, N, O, P, S, F, Cl, Br, I, and H) in the ligand; the *core set* contains three selected complexes with the highest, medium, and lowest binding affinity from each of the protein clusters generated from the refined set complexes. Conventionally, the refined and core sets are used to train and test protein–ligand docking methods. Here, we used the core set of the PDBbind v.2012 for evaluating the performance of the newly implemented PSO algorithm in Vina. In this version, it contains 201 complexes in 67 clusters of protein structures with binding constants spanning nearly 10 orders of magnitude.

In virtual screening experiments, docking algorithms are used to identify potential active ligands from large compound databases. To assess how PSOVina performs in

Table 1. The four selected targets from the DUD data set.

Protein target	Abbr.	PDB	Number of active ligands	Number of decoys	Number of rotatable bonds
Enoyl ACP reductase	inha	1p44	86	3266	3
HIV reverse transcriptase	hivrt	1rt1	43	1519	9
Phosphodiesterase 5	pde5	1xp0	88	1978	12
Angiotensin-converting enzyme	ace	1o86	49	1797	18

virtual screening, we tested both Vina and PSOVina on a subset of active ligands and decoys for four targets selected from the directory of useful decoys database (DUD release 2).[15] It includes the enoyl ACP reductase (inha), HIV reverse transcriptase (hivrt), phosphodiesterase 5 (pde5), angiotensin-converting enzyme (ace). These were selected as representatives of targets accommodating ligands with different rotatable bonds and hence with different flexibilities. A summary of the DUD target data sets used in this study is listed in Table 1.

2.2. *PSO algorithm*

PSO is a nature-inspired metaheuristic algorithm based on the swarm intelligence of birds.[16] It simulates their social behavior of searching for food in an area: Birds do not know the exact location of food; to increase the chance of success, they communicate with one another of its knowledge of the food location. With this information, each of the birds adjusts its flying direction to move closer to the region where the chance of finding food is higher.

In the PSO algorithm, birds are called *particles*. Starting from an initial guess, the position x and the velocity v of a particle i is updated at the time step t as follows:

$$v_i^{t+1} = \omega v_i^t + \alpha\epsilon_1(x_i^* - x_i^t) + \beta\epsilon_2(g^* - x_i^t), \tag{1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{2}$$

where x_i^* and g^* are the best known solution of itself (so-called *personal best* or *pbest*) and the globally best known solution among all particles up to iteration t (so-called *global best* or *gbest*) respectively. The parameters α and β are the learning constants which are to be specified by users, with suggested values of ≈ 2.[5] The parameters ϵ_1 and ϵ_2 are vectors of random numbers between 0 and 1 at each entry. Therefore, in each step, based on the distances of the position of the current particle from the *personal best* and the *global best*, a new position in the subspace defined by the two distances will be determined for exploitation. The parameter ω is the inertia weight. A large value of ω increases the volume of the subspace and hence encourages global search, and vice versa. This value can be a constant or updated iteratively to control the trade-off between global exploration and local exploitation during the optimization procedure.

The update of particles' positions and velocities are done iteratively with the goal of optimizing an objective function which is also used to evaluate the fitness

of each move. The pseudo code of the canonical PSO algorithm in Fig. 1 explains how it works:

```
Define the objective function f(x)
Define the number of particles N
Initialize x_i^0 and v_i^0 of each particle
Setting t = 0
repeat
  for i = 1 .. N do
    evaluate f(x_i^t)
    if fitness value is better than x_i^* then
        x_i^* ← x_i^t
    if fitness value is better than g^* then
        g^* ← x_i^t
    update velocity and position according to Eqs. (1) and (2)
  end for
  t = t + 1
until stopping criterion is met
```

Fig. 1. Pseudo code of the canonical PSO algorithm.

There exist many variants of the canonical PSO algorithm. For example, Clerc *et al.*[17] introduced a constriction coefficient in velocity update called the constricted PSO (CPSO) to prevent the explosion of system when particle velocities increase without control. Yang proposed the accelerated PSO (APSO) to improve the convergence behavior of the algorithm by eliminating the dependence of *personal best* in position update.[18] Recently, a variant of CPSO called PSO-random sampling in variable neighborhoods (PSO-RSVN) was introduced by Nápoles *et al.*[19] to deal with the problem of premature convergence in local optima by using random samples for dispersing the swarm. Furthermore, some studies researched on the effects of local neighborhood on search effectiveness and found that certain communication topologies such as Ring and Square improved search performances.[20,21]

2.3. *The hybrid PSO and BFGS algorithm in AutoDock Vina*

In this study, a hybrid PSO and BFGS algorithm was implemented within the framework of AutoDock Vina[13] — one of the most popular open-source programs for protein–ligand docking — to improve the time efficiency in searching for the optimal ligand conformation in the binding pocket.

In Vina, the conformation of a ligand is represented by its position (x, y, z) in Cartesian coordinates, orientation (q_0, q_1, q_2, q_3) in quaterion, and the torsional angles of rotatable bonds along the ligand $(r_0, r_1, \ldots, r_{\tau-1})$, where τ is the total number of ligand rotatable bonds. The fitness of the conformation is evaluated by a knowledge-based scoring function which is a weighted sum of interaction terms including steric interactions, hydrophobic interactions, and hydrogen bond interactions. Furthermore,

the docking simulation is performed using a Monte Carlo approach: Starting from an initial conformation, a candidate bound-conformation of the ligand is generated by randomly mutating its position, orientation, or selected torsion angle. The conformation then undergoes a local optimization where the optimized structure is assessed by Vina's scoring function. The local optimization method employed by Vina is an iterative quasi-Newton method called BFGS. It uses the gradients of the objective function to determine the direction to the local optimum. Instead of directly calculating the second-order derivatives which is often very expensive, in BFGS, they are approximated using rank-one updates specified by gradient evaluations.[22] Finally, the optimized structure will be checked for acceptance according to the Metropolis criterion and the accepted structure will become the starting structure in the next iteration. If the accepted structure scores better than the current best solution, its conformation will be further optimized and then saved as the current best solution. The search procedure is repeated until the maximum number of step is reached.

Our implementation, here called PSOVina, is a hybrid method of the PSO algorithm and the Vina's BFGS local search algorithm for finding the optimal ligand conformation. As a flexible ligand method, all conformational degrees of freedom, namely, the position, the orientation, and the torsional angle of each rotatable bond together form a *solution vector* — a candidate ligand conformation. The search problem is now to find the solution vector which gives the lowest score by the Vina's scoring function.

Figure 2 shows the general flow of the PSOVina algorithm. At the start of the PSO docking, each of the N particles is initialized with a different solution vector randomly distributed over the search space with random velocity. In each swarm generation, a ligand transformation is randomly selected. Three types of transformation are allowed: *position* — the whole-body translation of the ligand resulting in a change in the binding position, *orientation* — the whole-body rotation of the ligand resulting in a change of the ligand orientation, or *torsion* — rotation of a selected ligand torsional angle resulting in some local moves of the ligand. Then, for each single particle, it first undergoes the BFGS local optimization and the fitness value of the optimized solution vector is computed. If this solution scores better than its *pbest*, then its *pbest* is updated. Similarly, if it scores better than the population *gbest*, then the *gbest* is updated. Afterwards, the updated *pbest* and *gbest* are used to produce the next generation of the swarm by computing the new velocity and the new position pertaining to the selected transformation type for each particle. During the whole-body translation of a particle in the PSO move, if the particle goes out of the binding pocket, instead of putting it back at the border of the search space, it is reinitialized to a random position within the binding pocket. This so-called *regeneration principle*[10] helps to maintain the diversity of the swarm population. The final *gbest* obtained at this iteration will be further optimized using the BFGS local search algorithm and its score is saved. Based on our tests, applying local search on each particle in a generation helps to maintain the prediction accuracy of the docking

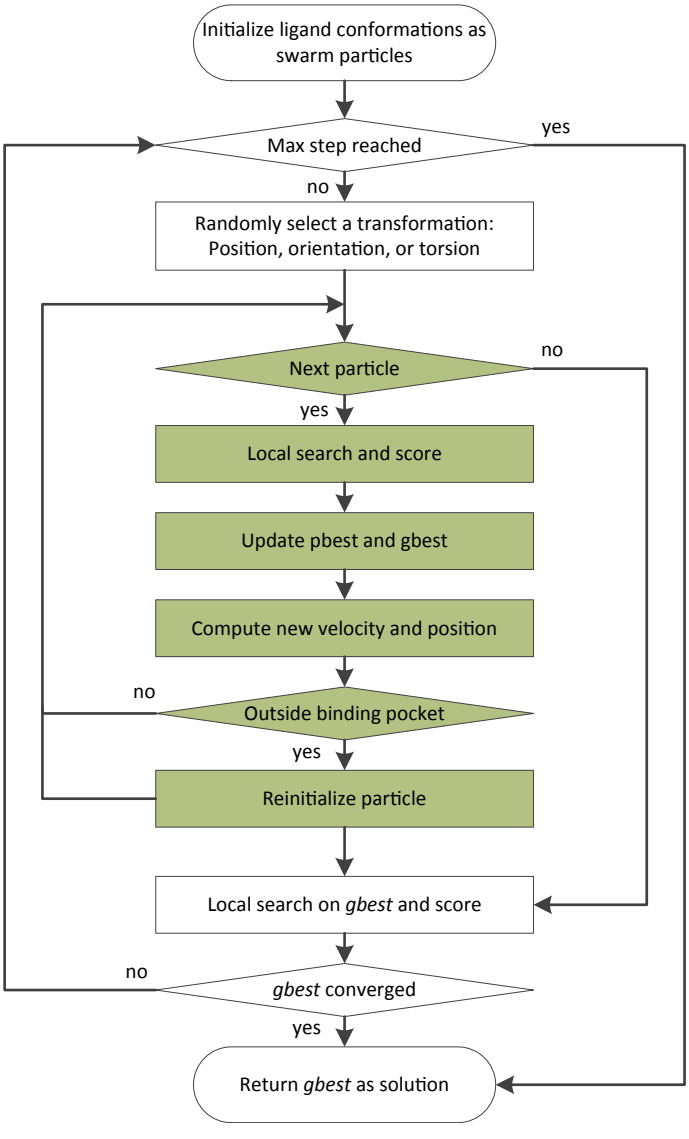


Fig. 2. Flowchart of the PSOVina docking algorithm. The steps in green are PSO moves performed on all particles in the same generation on the selected transformation (the ligand position, the ligand orientation, or a torsional angle of ligand rotatable bond). The BFGS algorithm originally implemented in Vina is used for the local search.

algorithm as Vina. Although the number of local search conducted in each iteration is increased in PSOVina (N number of particle per iteration versus one sample in Monte Carlo per iteration), because of the fast convergence property of the PSO method, the number of evaluations needed to reach the optimum solution is actually reduced.

Finally, the swarm search procedure is repeated until it shows convergence or reaches the maximum number of steps. The system is considered as converged when the fitness value of the *gbest* does not have a significant change within certain number of generation, i.e. the change in the energy value of the *gbest* within C consecutive iterations is $< 10^{-4}$. Finally, the program returns the last *gbest* as the optimal solution.

Based on the Vina framework, the PSO algorithm was implemented using C++, adopting Vina's scoring function as well as its numerous functions for performing ligand transformations, conducting local search, saving coordinates, etc. Source codes and Linux executables of PSOVina are freely available to non-commercial users at http://cbbio.cis.umac.mo.

2.4. *Docking simulations*

As for the docking simulation, a Vina docking run involves four key steps: (1) Prepare the protein and convert it into PDBQT format; (2) prepare the ligand and convert it into PDBQT format; (3) define the binding pocket region by giving the center and the dimensions of the region; (4) perform docking.

The AutoDock PDBQT file is an augmented PDB file which contains not only the atomic coordinates of the atoms but also partial charges and atom types from the AutoDock 4 force field. The protein and ligand PDBQT files were generated using the python script prepare_protein4.py and prepare_ligand4.py provided by AutoDock Tools. Missing hydrogens were added but non-polar hydrogens were merged to the carbon to which it was bonded, so a united-atom model was adopted. Since AutoDock does not consider water molecules during docking, all water molecules were removed. As for the definition of the binding pocket, it was calculated directly from the PDB file of pocket atoms included in the PDBbind data set. The center of the pocket was calculated as the geometrical center of all the pocket atoms and the X-, Y-, Z-dimension of the binding pocket were defined as the largest distance between any two atoms in the respective dimension.

A script was implemented in-house to streamline the aforementioned docking steps for testing of the algorithm with the PDBbind data set. Since both Vina's Monte Carlo and PSOVina's metaheuristic algorithms involve some randomness, for each complex the docking was repeated 30 times and the lowest-energy conformation among all runs is selected as the best binding conformation. The predictive performance of the algorithm was measured by the root mean square deviation (RMSD) of the lowest-energy structures with respect to experiments. In addition, the predicted binding affinity will also be compared to the experimental binding affinity (G), which is calculated by

$$G = -\log_{10} K, \tag{3}$$

where the binding constant K is either the inhibitory constant K_i or the disassociation constant K_d.

In our tests, all docking simulations were run using the option "--cpu 8 --exhaustiveness 8" in a PC with Intel i7 2.8 GHz quad-core processor and 4 GB memory on the Ubuntu 11.04 operating system.

For PSOVina, the number of particle N, parameters for the velocity update w, α, β, and the criteria to be considered as convergence have to be decided. To this end, we systematically tested different combinations of parameter values. In our tests, we observed that using more particles will give slightly more accurate predictions but it also increases the run time. The magnitude of change crucially depends on the number of CPU available in the machine. In our case, when N is larger than 8, the docking time significantly increases, therefore the maximum number of particles should be bounded by this number. On the other hand, we also observed that when the number of rotatable bonds of ligand is increased over 10, employing more particles do not help in boosting prediction accuracy. This is probably due to the deficiency of Vina's scoring function in evaluating highly flexible ligands. In addition, larger values of w, α and β which result in a larger step size will allow more thorough exploration of the search region but it will also take more time to reach convergence. As prediction accuracy is primarily determined by the scoring function, these parameters can be adjusted to reduce the docking time the most without compromising the prediction accuracy. Table 2 summarized the optimal parameter settings found in our tests.

2.5. *Virtual screening experiments*

In virtual screening experiments, the docking procedure mentioned in the previous section is executed for a set of compounds (active ligands or decoys) against the target protein. In the same way, for each protein-compound pair, the overall lowest energy conformation among 10 docking runs is considered as the best binding conformation of the compound. In virtual screening, the primary goal is to select few candidate drug compounds out of the large compound databases. Therefore, a rapid docking algorithm which is able to distinguish active compounds from decoys is desirable. A metric to assess the overall quality of a docking algorithm in virtual screening is the area under the ROC curve (AUC). Since the ROC curve summarizes the relationship between the true positive rate and the false positive rate at different threshold settings, the AUC is a single and objective measure of how the docking

Table 2. Parameter setting used in the PSOVina algorithm.

PSOVina parameter	Setting
Number of particles, N	8
Inertia weight, ω	0.36
Cognitive weight, α	0.99
Social weight, β	0.99
Convergence criteria, C	350

algorithm ranks an active ligand compared to a randomly chosen decoy. An ideal algorithm that ranks all actives higher than all decoys gives an AUC value of 1; an algorithm which produces a ranking close to random would give a value around 0.5.

In our tests, the same parameter settings for docking simulations were used in all virtual screening experiments. The computations were performed in a CentOS 6.2 cluster with 32-core of Intel Xeon 2.60 GHz and 198 GB memory per node.

3. Result

To assess the docking performance of PSOVina in terms of accuracy and time efficiency in comparison to AutoDock Vina, we performed 30 docking simulations on each of the protein–ligand complexes on the PDBbind data set. Docking accuracy was measured by the average RMSD of the predicted ligand conformations and the correlation coefficient of predicted binding affinities to experimental binding affinities, whereas time efficiency is quantified by the number of iterations to reach convergence and the total run time.

Figure 3 shows the evolution of the average RMSD as the number of docking runs increases. Interestingly, abrupt changes in average RMSD were observed in both Vina and PSOVina within the first 2–3 runs, and then they were followed by small variations and final convergence.

The "converged" predictions after 30 runs are the overall lowest-energy ligand conformations, however, these lowest-energy conformations do not correspond to the lowest RMSD conformations. This is probably due to the imperfect scoring function in Autodock Vina which scores some non-native conformations better than the native ones. Based on the converged predictions, the docking simulation results can be summarized as shown in Table 3.

On average, PSOVina predicts with an RMSD of 0.288813 nm, about 9% smaller as compared to the 0.31880 nm of Vina even though their correlation

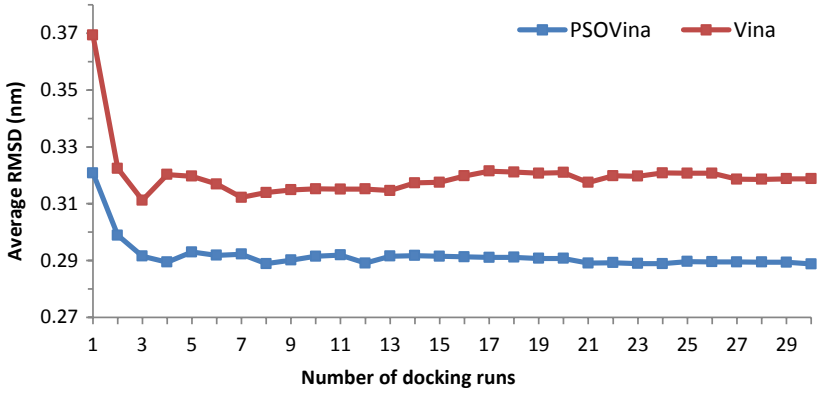


Fig. 3. Average RMSD versus number of docking runs on the PDBbind data set.

Table 3. Docking performance of PSOVina and Vina on the PDBbind data set from 30 docking runs.

	PSOVina		Vina	
	Value	SD[f]	Value	SD[f]
RMSD[a]	0.288813	0. 354361	0.31880	0.37926
r[b]	0.51538	—	0.53428	—
Success rate[c]	63.18%	—	59.7%	—
Iteration[d]	1035.77	938.74	22709.8	7003.7
Time[e]	17.17	3.6	34.7	0.4

[a]Average RMSD (in nm) of predicted ligand conformations to experiments.
[b]The correlation coefficient of predicted and experimental binding affinities.
[c]Success rate is the ratio of successful predictions with RMSD < 0.2 nm to all predictions.
[d]Average number of iterations to reach convergence in one docking run.
[e]Average elapsed time (in second) per docking instance.
[f]Standard deviation.

coefficients to experiments are very similar. Indeed, the binding affinities of predicted conformations by PSOVina and Vina are highly correlated as indicated by a correlation coefficient of 0.994154 (see Fig. 4). In addition, their success rates (predictions with RMSD < 0.2 nm) are 63.18% and 59.7% for PSOVina and Vina respectively, some improvements in accuracy can be seen in PSOVina.

In terms of time efficiency, the swarm-based algorithm with local search and stopping criteria was able to locate the solution and converged much faster than MC-based method which terminates only until the maximum number of steps has

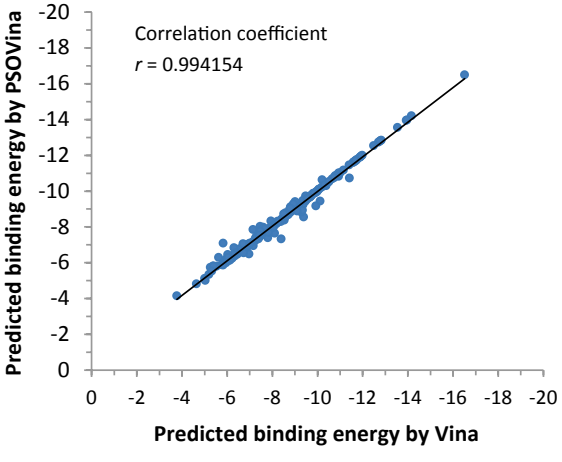
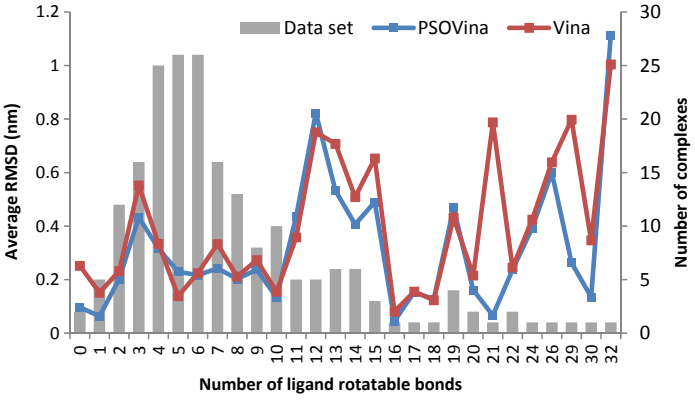


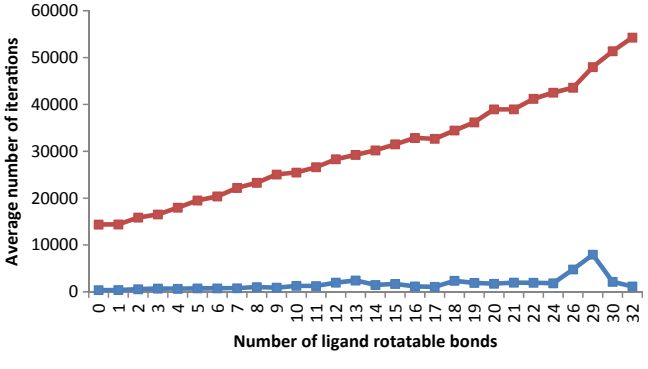
Fig. 4. Comparison of the predicted binding energy by PSOVina and Vina on the PDBbind data set. The correlation coefficient is 0.994154 which indicates that the two docking results are highly correlated.

reached. In our docking simulations, PSOVina attains a remarkable execution time reduction of 51% using only 5% of the iterations required in Vina to complete the docking.

To understand how PSOVina behaves in the variation of different ligand torsion sizes, we calculated the average RMSD of ligand predictions and the average number of iterations with respect to the number of ligand rotatable bonds and the results are presented in Fig. 5. It is noteworthy that the data set has an uneven distribution of samples over the ligand torsion sizes and most ligands have only 1 to 14 rotatable bonds. So, results for large ligand sizes may not be statistically significant. Note also that data for some ligand sizes were not available, such as 23, 25, 27, 28, 31 ligand rotatable bonds, so these data points are omitted in the figure. Nevertheless, we observe that predictions by PSOVina are generally better in RMSD than those by



(a)



(b)

Fig. 5. Comparison of (a) average RMSD and (b) average number of iterations against the number of ligand rotatable bonds. The count of complexes for each case is also shown and the available complexes with many torsional degrees of freedom (> 15) is few. Overall, it shows that PSOVina predicts comparably or better than Vina with large reduction in docking time.

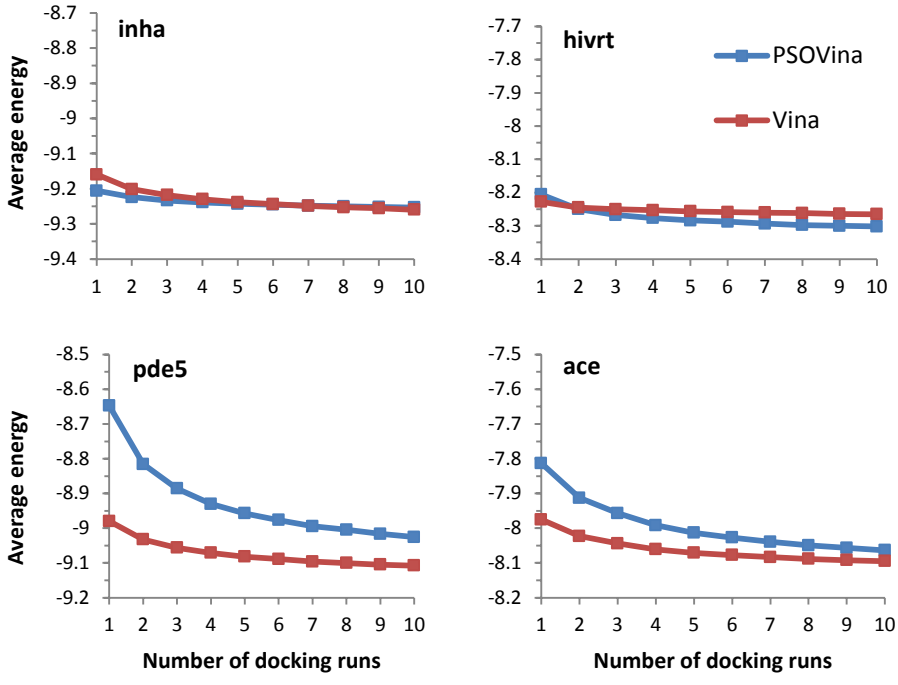


Fig. 6. Average energy (in kcal/mol) of the lowest-energy conformations versus number of docking runs.

Vina. In agreement to Table 3, the number of iterations to reach convergence for each ligand size was greatly reduced in PSOVina as compared to Vina.

To assess the performance of PSOVina on virtual screening, we conducted docking on the DUD data set for four targets. Our selections include proteins with crystallographic ligands of 3, 9, 12, and 18 rotatable bonds, as representatives of rigid, moderate and highly flexible ligands. Each docking is repeated 10 times and the lowest-energy conformation among all runs is returned as the best solution. In order to find out this "optimal" choice of the number of docking runs (weighing trade-off between cost and accuracy), we observed the changes in the average energy computed from the lowest-energy conformations up to that docking run. As shown in Fig. 6, except pde5 with PSOVina, the other 3 targets (hivrt, inha, and ace) show converged average energy within 10 docking runs for both PSOVina and Vina. In addition, the average energies of PSOVina and Vina after 10 docking runs are within 0.1 kcal/mol of each other.

We calculated the AUC of ROC metric on the virtual screening results to assess the performance of PSOVina and Vina and their values are presented in Table 4. The AUC values for PSOVina and Vina for each target are very close, which agrees with our observation in the previous docking studies that PSOVina maintains comparable prediction accuracy to Vina. Nevertheless, the run time cost for a complete screening

Table 4. Virtual screening performance assessed by the AUC of ROC on the DUD data set.

Target	PSOVina	Vina
inha	0.54521	0. 54658
hivrt	0.61659	0.62080
pde5	0.68386	0.66261
ace	0.41246	0.41729
Average	0.56453	0.56182

is reduced by 60% on average (see Table 5) making PSOVina a better choice of the docking tool in virtual screening experiments than Vina.

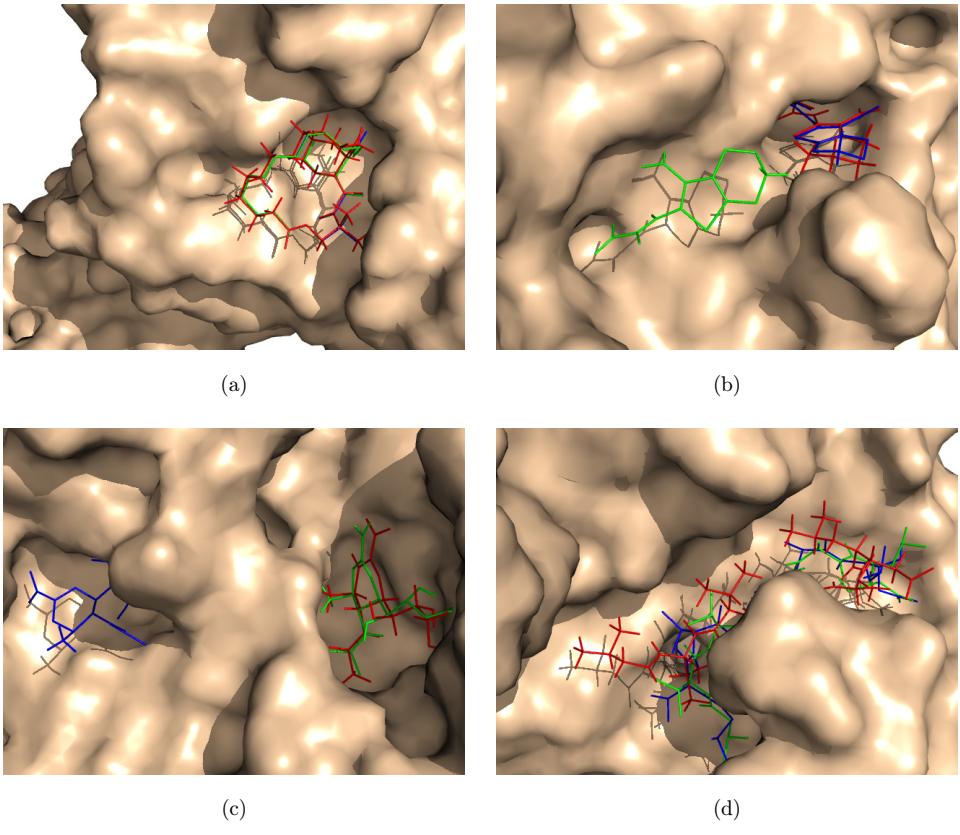
To further our understanding on PSO algorithm in application on protein–ligand docking, we selected four example complexes with different torsion sizes and visualized the predicted conformations by PSOVina and Vina against the experiments in Fig. 7. Measurements of docking accuracy of these examples are presented in Table 6.

The first example, 1FKI, has no torsional degrees of freedom and thus the search problem is to locate the correct position and orientation of the ligand. In this case, the two methods give almost identical predictions which perfectly match with the known experimental structure with exactly the same score. The second example, 1C88 is a ligand with small number of rotatable bonds and it has shown a case where Vina fails to locate the correct binding position. From the snapshot, we see that the pocket defined in the original PDBbind data is very wide that includes also the surface far from the binding groove. Nevertheless, PSOVina is able to locate the position of the binding groove with successful predicted ligand conformation. This example reveals that combination of both global and local search in PSO algorithm is the key to probe promising regions when the search space is large. The third example 2QWD is a ligand with moderate number of rotatable bonds. In this case, PSOVina finds a ligand conformation which is very close to the binding affinity of the experimental affinity but at the border region of the binding pocket. We visually inspected the docking process and found that PSOVina explored both regions thoroughly. The binding scores of the lowest-energy conformations at the end

Table 5. Average run time (in seconds) of PSOVina and Vina on the DUD data set.

		PSOVina		Vina	
Target	Total # of a full VS docking	Per compound	Full VS[a]	Per compound	Full VS[a]
inha	20,660	5.2	173,962	10.4	348,612
hivrt	15,620	4.8	75,506	10.5	164,304
pde5	33,520	8.6	178,149	25.4	525,479
ace	18,460	5.6	103,971	13.5	249,260
Average	22,065	6.0	132,897	15.0	321,914

[a]Screening of a full compound set including all active ligands and decoys.



(a) (b)

(c) (d)

Fig. 7. Visualization of four selected docking results from the PDBbind data set. Ligands are represented in sticks and the binding pockets are displayed with surface representation. Color code: red for experiment, blue for PSOVina, and green for Vina.

of each docking run are very close; the jumping between the two regions requires overcoming a large energy barrier but this was not a problem for PSOVina. While the near-native conformation was sampled during the docking, we expected that an improved scoring function might be the key to select the active conformation from the decoys. The last example 4ER2 with highly flexible ligand is shown. In this case, although both predictions locate the correct binding position, neither of them predicts the correct binding conformation. Looking at the structures closely, we observe that the ligand orientations are entirely flipped over and therefore produce large RMSDs. While the number of rotatable bonds is large, the number of particles in PSOVina is probably too small for a complete exploration of the search space. Together with the stringent convergence criteria, the search converges too quickly to a local optimum. We anticipated that further local search starting from this solution would not be able to flip back the ligand. This discloses a disadvantage of PSOVina in predicting ligand with large torsion sizes.

Table 6. Details of the predicted conformations of the four examples in Fig. 7.

PDB ID	1FKI	1C88	2QWD	4ER2
Number of rotatable bonds	0	3	9	24
Experimental binding affinity	7.00	5.29	4.85	9.3
Predicted binding affinity				
PSOVina	8.512	5.161	4.771	6.092
Vina	8.512	4.200	4.615	5.841
RMSD to experiment (nm)				
PSOVina	0.023052	0.092921	0.67861	0.392646
Vina	0.023056	1.110490	0.09927	0.46395

4. Conclusion

Protein–ligand docking is an essential step in modern drug discovery process. To enhance the docking performance in terms of time efficiency, we proposed a hybrid conformational search method based on PSO algorithm. The method was implemented into the popular docking program AutoDock Vina, here called PSOVina. In this implementation, we combined the advantages of search diversity and quick convergence of PSO with the efficient BFGS local search method already adopted in Vina. Our docking and virtual screening experiments with two large and diverse data sets showed that PSOVina has a remarkable execution time reduction of 51–60% without compromising the docking accuracies of the original Vina method in terms of RMSD, binding affinity, success rate, and AUC of ROC. Our detailed analysis of docking performance with respect to the ligand sizes reveals room for further improvement of the algorithm. Future work will be focused on improving predictions in flexible ligands, fine-tuning the parameter settings, applying different nature-inspired algorithms and assessing the method with other benchmark data sets to facilitate comparison with other state-of-the-art docking methods.

Acknowledgments

This work was supported by the research grant of University of Macau (grant number SRG022-FST13-SWI and MYRG2014-00104-FST). The authors would like to thank the Information and Communication Technology Office and the laboratory in the Department of Computer and Information Science of University of Macau for their support of computing facilities.

References

1. Kitchen DB, Decornez H, Furr JR, Bajorath J, Docking and scoring in virtual screening for drug discovery: Methods and applications, *Nat Rev Drug Discov* **3**:935–949, 2004.
2. Sousa SF, Fernandes PA, Ramos MJ, Protein-ligand docking: Current status and future challenges, *Proteins Struct Funct Bioinforma* **65**:15–26, 2006.
3. Huang, S-Y, Zou X, Advances and challenges in protein–ligand docking, *Int J Mol Sci* **11**:3016–3034, 2010.

4. Miller D, Dill K, Ligand binding to proteins: The binding landscape model, *Protein Sci* **6**:2166–2179, 1997.
5. Yang, X-S, *Nature-Inspired Optimization Algorithms. Nature-Inspired Optim. Algorithms* 99–110, 2014, doi: 10.1016/B978-0-12-416743-8.00007-5.
6. Eberhart R, Shi Y, Particle swarm optimization: Developments, applications and resources, *..., 2001. Proc. 2001 Congr. ...*, pp. 81–86, 2001, Available <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=934374>.
7. Dong Y, Tang J, Xu B, Wang D, An application of swarm optimization to nonlinear programming, *Comput Math Appl* **49**:1655–1668, 2005.
8. Chen HM, Liu BF, Huang HL, Hwang SF, Ho SY, SODOCK: Swarm optimization for highly flexible protein–ligand docking, *J Comput Chem* **28**:612–623, 2007.
9. Korb O, St T, Exner TE, PLANTS: Application of ant colony optimization to structure-based drug design, *Ant Colony Optimization and Swarm Intelligence*, Lecture Notes in Computer Science, Vol. 4150, pp. 247–258, 2006.
10. Namasivayam V, Günther R, PSO@ AUTODOCK: A fast flexible molecular docking program based on swarm intelligence, *Chem Biol Drug Des* **70**:475–484, 2007.
11. Meier R, Pippel M, Brandt F, Sippl W, Baldauf C, ParaDockS: A framework for molecular docking with population-based metaheuristics, *J Chem Inf Model* **50**:879–889, 2010.
12. Liu Y *et al.*, FIPSDock: A new molecular docking technique driven by fully informed swarm optimization algorithm, *J Comput Chem* **34**:67–75, 2013.
13. Trott O, Olson A, AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, *J Comput Chem* **31**:455–461, 2010.
14. Wang R, Fang X, Lu Y, Wang S, The PDBbind database: Collection of binding affinities for protein–ligand complexes with known three-dimensional structures, *J Med Chem* **47**:2977–2980, 2004.
15. Huang N, Shoichet BK, Irwin JJ, Francisco S, Benchmarking sets for molecular docking, *J Med Chem* **49**:6789–6801, 2006.
16. Kennedy J, Eberhart R, Particle swarm optimization, *IEEE Int Conf Neural* **4**:1942–1948, 1995.
17. Clerc M, Kennedy J, The particle swarm — explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans Evol Comput* **6**:58–73, 2002.
18. Yang X, *Nature-Inspired Metaheuristic Algorithms*, 2nd edn. (Luniver Press, UK).
19. Nápoles G, Grau I, Bello R, Constricted particle swarm optimization based algorithm for global optimization, *Polibits* **46**:5–11, 2012.
20. Kennedy J, Mendes R, Population structure and particle swarm performance, Vol. 2, *Proc 2002 Congr Evol Comput CEC'02 (Cat. No.02TH8600)*, pp. 1671–1676, 2002.
21. Mendes R, Kennedy J, Neves J, The fully informed particle swarm: Simpler, maybe better, *IEEE Trans Evol Comput* **8**:204–210, 2004.
22. Shanno D, Conditioning of quasi-Newton methods for function minimization, *Math Comput* **24**:647–656, 1970.

Marcus C. K. Ng received the B.Sc. degree in Software Engineering from University of Macau, Macau S.A.R, China, in 2014. He is now working in University of Macau.

His research focuses on protein–ligand docking and optimization algorithms in bioinformatics.

Simon Fong graduated from La Trobe University, Australia, with a 1st Class Honours B.Eng. Computer Systems degree and a Ph.D. Computer Science degree in 1993 and 1998 respectively. Simon is now working as an Associate Professor at the Computer and Information Science Department of the University of Macau. He is also one of the founding members of the Data Analytics and Collaborative Computing Research Group in the Faculty of Science and Technology. Prior to joining the University of Macau, he worked as an Assistant Professor in the School of Computer Engineering, Nanyang Technological University, Singapore.

Shirley Weng In Siu completed her Master's degree at the Max Planck Research School for Computer Science in 2005. Afterwards, she joined the Theoretical and Computational Membrane Biology group at Saarland University where she received her Ph.D. degree (Dr. rer. nat.) in 2010. Then, she joined the Computational Biology group at University of Erlangen for postdoctoral training. Currently, she is an Assistant Professor in the Department of Computer and Information Science at University of Macau. Her research interests include molecular dynamics simulations, protein structure prediction, machine learning and its application in drug discovery and bioinformatics.