



# Augmented ant colony algorithm for virtual drug discovery

Luca Donati<sup>1,2</sup> · Konstantin Fackeldey<sup>1,3</sup> · Marcus Weber<sup>1</sup>

Received: 15 September 2023 / Accepted: 31 October 2023 / Published online: 14 November 2023  
© The Author(s) 2023

## Abstract

Docking is a fundamental problem in computational biology and drug discovery that seeks to predict a ligand's binding mode and affinity to a target protein. However, the large search space size and the complexity of the underlying physical interactions make docking a challenging task. Here, we review a docking method, based on the ant colony optimization algorithm, that ranks a set of candidate ligands by solving a minimization problem for each ligand individually. In addition, we propose an augmented version that takes into account all energy functions collectively, allowing only one minimization problem to be solved. The results show that our modification outperforms in accuracy and efficiency.

**Keywords** Docking problem · Ligand · Receptor · Global optimization · Ant colony algorithm

## 1 Introduction

A major challenge in computational chemistry and drug discovery is the search for new molecules that bind with specific receptors in order to activate beneficial biochemical mechanisms minimising side effects [1–4]. However, the search for new molecules, commonly referred to as virtual screening, is limited by the huge number of possible candidates that exist, as well as by a limited knowledge of the receptors of interest.

---

✉ Luca Donati  
donati@zib.de

Konstantin Fackeldey  
fackeldey@math.tu-berlin.de

Marcus Weber  
weber@zib.de

<sup>1</sup> Zuse Institute Berlin, Takustr. 7, Berlin D-14195, Germany

<sup>2</sup> Freie Universität Berlin, Takustr. 3, Berlin D-14195, Germany

<sup>3</sup> Technische Universität Berlin, Strasse des 17. Juni 135, Berlin 10623, Germany

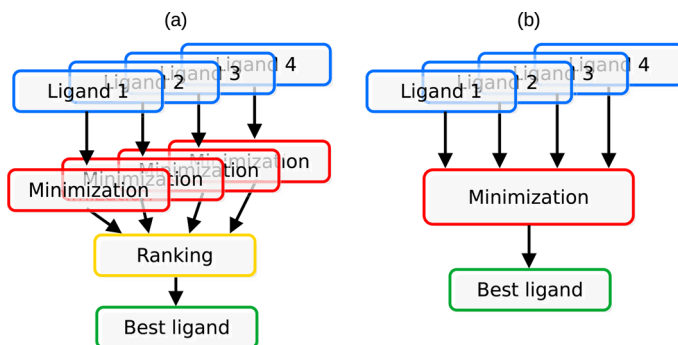
Various strategies have been developed over the years. Receptor-ligand docking methods identify the optimal binding orientation and position between ligand and receptor, as well as rank the ligands of a database according to their binding affinity [5–7]. However, docking requires complete knowledge of the receptor's molecular structure. Alternative methods are based on molecular similarity, i.e. on the search for molecules with a molecular shape similar to that of a molecule already known to bind with a receptor [8–12]. The principle behind this strategy is that the chemical and physical properties of a ligand-receptor system are contained in the electronic distribution, so ligands with a similar electronic density cloud shape have approximately the same probability of binding to a receptor [13–17]. The advantage of this approach is that complete knowledge of the receptor is not required. Both strategies attempt to solve an optimization problem, however, in this work, we focus on docking, whose objective is to minimize an energy function that describes the strength of the binding between the ligand and the receptor [18].

In docking, finding the global minimum of the energy function is hampered by the presence of local minima and energy barriers that prevent the exploration of the function (trapping problem). A second obstacle is the high dimensionality of the energy functions due to the degrees of freedom of the system, and thus the vast size of the space in which the solution resides.

In a nutshell, finding the global minimum is a hard computational task that requires advanced algorithms and techniques to optimize computational resources without losing accuracy. Ongoing research in this field has proposed a vast spectrum of methods and strategies to improve the accuracy and efficiency of docking algorithms. For instance, the programs GOLD [19] and AutoDock [20] use genetic algorithms, ICM [21] and QXP [22] use Monte Carlo minimization, PRO\_LEADS [23] implements simulated annealing, evolutionary programming, and tabu search, and FlexX [24] and DOCK [25] use fragments algorithms.

In this article, we review the stochastic optimization method PLANTS (Protein-Ligand ANT System) [26] implemented in the software platform VirtualFlow [27, 28] for virtual screening and ligand preparation, and propose an augmented version of PLANTS, which we will refer to as PLANTS+. PLANTS is an Ant Colony Optimization algorithm [29], i.e. an algorithm inspired by the behavior of ants that search for short paths between the anthill and food sources, guided by pheromone trails. This kind of algorithm was first designed for discrete optimization problems that can be represented by a graph made of nodes and edges. A typical example is the traveling salesman problem, where the shortest path connecting two network nodes is sought. The application to continuous function minimization problems is possible by discretizing the domain of the function in disjoint subsets that are analogous to the network nodes in the traveling salesman problem. The artificial ants, however, do not search for the shortest path between two subsets, but rather search the subset of the domain that contains the global minimum, and then deposit the pheromone in areas with low energy values.

Given a set of ligand candidates for a specific receptor, PLANTS is used to determine the global minimum of each binding energy function. The ligands are then ranked according to their energy values when they are in their optimal position and orientation, and the best ligand for the receptor is the one that takes the lowest energy value.



**Fig. 1** Overview of the workflow of the algorithm PLANTS **a** and PLANTS+ **b**

PLANTS+ extends the search for the global minimum from the space of degrees of freedom to the space of binding energy functions by analyzing all energy functions simultaneously, rather than individually. Indeed, the artificial ants deposit the pheromone in the regions of the space of degrees of freedom where the energy value is lowest among all energy functions, and the algorithm returns not only the coordinates of the global minimum but also the energy function that takes the lowest value from those coordinates. Then, the main difference between PLANTS and PLANTS+ is that the former solves as many minimization problems as the number of ligands under consideration, whereas the latter solves a single minimization problem, as illustrated by the workflows in Fig. 1.

In order to test the new algorithm, we studied sets of artificial potentials where the ligand and the receptor are assumed to be rigid and only translational and rotational degrees of freedom are taken into account. This is actually a simplification. Indeed, both the receptor and the ligand are flexible, and how the electron density clouds interact determines the chemical and physical properties of the system [30, 31]. However, the scope of this paper is to show how to improve the ligands classification process by optimizing the use of available information, regardless of how the binding free energy is calculated.

The article is outlined as follows. We review PLANTS and describe the new version PLANTS+ in Sect. 2. We present an illustrative example that highlights the differences between the two methods and present the results of a comparative analysis 3 that show under which conditions PLANTS+ is more accurate and more efficient than PLANTS. Section 4 concludes the article with a short summary of our findings as well as an outlook for future research.

## 2 Methods

Consider a ligand-receptor system, where the receptor is kept fixed in the Euclidean space, while the ligand can rotate or translate with respect to the receptor determining 3 rotational degrees of freedom and 3 translational degrees of freedom. Additionally,

there are  $n_t$  torsional degrees of freedom due to molecules' bonds that are not associated with a ring-chain. The system is therefore characterized by a  $(6+n_t)$ -dimensional binding energy surface

$$E(\Psi_1, \Psi_2, \Psi_3, \Psi_{n_1}, \dots, \Psi_{n_t}, r_1, r_2, r_3) : \Omega \subset [-\pi, \pi]^{3+n_t} \times \mathbb{R}_{\geq 0}^3 \rightarrow \mathbb{R}, \quad (1)$$

where  $\Omega$  denotes the space of all possible positions and orientations of the ligand with respect to the receptor,  $\Psi$  denotes the 3 rotational and  $n_t$  torsional degrees of freedom, and  $r$  denotes the 3 translational degrees of freedom. Note that, in order to shorten the notation in eq. 1, from this point onwards we denote the energy function with  $E(\Psi_i, r_j)$  with  $i = 1, 2, \dots, 3 + n_t$  and  $j = 1, 2, 3$ .

Consider now a set of  $N_{\text{ligands}}$  ligands, each with an energy function

$$E_\alpha(\Psi_{\alpha,i}, r_{\alpha,j}), \quad \forall i = 1, 2, \dots, 3 + n_t; \forall j = 1, 2, 3, \quad (2)$$

where the index  $\alpha = 1, 2, \dots, N_{\text{ligands}}$  denotes the  $\alpha$ th ligand of the set, to solve the docking problem means finding the energy function  $E_\alpha$  that takes the lowest value in its global minimum.

## 2.1 PLANTS

In PLANTS, the global minimum of a binding energy function  $E$  is determined using the Ant Colony Optimization algorithm, which was originally developed for discrete problems. Then, for a docking problem, the rotational and torsional degrees of freedom  $\Psi_i$ , and the translational degrees of freedom  $r_j$  need to be discretized  $\forall i = 1, 2, \dots, 3 + n_t$  and  $\forall j = 1, 2, 3$ . Various discretizations are possible and affect the efficiency and accuracy of the algorithm. However, when no a priori knowledge of the system is available, it is recommended that each degree of freedom is discretized into equal intervals. For example, a rotational degree of freedom  $\Psi$  with  $\Psi(0) = -\pi$  and  $\Psi(N_\Psi) = \pi$  can be discretized into  $N_\Psi$  equal disjoint subsets such that

$$[-\pi, \pi] = \bigcup_{l=1}^{N_\Psi} [\Psi(l-1), \Psi(l)], \quad (3)$$

while a translational degree of freedom  $r$  with  $r(0) = 0$  and  $r(N_r) = r_{\text{max}}$  can be discretized into  $N_r$  disjoint subsets such that

$$[0, r_{\text{max}}] = \bigcup_{k=1}^{N_r} [r(k-1), r(k)], \quad (4)$$

where  $r_{\text{max}} \in \mathbb{R}_{\geq 0}$  is the maximum distance between ligand and receptor appropriately chosen. This gives rise to a discretization of the space  $\Omega$  into  $N = N_\Psi^{3+n_t} \times N_r^3$  subsets

$A_1, A_2, \dots, A_N \in \Omega$  with

$$A_m = \prod_{i=1}^{3+n_l} [\Psi_i(l-1), \Psi_i(l)] \times \prod_{j=1}^3 [r_j(k-1), r_j(k)], \quad \forall m = 1, 2, \dots, N, \quad (5)$$

where  $l = 1, 2, \dots, N_\Psi$  and  $k = 1, 2, \dots, N_r$ . The discretization subsets of the space  $\Omega$  of the degrees of freedom are analogs to the network nodes in the traveling salesman problem, and the artificial ants move from one subset to another one looking for the subset  $A^* \in \Omega$  that contains the coordinates  $\Psi_i^*, r_j^*$ , such that the energy function is minimized:

$$E(\Psi_i^*, r_j^*) \leq E(\Psi_i, r_j) \quad \forall \Psi_i, r_j \in \Omega. \quad (6)$$

Consequently, an artificial ant, solution to the docking problem, is a specific combination of degrees of freedom that minimizes the energy function. In a nutshell, the algorithm works by generating certain pheromone vectors  $\tau_{\Psi_i}$  and  $\tau_{r_j}$  for each degree of freedom, which normalized give rise to probability vectors  $p_{\Psi_i}$  and  $p_{r_j}$  that are used by artificial ants to move around the domain of possible solutions.

### 2.1.1 Pheromone update rule

Initially, all pheromone vector entries are equal to one, then the probability vectors approximate uniform distributions and the initial  $N_{\text{ants}}$  solutions  $\Psi_i^a, r_j^a$  are generated with equal probability. Here the index  $a$  denotes one of  $N_{\text{ants}}$  possible solutions. These solutions are then locally minimized:

$$\begin{cases} \Psi_i^a = \operatorname{argmin}_{\Psi_i} \left( E \left( \Psi_i^a, r_j^a \right) \right) \\ r_j^a = \operatorname{argmin}_{r_j} \left( E \left( \Psi_i^a, r_j^a \right) \right) \end{cases}, \quad \forall a = 1, 2, \dots, N_{\text{ants}}, \quad \forall i = 1, 2, \dots, 3 + n_l, \quad \forall j = 1, 2, 3. \quad (7)$$

Local minimization can be achieved by means of several algorithms. PLANTS uses the Nelder-Mead algorithm [32], a simplex algorithm that does not need the calculation of derivatives, but other choices are available. Afterwards, the best artificial ant, i.e. the solution with coordinates  $\Psi_i^*$  and  $r_j^*$  that gives the lowest energy value, is identified and used to update the vectors  $\tau_{\Psi_i}$  and  $\tau_{r_j}$ . This ensures that the global minimum is approached step by step by generating new solutions in accordance with the new probability vectors.

The rule for updating the pheromone vectors  $\tau_{\Psi_i}$  and  $\tau_{r_j}$  is the same as described in [28]. Given the best solutions  $\Psi^*$  and  $r^*$ , the  $l$ th and  $k$ th entries of the pheromone vectors respectively associated with the rotational degree of freedom  $\Psi_i$  and the translational degree of freedom  $r_j$ , are updated as

$$\begin{aligned} \tau_{\Psi_i}(l) &= (1 - \rho)\tau_{\Psi_i}(l) + \mathbf{1}_{\Psi_i}(l) \\ \tau_{r_j}(k) &= (1 - \rho)\tau_{r_j}(k) + \mathbf{1}_{r_j}(k), \end{aligned} \quad (8)$$

where the functions  $\mathbf{1}_{\Psi_i}(l)$  and  $\mathbf{1}_{r_j}(k)$  are given by

$$\begin{aligned} \mathbf{1}_{\Psi_i}(l) &= \begin{cases} 1 & \text{if } \Psi_i^* \in [\Psi_i(l-3), \Psi_i(l+2)] \\ 0 & \text{else} \end{cases}, \\ \mathbf{1}_{r_j}(k) &= \begin{cases} 1 & \text{if } r_j^* \in [r_j(k-2), r_j(k+1)] \\ 0 & \text{else} \end{cases}. \end{aligned} \quad (9)$$

In Eq. 8,  $\rho$  is an input parameter between 0 and 1 called evaporation rate. The function of the evaporation rate is to partially reset the pheromone vector to prevent the algorithm from being trapped in a local minimum. Finally, the corresponding probability vectors are obtained by normalization:

$$\begin{aligned} p_{\Psi_i}(l) &= \frac{\tau_{\Psi_i}(l)}{\sum_{l=1}^{N_{\Psi_i}} \tau_{\Psi_i}(l)}, \\ p_{r_j}(k) &= \frac{\tau_{r_j}(k)}{\sum_{k=1}^{N_r} \tau_{r_j}(k)}. \end{aligned} \quad (10)$$

The procedure runs for  $N_{\text{steps}}$  iterative steps by generating new solutions which converge to the global minimum. The algorithm is finally applied to all of the ligands under consideration, which are ranked based on their global minimum energy value. Depending on the infrastructure, this can be implemented using a for loop or by parallelizing the process. The complete algorithm is described in alg. 1.

---

### Algorithm 1 PLANTS algorithm

---

**Require:**  $N_{\text{ants}}, \rho, N_{\text{steps}}, N_{\Psi}, N_r$

- 1: Initialize the pheromone vectors:  $\tau_{\Psi}[N_{\Psi}], \tau_r[N_r]$
- 2: Initialize the probability vectors:
- 3:  $p_{\Psi} \leftarrow \text{normalize}(\tau_{\Psi})$ ;
- 4:  $p_r \leftarrow \text{normalize}(\tau_r)$
- 5: Initialize the vector Energy[ $N_{\text{ligands}}$ ]
- 6: // The outermost for loop can be parallelized
- 7: **for**  $i = 1$  to  $N_{\text{ligands}}$  **do**
- 8:   **for**  $t = 1$  to  $N_{\text{steps}}$  **do**
- 9:      $\Psi \leftarrow$  Generate  $N_{\text{ants}}$  coordinates  $\Psi$  with probability  $p_{\Psi}$
- 10:      $r \leftarrow$  Generate  $N_{\text{ants}}$  coordinates  $r$  with probability  $p_r$
- 11:     **for**  $a = 1$  to  $N_{\text{ants}}$  **do**
- 12:        $\Psi[a] \leftarrow \text{local\_minimization}(\Psi[a], \text{energy } i\text{th ligand})$
- 13:        $r[a] \leftarrow \text{local\_minimization}(r[a], \text{energy } i\text{th ligand})$
- 14:     **end for**
- 15:      $\Psi^*, r^* \leftarrow \text{get\_best\_solution}(\Psi, r, \text{energy } i\text{th ligand})$
- 16:      $\tau_{\Psi}, \tau_r \leftarrow \text{update\_pheromone}(\tau_{\Psi}, \tau_r, \Psi^*, r^*; \rho)$
- 17:      $p_{\Psi} \leftarrow \text{normalize}(\tau_{\Psi})$ ;
- 18:      $p_r \leftarrow \text{normalize}(\tau_r)$
- 19:     Energy[ $i$ ] = calculate\_energy( $\Psi^*, r^*, i\text{th ligand}$ )
- 20:   **end for**
- 21: **end for**
- 22: **return** ligand with the lowest energy in its global minimum.

---

## 2.2 PLANTS+

The PLANTS algorithm searches for the optimal position and orientation of each candidate ligand by analyzing each energy function individually. The best candidate is finally determined by evaluating the energy value of each ligand in its global minimum. This corresponds to solving  $N_{\text{ligands}}$  optimization problems, i.e. the number of ligands considered, where each optimization problem consists of minimizing an energy function  $E(\Psi_i, r_j)$  with  $6 + n_t$  degrees of freedom.

Alternatively, here we propose to solve a single optimization problem in which the function to be minimized is constructed as the sum of all energy functions defined in eq. 2:

$$\begin{aligned} E(\Psi, r) &= \sum_{\alpha=1}^{N_{\text{ligands}}} E_{\alpha}(\Psi_{\alpha,i}, r_{\alpha,j}) \\ &= E_1(\Psi_{1,i}, r_{1,j}) + E_2(\Psi_{2,i}, r_{2,j}) + \dots \\ &\quad + E_{N_{\text{ligands}}}(\Psi_{N_{\text{ligands}},i}, r_{N_{\text{ligands}},j}). \end{aligned} \quad (11)$$

This function, which we will refer to as the global energy function, has  $(6 + n_t) \times N_{\text{ligands}}$  degrees of freedom, so the new problem consists of minimizing a  $(6 + n_t) \times N_{\text{ligands}}$ -dimensional function.

To solve this problem, we propose an augmented version of the original PLANTS algorithm, which we will refer to as PLANTS+. In practice, PLANTS is modified as follows. The outermost for-loop, over the candidate ligands, is removed, and the simplex algorithm is applied to the global energy function defined in Eq. 11. Then, the pheromone and probability vectors are updated according to the rule defined in Eq. 8. The crucial difference from the original algorithm is that the solutions (artificial ants) are minimized with respect to the global energy function. This entails that the pheromone and probability vectors contain information on all ligands taken together, and not individually as in the original PLANTS. The algorithm is fully illustrated in alg. 2, where we denote in line 15 the best solution as  $\Psi_L^*$  and  $r_L^*$  in Alg. 2 since it is calculated with respect to all ligands.

The limitation of this approach is the high number of dimensions of the energy function defined in Eq. 11, which grows with the number of ligands examined. It is in fact well known that the simplex algorithm loses accuracy with the dimensionality of the function to be minimized [33]. On the other hand, because the ligands' energy functions (Eq. 1) are independent of each other, there is no need to apply the simplex algorithm to the global energy function (Eq. 11). Instead, it is more convenient to apply the local minimization algorithm to the individual energy functions either within a for loop, as in Alg. 2, or by paralleling the calculation.

**Algorithm 2** PLANTS+ algorithm

---

**Require:**  $N_{\text{ants}}, \rho, N_{\text{steps}}, N_{\Psi}, N_r$

- 1: Initialize the pheromone vectors:  $\tau_{\Psi}[N_{\Psi}], \tau_r[N_r]$
- 2: Initialize the probability vectors:
- 3:  $p_{\Psi} \leftarrow \text{normalize}(\tau_{\Psi})$ ;
- 4:  $p_r \leftarrow \text{normalize}(\tau_r)$
- 5: **for**  $t = 1$  to  $N_{\text{steps}}$  **do**
- 6:    $\Psi \leftarrow$  Generate  $N_{\text{ligands}} \times N_{\text{ants}}$  coordinates  $\Psi$  with probability  $p_{\Psi}$
- 7:    $r \leftarrow$  Generate  $N_{\text{ligands}} \times N_{\text{ants}}$  coordinates  $r$  with probability  $p_r$
- 8:   // The following double for loop can be parallelized
- 9:   **for**  $i = 1$  to  $N_{\text{ligands}}$  **do**
- 10:     **for**  $a = 1$  to  $N_{\text{ants}}$  **do**
- 11:        $\Psi[a, i] \leftarrow \text{local\_minimization}(\Psi[a, i], \text{energy } i\text{th ligand})$
- 12:        $r[a, i] \leftarrow \text{local\_minimization}(r[a, i], \text{energy } i\text{th ligand})$
- 13:     **end for**
- 14:   **end for**
- 15:    $\Psi_{\text{L}}^*, r_{\text{L}}^* \leftarrow \text{get\_best\_solution}(\Psi, r, \text{energy all ligands})$
- 16:    $\tau_{\Psi}, \tau_r \leftarrow \text{update\_pheromone}(\tau_{\Psi}, \tau_r, \Psi_{\text{L}}^*, r_{\text{L}}^*, \rho)$
- 17:    $p_{\Psi} \leftarrow \text{normalize}(\tau_{\Psi})$ ;
- 18:    $p_r \leftarrow \text{normalize}(\tau_r)$
- 19: **end for**
- 20: **return** ligand with the lowest energy in its global minimum.

---

## 3 Results

### 3.1 Illustrative example

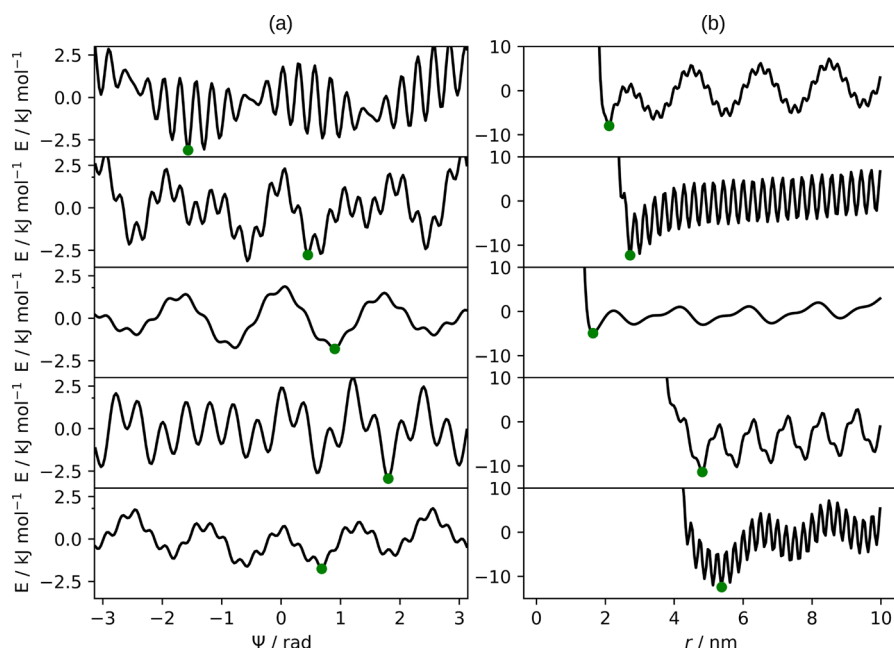
To present the main differences between PLANTS and PLANTS+, we have studied sets of two-dimensional energy functions with one rotational and one translational degree of freedom, representing the binding energy functions of five ligands interacting with a receptor. This is a simplification, as realistic models should also include torsional degrees of freedom describing ligand and receptor flexibility. However, the purpose of the following numerical tests is to demonstrate the potential of PLANTS+ independently of how the energy functions are defined.

For each ligand, the energy function is written as

$$\begin{aligned}
 E(\Psi, r) &= E_{\Psi}(\Psi) + E_r(r) \\
 &= \cos(\mu_1\Psi) + \mu_2 \cos(\Psi) + \\
 &\quad + \mu_3 \sin(\mu_4\pi\Psi) + \mu_5 \cos(\mu_6\pi\Psi) + \\
 &\quad + \frac{1}{2} \left( r - \frac{7}{2} \right)^2 + \epsilon \left[ \left( \frac{r_0}{r} \right)^{12} - 2 \left( \frac{r_0}{r} \right)^6 \right] + \\
 &\quad + \nu_1 \sin(\nu_2\pi r) + \nu_3 \cos(\nu_4\pi r), \tag{12}
 \end{aligned}$$

where  $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6; \epsilon, r_0, \nu_1, \nu_2, \nu_3, \nu_4$  are parameters randomly generated from a uniform distribution (see supplementary information for details). The function  $E(\Psi, r)$  can be decoupled into two components  $E_{\Psi}(\Psi)$  and  $E_r(r)$  which depend on

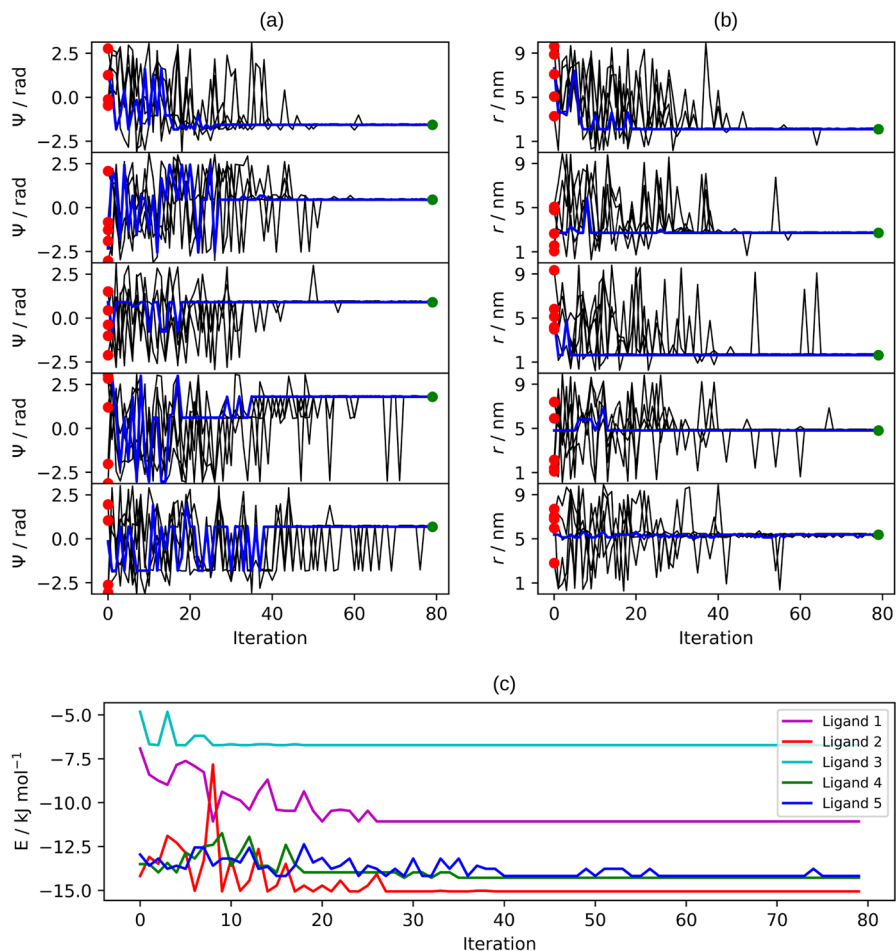




**Fig. 2** Binding energy functions of the illustrative example:  $E_{\Psi}(\Psi)$  component **a**,  $E_r(r)$  component **b**. The green dots mark the global minima

two degrees of freedom corresponding to a rotational and a translational degree of freedom. The pairs of five functions  $E_{\Psi}(\Psi)$  and  $E_r(r)$  are illustrated respectively on the left and the right column of Fig. 2 and are characterized by irregular sequences of valleys and ridges to make the search for the global minimum more difficult. In particular, the functions  $E_{\Psi}(\Psi)$  of the rotational degree of freedom are periodic and are realized by superposing sinusoidal functions with low and high frequencies. Similarly, the translational degree of freedom functions  $E_r(r)$  are realized by means of the Lennard–Jones potential, which well describes the interaction between non-bonding molecules, perturbed by sinusoidal functions to generate local minima. The functions greatly differ from each other due to the randomness of the parameters and the global minima, marked by the green dots, are located in different positions. We also mark that the height of the barriers of the functions  $E_{\Psi}(\Psi)$  are considerably higher than the barriers of the function  $E_r(r)$ : approximately  $3 \text{ kJ mol}^{-1}$  versus  $10 \text{ kJ mol}^{-1}$ .

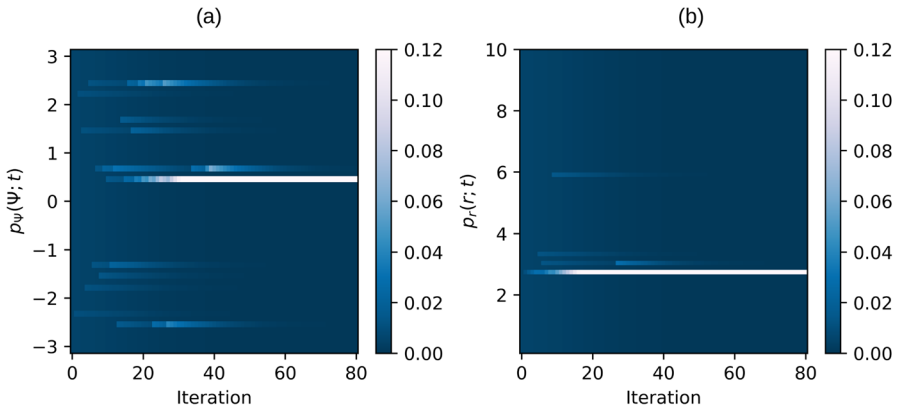
PLANTS and PLANTS+ were applied with the same input parameters: the evaporation rate was  $\rho = 0.1$ , the number of solutions, i.e. the artificial ants generated at each iteration was  $N_{\text{ants}} = 5$ , and the total number of iteration was  $N_{\text{steps}} = 80$ . The range  $[-\pi, \pi]$  rad of the rotational degree of freedom and the range  $[0, 10]$  nm of the translational degree of freedom were both discretized into 200 equal subsets. In order to validate the results, we used the brute force algorithm `scipy.optimize.brute`, included in the SciPy library [34], to find the exact global minima of all the functions.



**Fig. 3** PLANTS. Time evolution of the coordinates (black lines) for each potential:  $\Psi$ -coordinate **a**,  $r$ -coordinate **b**; one ligand per row. The blue line marks the evolution of the best solution at each iteration. The red dots mark the initial solutions, the green dots mark the global minima. **c** Evolution of the energy of the best solution for each ligand

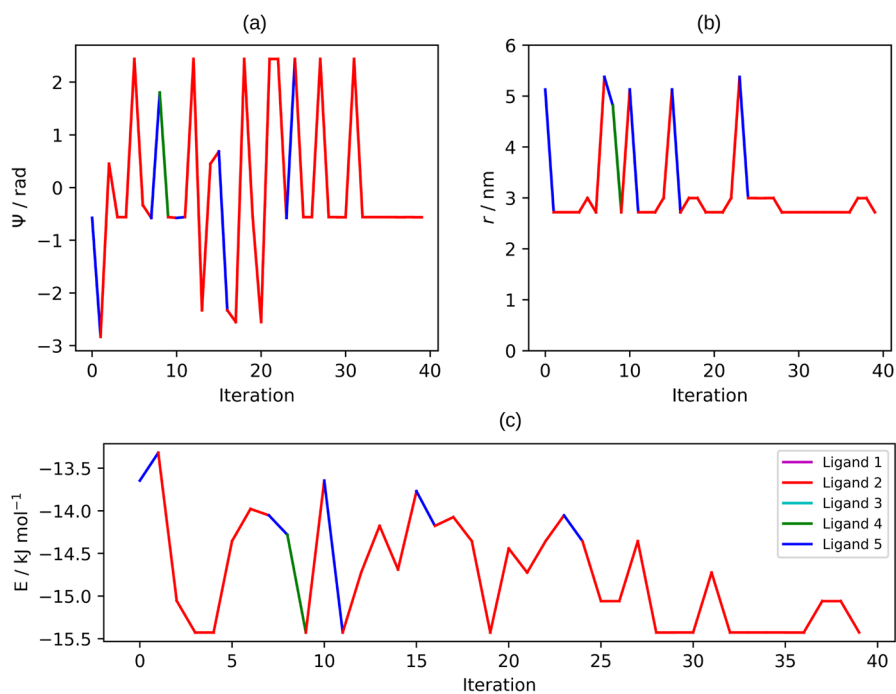
### 3.1.1 PLANTS

We applied PLANTS, in its original form, seeking the global minimum of all five functions. For each energy function, we generated  $N_{\text{ants}} = 5$  artificial ants, i.e. five solutions  $\Psi$  and  $r$  uniformly distributed respectively in the range  $[-\pi, \pi]$  rad and  $[0, 10]$  nm. The initial solutions are marked with red dots in Fig. 3a, where we also plot the solutions generated at each generation (black lines) and the coordinates of the best solutions  $\Psi^*$  and  $r^*$  (blue lines), i.e. the solutions which take the lowest energy value after being minimized. The green dots denote the global minima. Correspondingly, Fig. 3b shows the time evolution of the energy  $E(\Psi^*, r^*)$  of the best solutions.



**Fig. 4** PLANTS. Time evolution of the probability vectors  $p_{\Psi}(\Psi; t)$  **a** and  $p_r(r; t)$  **b** of the second energy function of the illustrative example. Dark and light blue are associated respectively with a low and high probability

Analyzing the results, we observe that the algorithm converges slower with rotational than translational degree-of-freedom functions. Indeed, the coordinates  $\Psi^*$  stabilize after approximately 30 iterations, while the coordinates  $r^*$  need approximately 10–20 iterations. The different behavior is due to how the probability vectors  $p_{\Psi}(\Psi)$  and  $p_r(r)$ , associated with the pheromone vectors, update during the search for minima. In the beginning, the probability vectors describe a uniform distribution, and ideally, they converge over time to a delta function, with the peak located in a neighborhood of the global minimum. Thus, solutions generated subsequently have a higher probability of belonging to a subset that contains the global minimum. However, if the function under consideration has several local minima that assume almost the same energy value as the global minimum, then the pheromone will be distributed over a wider spectrum and new solutions will be generated accordingly. This is what happens with the energy functions  $E_{\Psi}(\Psi)$  under consideration. As an example, we report in Fig. 4 the time evolution of the probability vectors  $p_{\Psi}(\Psi; t)$  (left side) and  $p_r(r; t)$  (right side) related to the second energy function, where the index  $t$  denotes a specific iteration step. We observe that for the rotational degree of freedom, the convergence of the probability vector takes more time despite the average height of the barriers of  $E_{\Psi}(\Psi)$  being greater than  $E_r(r)$ . What makes the determination of the global minimum of  $E_{\Psi}(\Psi)$  difficult is the presence of four local minima at  $\Psi \approx -2.2, -0.5, 0.7$ , and  $2.5$  rad that take approximately the same energy value. Only after 40 iterations does the probability associated with the exact global minimum at  $\Psi \approx -0.5$  rad dominate the rest. In contrast, the global minimum of the function  $E_r(r)$  is less ambiguous, and the probability  $p_r(r)$  converges with less than 10 iterations, identifying the global minimum in the range  $r \in [2, 4]$  nm. Taking a closer look, we identify three close peaks in this interval due to the serrated pattern that characterizes the second function  $E_r(r)$ , but, after a further 30 iterations, only the peak associated with the exact global minimum survives.



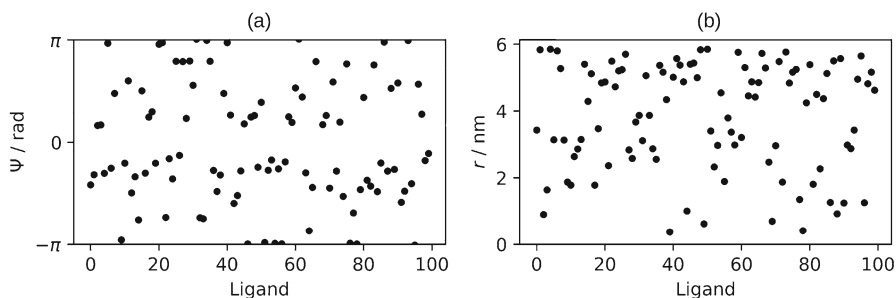
**Fig. 5** PLANTS+. Evolution of the  $\Psi$ -coordinate **a** and  $r$ -coordinate **b** of the best solution with respect to all energy functions. The color denotes the energy function which takes the lowest value during a specific iteration step. **c** Time evolution of the energy of the best solution

In this example, after about 40 iterations, the global minimum for each function is identified and the functions can be ranked according to their energy value shown in Fig. 3b. Thus, we conclude that the second ligand, represented by a red line in the figure, is the most suitable for the receptor in consideration.

### 3.1.2 PLANTS+

We applied PLANTS+ to the same five functions  $E(\Psi, r)$  examined in the previous section. However, unlike PLANTS, we did not use PLANTS+ for finding the global minimum of each energy function individually, but for the global minimum of the global energy function defined in Eq. 11. For this, we defined only one probability vector  $p_\Psi(\Psi)$  for the rotational degree of freedom, and one probability vector  $p_r(r)$  for the translational degree of freedom, which were updated at each iteration accordingly to the best solution  $(\Psi_L^*, r_L^*)$  with respect to the orientation, the position, and the energy function. The probability vectors were used to generate 5 solutions  $(\Psi, r)$  over the range  $[-\pi, \pi]$  rad and  $[0, 10]$  nm, that converge to the global minimum.

In Fig. 5a, we show the time evolution of the coordinate  $\Psi_L^*$  of the best solution on the left side, and of the coordinate  $r_L^*$  on the right side. During each iteration, the lines change color to emphasize which energy function assumes the lowest energy value. In this example, initially, the lines are colored blue since the first ligand takes the least



**Fig. 6** Global minima of each energy function composing the benchmark dataset:  $\Psi$ -coordinate **a**,  $r$ -coordinate **b**

value, then the color becomes red between steps 2 and 7, then it turns blue again. After the 25th iteration step, the color remains red, as the second energy function takes the lowest value.

Fig. 5b shows the time evolution of the energy. Precisely, this graph shows the energy value taken by the best solution ( $\Psi_L^*$ ,  $r_L^*$ ), at a given iteration step, using the energy function that returns the lowest value. In accordance with the graph depicting the evolution of the solution, the best energy function is used to color the line. At first, the algorithm identifies the first ligand as optimal for the receptor, but at the end of the execution, the algorithm returns the second ligand, in accordance with results obtained by PLANTS.

We also observe that between steps 0 and 25, the competition between energy functions involves ligands 2, 4, and 5. Similarly, we observed the same competition with PLANTS in Fig. 3. This is due to the fact that the global minima of each of these ligands assume approximately the same value, as shown in Fig. 2.

### 3.2 Comparative analysis

With the next numerical experiment, we investigate how PLANTS and PLANTS+ depend on the input parameters, and we seek under which settings PLANTS+ is more accurate and more efficient than PLANTS. For this purpose, we randomly generated a set of 100 energy functions  $E(\Psi, r)$  using Eq. 12 and we searched for the energy function which takes the least energy value at its global minimum. The choice of the energy functions was made in order to make difficult the optimization problem, not only generating random sequences of local minima and barriers but distributing the global minima over a large range, as illustrated in Fig. 6. We studied 64 different sets of parameters defined by the combination of the following input parameters:

$$\begin{aligned} N_{\text{ants}} &= 1, 5, 10, 15, \\ \rho &= 0.1, 0.2, 0.3, 0.4, \\ N_{\text{steps}} &= 20, 30, 40, 50. \end{aligned}$$

A total of 20 runs of each algorithm were performed on each set of parameters in order to produce meaningful statistical results. For each set of parameters we measured and compared the execution time and the success rate. The execution time is the real-time in seconds, averaged over all 20 repetitions, required to complete the instructions described in Algs. 1 and 2. To evaluate the success rate, we first applied the brute force algorithm `scipy.optimize.brute` to each energy function to find the exact global minima. Then we defined a prediction, realized by PLANTS or PLANTS+, as “successful” if the relative error was smaller than 5%, and the success rate of a specific set of parameters as the percentage of successful predictions over all 20 repetitions.

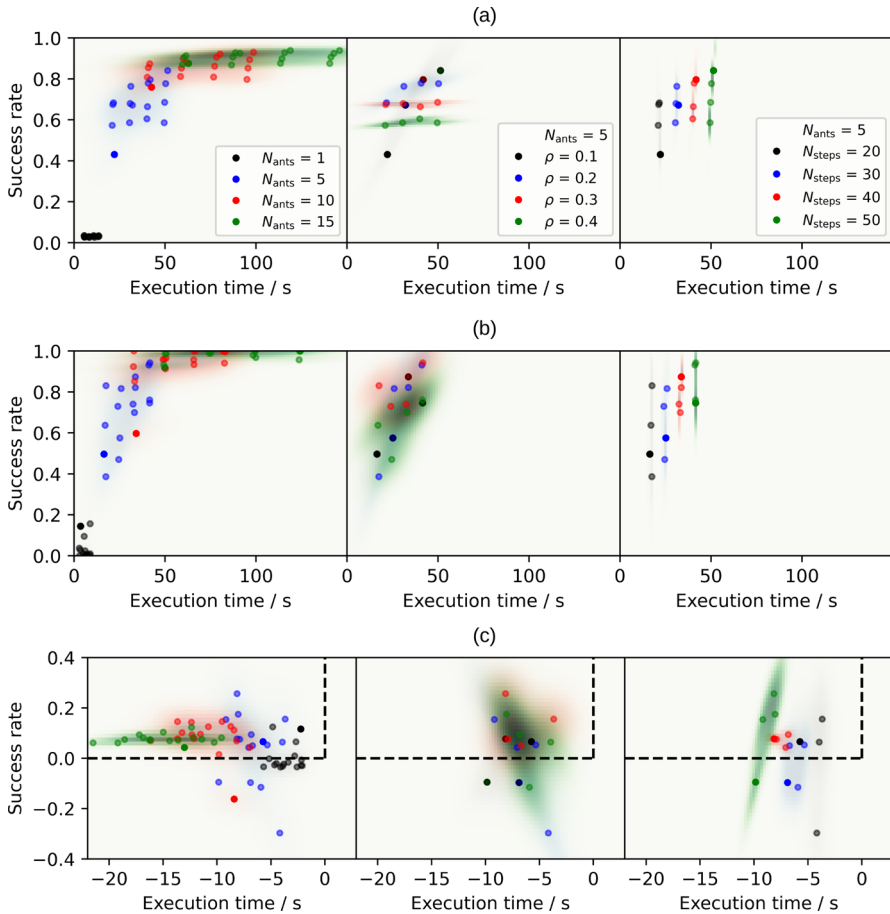
### 3.2.1 PLANTS

Figure 7a reports the results for the algorithm PLANTS. The leftmost panel shows 64 points representing the execution time (horizontal axis) versus the success rate (vertical axis) of each set of parameters. The points have been colored according to the number of generated solutions  $N_{\text{ants}}$ , thus we identify four clusters: using  $N_{\text{ants}} = 1$ , PLANTS never converges, with  $N_{\text{ants}} = 5$  the success rate varies between 40% and 80%, with  $N_{\text{ants}} = 10$  the success rate is between 80% and 90%, and with  $N_{\text{ants}} = 20$  is between 85% and 95%. Correspondingly the execution time increases from approximately 10s to 150s. This result is explained by the fact that more solutions require more calculations, especially during the execution of the simplex algorithm. However, a large number of solutions also facilitates the exploration of the domain and the search for the global minimum.

Likewise, we could color the points to highlight the dependence on the parameters  $\rho$  and  $N_{\text{steps}}$ , but for these two parameters, the figures would be unreadable. Thus, we decided to consider only the set of 16 algorithm executions realized with  $N_{\text{ants}} = 5$  solutions, and we plot in the middle and rightmost panel respectively the dependence on the evaporation rate  $\rho$  and on the iteration steps  $N_{\text{steps}}$ . We observe that  $\rho = 0.4$  (green) and  $\rho = 0.3$  (red) have no impact on execution time, while the success rate is approximately 60% and 70%. This suggests that small evaporation rate values favor accuracy. Intuitively, the reason for this is that if the evaporation rate is too high, the pheromone and probability vectors are reset too quickly, and the information acquired during the first few iterations is lost. By further decreasing the evaporation rate to  $\rho = 0.2$ , the success rate is improved to 80% only if  $N_{\text{steps}} = 30, 40, 50$ . With  $\rho = 0.1$ , the dependence on  $N_{\text{steps}}$  is further emphasized, and the success rate reaches 85% only if  $N_{\text{steps}}$  is equal to 50. Thus, low evaporation rate values require longer algorithm runs to get out of any local minima in which the system could be trapped. Analyzing the rightmost panel, we deduce that the number of iteration steps mainly impacts the execution time and the difference in success rate is due to the parameter  $\rho$ .

In essence, we conclude the following:

- Increasing the number of solutions improves accuracy, but the required execution time exponentially increases.
- Small evaporation rates improve the accuracy, but only if the number of iteration steps is large enough.



**Fig. 7** **a** Success rate (vertical axis) and execution time (horizontal axis) of PLANTS; **b** Success rate (vertical axis) and execution time (horizontal axis) of PLANTS+; **c** Difference in execution time and success rate between PLANTS+ and PLANTS. The points falling within the rectangle with dashed edges indicate that PLANTS+ is more accurate and more efficient than PLANTS using the same input parameters. The legend box of figure **a** applies also to figures **b** and **c**

- The number of iteration steps affects the execution time, but it improves the accuracy only in combination with a small evaporation rate.

### 3.2.2 PLANTS+

We carried out the same analysis with PLANTS+, whose results are reported in Fig. 7b. The color of the points in the graphs highlights how success rate and execution time depend on the parameter  $N_{ants}$  (left panel),  $\rho$  (middle panel), and  $N_{steps}$  (right panel). As for PLANTS, we fixed the number of solutions  $N_{ants} = 5$  to better illustrate the dependence on  $\rho$  and  $N_{steps}$ .

With regard to the number of generated solutions  $N_{\text{ants}}$ , we observe the same behavior as PLANTS: accuracy increases with  $N_{\text{ants}}$ , but at the expense of an exponentially increased execution time. Despite the difficulty of reading the second panel, we can still discern a cluster associated with  $\rho = 0.3$  (red) above  $\rho = 0.4$  (green). Thus, reducing the evaporation rate increases accuracy, but execution time depends on the other parameters. Lastly, the number of iterations  $N_{\text{steps}}$  affects execution time primarily. Then, we conclude that PLANTS and PLANTS+ are dependent on the input parameters in the same manner.

In Fig. 7c, we compare the success rate and execution time of the two algorithms, in order to determine which algorithm is more efficient and accurate. The graph reports the difference

$$(\text{success rate of PLANTS+}) - (\text{success rate of PLANTS}),$$

on the vertical axis, and

$$(\text{execution time of PLANTS+}) - (\text{execution time of PLANTS}),$$

on the horizontal axis. Hence, a positive value on the vertical axis and a negative value on the horizontal axis indicate that PLANTS+ is respectively more accurate and more efficient than PLANTS using the same parameters  $N_{\text{ants}}$ ,  $\rho$  and  $N_{\text{steps}}$ . In all cases, PLANTS+ executions were more efficient than PLANTS: the gain varies from 5 s with  $N_{\text{ants}} = 5$  (black dots) to 20 s  $N_{\text{ants}} = 15$  (green dots). Accuracy worsened in 16 cases, those below the dashed line, but improved in all the other 48 cases. In particular, we observe that in 5 cases with  $N_{\text{ants}} = 1$ , in 12 cases with  $N_{\text{ants}} = 5$ , in 15 cases with  $N_{\text{ants}} = 10$ , and in all cases with  $N_{\text{ants}} = 15$ , the success rate improves. Then, PLANTS+ benefits from increasing the number of solutions  $N_{\text{ants}}$  both in accuracy and efficiency. The rightmost panel reveals another important result. With the same settings, PLANTS+ is more efficient when the number of iteration steps is increased, and the gain in efficiency is proportional to  $N_{\text{steps}}$ . In other words, with PLANTS+ the  $N_{\text{steps}}$  parameter has less impact on execution time than with PLANTS.

## 4 Conclusion and outlook

In this article, we reviewed the optimization method PLANTS and proposed an augmented version, referred to as PLANTS+ in the text, for finding the optimal orientation and position of a ligand at the receptor's binding site. The main difference between the two methods is that PLANTS examines the binding energy functions of ligands individually and determines the best ligand for a given receptor by comparing the energy that each ligand takes at its optimal pose, while PLANTS+ solves a single optimization problem and searches for the optimal position and orientation by comparing the energy functions of the ligands at each iteration step. This modification permits better use of computational resources because the probability vector used to generate new possible solutions in PLANTS+ embeds more information than PLANTS. In fact, the



probability vector is not updated by taking into account only the degrees of freedom of a single ligand, but of a set of ligands under investigation.

We studied how the input parameters affect the results and our analysis shows that PLANTS+ is always more efficient than PLANTS. The accuracy also improves in most of the cases, in particular when the number of solutions generated for each ligand-receptor system, i.e., the input parameter  $N_{\text{ants}}$ , is sufficiently high.

However, we emphasize that in our numerical experiments we have considered the ligand and the receptor as static systems. This approximation is not justified from a chemical point of view, but the primary objective of this work is to show how the Ant Colony Optimisation algorithm can improve its performance with a different use of the available information. In further development of the algorithm, we will take into account also the flexibility of the ligand and receptor and how the electron density clouds interact. Additionally, we plan to take into account the inclusion of kinetic observables, e.g., transition rates, as a criterion for classifying ligands. This will provide a more accurate screening of ligands [35] and will make it possible to study the dependence of docking on external variables [36], e.g. the acidity of the cell membrane, a variable that is considered crucial in the development of undesirable effects [37, 38]. In this work, we have addressed the docking problem. However, as mentioned in the introduction, another important strategy in drug design is based on molecular similarity, i.e., the search for molecules with similar molecular structure. As this is also an optimisation problem that can be solved by heuristic algorithms, a further line of development of PLANTS+ is its application to virtual screening methods based on molecular similarity.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s10910-023-01549-6>.

**Acknowledgements** Not applicable

**Author Contributions** Luca Donati, Konstantin Fackeldey, and Marcus Weber developed the algorithm PLANTS+. Luca Donati designed and conducted the numerical experiments. Luca Donati wrote the manuscript text. Konstantin Fackeldey and Marcus Weber reviewed the manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL. This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Cluster of Excellence MATH+, project AA1-15: “Math-powered drug-design”.

**Availability of data and material** See the supplementary material for an example script that implements both PLANTS and PLANTS+ in Python.

## Declarations

**Conflict of interest** The authors declare that they have no competing interests.

**Consent for publication** All authors gave their consent.

**Ethics approval and consent to participate** This article does not contain any studies with human participants or animals performed by any of the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give

appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. C. Berger, S. Weber-Bornhauser, J. Eggenberger, J. Hanes, A. Plückthun, H.R. Bosshard, Antigen recognition by conformational selection. *FEBS Lett.* **450**(1–2), 149–153 (1999). [https://doi.org/10.1016/s0014-5793\(99\)00458-5](https://doi.org/10.1016/s0014-5793(99)00458-5)
2. J.R. Williamson, Induced fit in RNA-protein recognition. *Nat. Struct. Mol. Biol.* **7**, 834–837 (2000). <https://doi.org/10.1038/79575>
3. K.A. Johnson, Role of induced fit in enzyme specificity: A molecular forward/reverse switch. *J. Biol. Chem.* **283**(39), 26297–26301 (2008). <https://doi.org/10.1074/jbc.r800034200>
4. Y. Miller, B. Ma, R. Nussinov, Polymorphism in Alzheimer A $\beta$  Amyloid Organization Reflects Conformational Selection in a Rugged Energy Landscape. *Chem. Rev.* **110**(8), 4820–4838 (2010). <https://doi.org/10.1021/cr900377t>
5. E. Fischer, Einfluss der configuration auf die wirkung der enzyme. *Chem. Ber.* **27**(3), 2985–2993 (1894). <https://doi.org/10.1002/cber.18940270364>
6. J. Monod, J. Wyman, J.-P. Changeux, On the nature of allosteric transitions: A plausible model. *J. Mol. Biol.* **12**, 88–118 (1965). [https://doi.org/10.1016/s0022-2836\(65\)80285-6](https://doi.org/10.1016/s0022-2836(65)80285-6)
7. D.E. Koshland, G. Namethy, D. Filmer, Comparison of experimental binding data and theoretical models in proteins containing subunits. *Biochemistry* **5**(1), 365–385 (1966). <https://doi.org/10.1021/bi00865a047>
8. P.M. Dean, *Molecular Similarity in Drug Design.*, 1st edn. Springer, Glasgow (1995). <https://doi.org/10.1007/978-94-011-1350-2>
9. A.T. Balaban, Topological and stereochemical molecular descriptors for databases useful in qsar, similarity/dissimilarity and drug design. *SAR and QSAR in Environmental Research* **8**(1–2), 1–21 (1998). <https://doi.org/10.1080/10629369808033259>
10. P.G. Mezey, Computer aided drug design: Some fundamental aspects. *J. Mol. Model.* **6**, 150–157 (2000). <https://doi.org/10.1007/PL00010725>
11. J.M. Amigó, J. Gálvez, V.M. Villar, A review on molecular topology: applying graph theory to drug discovery and design. *Naturwissenschaften* **96**, 749–761 (2009)
12. A. Kumar, K.Y.J. Zhang, Advances in the development of shape similarity methods and their application in drug discovery. *Front. Chem.* **6**, 315 (2018)
13. G.A. Arteca, V.B. Jammal, P.G. Mezey, Shape group studies of molecular similarity and regioselectivity in chemical reactions. *J. Comp. Chem.* **9** (1988)
14. P.G. Mezey, *Shape in Chemistry: An Introduction to Molecular Shape and Topology*, 1st edn. (1993). <https://books.google.de/books?id=SgJwAAAAMAAJ>
15. P.G. Mezey, *Advances in molecular similarity*, 2nd edn. (1998). <https://books.google.de/books?id=katKAQAIAAJ>
16. P. Mezey, Shape-similarity relations based on topological resolution. *J. Math. Chem.* **27**, 61–69 (2000). <https://doi.org/10.1023/A:1019179105155>
17. P. Mezey, A uniqueness theorem of molecular recognition. *J. Math. Chem.* **30**, 305–313 (2001). <https://doi.org/10.1023/A:1015175727727>
18. J.-Y. Trosset, H.A. Scheraga, Reaching the global minimum in docking simulations: a monte carlo energy minimization approach using bezier splines. *Proc. Natl. Acad. Sci. USA* **95**(14), 8011–5 (1998). <https://doi.org/10.1073/pnas.95.14.8011>
19. G. Jones, P. Willett, R.C. Glen, A.R. Leach, R.B.E. Taylor, Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.* **267**(3), 727–48 (1997). <https://doi.org/10.1006/JMBI.1996.0897>
20. 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. *J. Comput.*

- Chem. **19**(14), 1639–1662 (1998). [https://doi.org/10.1002/\(SICI\)1096-987X\(19981115\)19:14<1639::AID-JCC10>3.0.CO;2-B](https://doi.org/10.1002/(SICI)1096-987X(19981115)19:14<1639::AID-JCC10>3.0.CO;2-B)
21. R. Abagyan, M. Totrov, D. Kuznetsov, Icm?: a new method for protein modeling and design?: applications to docking and structure prediction from the distorted native conformation. *J. Comput. Chem.* (1994). <https://doi.org/10.1002/jcc.540150503>
  22. C. McMartin, R.S. Bohacek, Qxp: Powerful, rapid computer algorithms for structure-based drug design. *J. Comput. Aided Mol. Des.* **11**, 333–344 (1997). <https://doi.org/10.1023/A:1007907728892>
  23. C.A. Baxter, C.W. Murray, D.E. Clark, D.R. Westhead, M.D. Eldridge, Flexible docking using tabu search and an empirical estimate of binding affinity. *Proteins.* **33** (1998). [https://doi.org/10.1002/\(SICI\)1097-0134\(19981115\)33:3<367::AID-PROT6>3.0.CO;2-W](https://doi.org/10.1002/(SICI)1097-0134(19981115)33:3<367::AID-PROT6>3.0.CO;2-W)
  24. M. Rarey, B. Kramer, T. Lengauer, G. Klebe, A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.* **261**(3), 470–489 (1996). <https://doi.org/10.1006/jmbi.1996.0477>
  25. T. Ewing, S. Makino, G. Skillman, I. Kuntz, Dock 4.0: Search strategies for automated molecular docking of flexible molecule databases. *J. Comput. Aided Mol. Des.* **15**, 411–28 (2001). <https://doi.org/10.1023/A:1011115820450>
  26. O. Korb, T. Stützel, T. Exner, Plants: Application of ant colony optimization to structure-based drug design. **4150**, 247–258 (2006). [https://doi.org/10.1007/11839088\\_22](https://doi.org/10.1007/11839088_22)
  27. C. Gorgulla, A. Boeszoermyeni, Z.-F. Wang, P. Fischer, P. Coote, K. Padmanabha Das, Y. Malets, S. Dmytro, Y. Moroz, D. Scott, K. Fackeldey, M. Hoffmann, I. Iavniuk, G. Wagner, H. Arthanari, An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **580**, 1–8 (2020). <https://doi.org/10.1038/s41586-020-2117-z>
  28. C. Gorgulla, S.S. Çınaroğlu, P.D. Fischer, K. Fackeldey, G. Wagner, H. Arthanari, Virtualflow ants: ultra-large virtual screenings with artificial intelligence driven docking algorithm based on ant colony optimization. *Int. J. Mol. Sci.* **22**(11), (2021). <https://doi.org/10.3390/ijms22115807>
  29. M. Dorigo, M. Birattari, T. Stützel, Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006). <https://doi.org/10.1109/MCI.2006.329691>
  30. O. Korb, T. Stützel, T. Exner, An ant colony optimization approach to flexible protein-ligand docking. *Swarm Intell.* **1**, 115–134 (2007). <https://doi.org/10.1007/s11721-007-0006-9>
  31. O. Korb, T. Stützel, T.E. Exner, Empirical scoring functions for advanced protein-ligand docking with plants. *J. Chem. Inf. Model.* **49**(1), 84–96 (2009). <https://doi.org/10.1021/ci800298z>
  32. J.A. Nelder, R. Mead, A simplex method for function minimization. *Comput. J.* **7**(4), 308–313 (1965). <https://doi.org/10.1093/comjnl/7.4.308>
  33. L. Han, M. Neumann, Effect of dimensionality on the nelder-mead simplex method. *Optim. Methods Softw.* **21**(1), 1–16 (2006). <https://doi.org/10.1080/10556780512331318290>
  34. P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, Í Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272 (2020). <https://doi.org/10.1038/s41592-019-0686-2>
  35. B.G. Keller, S. Aleksic, L. Donati, Markov state models in drug design. In: Gervasio, F.L. (ed.) *Biomolecular Simulations in Structure-based Drug Discovery*, p. 67. Wiley-Interscience, Weinheim (2019). <https://doi.org/10.1002/9783527806836>
  36. L. Donati, M. Weber, Assessing transition rates as functions of environmental variables. *J. Chem. Phys.* **157**(22), 224103 (2022). <https://doi.org/10.1063/5.0109555>
  37. D.T. Manallack, R.J. Pranker, E. Yuriev, T.I. Oprea, D.K. Chalmers, The significance of acid/base properties in drug discovery. *Chem. Soc. Rev.* **42**, 485–496 (2013). <https://doi.org/10.1039/C2CS35348B>
  38. G. Del Vecchio, D. Labuz, J. Temp (Arnold), V. Seitz, M. Klöner, R. Negrete, A. Rodriguez-Gaztelumendi, M. Weber, H. Machelska, C. Stein, pka of opioid ligands as a discriminating factor for side effects. *Sci. Rep.* **9**, 19344 (2019). <https://doi.org/10.1038/s41598-019-55886-1>