LightDock: a new multi-scale approach to protein–protein docking

摘要

Computational prediction of protein–protein complex structure by docking can provide structural and mechanistic insights for protein interactions of biomedical interest. However, current methods struggle with difficult cases, such as those involving flexible proteins, low-affinity complexes or transient interactions. A major challenge is how to efficiently sample the structural and energetic landscape of the association at different resolution levels, given that each scoring function is often highly coupled to a specific type of search method. Thus, new methodologies capable of accommodating multi-scale conformational flexibility and scoring are strongly needed.

Keyword：protein docking；scoring function

1 Introduction

Protein–protein interactions are involved in virtually all cellular processes, such as protein expression regulation, cell-cycle control or immune response, among many others (Eisenberg et al., 2000). Characterizing such interactions at atomic level is of paramount importance to better understand pathological conditions at molecular level. However,structural data at atomic resolution is only available for a tiny fraction of the estimated number of protein–protein complexes in human (Mosca et al., 2013; Stumpf et al., 2008;Venkatesan et al., 2009). In this context, computational docking is being increasingly applied for the structural modeling of protein-protein interactions, aiming to complement experimental methods.From a technical point of view, the docking problem presents two main challenges: the efficient sampling of the conformational and orientation space in search of near-native structures (sampling),and the identification of such near-native structures among the many models generated (scoring) (Moal and Bates, 2010). In most of the cases, the applicability of a given scoring function is strongly dependent on the sampling approaches used. The widely used Fast-Fourier Transform (FFT) based methods can efficiently generate geometrically complementary rigid-body docking poses (Gabb et al.,1997; Katchalski-Katzir et al., 1992).

2 Related works

The development of new scoring functions that can be independently applied to different sets of docking models generated by a variety of docking methods is an active area of research (Brenke et al., 2012;Moal et al., 2013a,b; Schneidman-Duhovny et al.,2012). However,as above mentioned, the use of new scoring functions in docking has been traditionally limited by the type of sampling method. On the one hand, grid-based docking search methods have difficulties in efficiently including energy-based scoring functions. On the other hand,molecular dynamics, minimization or Monte-Carlo sampling methods usually are linked to a specific force-field and cannot easily accept new scoring schemes. It is thus necessary the development of new sampling schemes in docking that can use multi-scale representation of the proteins, accept flexibility at different degrees and accommodate a large variety of new scoring functions.[1]。

2.1 Swarm Intelligence

Swarm Intelligence (SI) is a family of the artificial intelligence algorithms inspired by emergent systems in nature,which can perform a more efficient search in a complex space, quite independently on the scoring function to optimize. Basically, those algorithms make use of simple agents that interact locally in a decentralized way, and whose interactions lead to complex emergent patterns or systems in nature, e.g. fish schooling or termite mounds. SI algorithms have been applied to protein–protein docking, such as Particle Swarm Optimization (PSO) in SwarmDock (Li et al., 2010).

2.2 Glowworm Swarm Optimization

a bio-inspired algorithm from the SI family, which is based in the concept that in nature, glowworms are being attracted by other mates depending on the quantity of emitted light. This metaphor is used by the GSO algorithm for simultaneously capturing multiple local optima in multimodal functions. Each agent in the algorithm, a glowworm, carries out a quantity of luciferin which encodes the actual fitness of the position of the agent in the explored search space. The algorithm has been applied to many different problems (Huang and Zhou, 2011;Krishnanand and Ghose, 2009b; Liao et al., 2011), but not explicitly to protein–protein docking. GSO has some advantages over PSO(Krishnanand and Ghose, 2009a).

3 Method

3.1 GSO algorithm

The agents in the GSO algorithm are defined as glowworms which carry a luminescent quantity called luciferin. At each step of the simulation, the quantity of luciferin l depends on the evaluation of the complex energy by the user-defined scoring S function in the actual search space x and the previous value of the luciferin based on the trajectory of the given glowworm (Eq. 1). Decay of the quantity of luciferin is controlled by the q variable, and c represents the enhancement constant, i.e. how much affects the actual evaluation of the energy in the luciferin quantity.

$$l_i(t+1) = (1-\rho) \cdot l_i(t) + \gamma \cdot S\left(x_i(t+1)\right) \tag{1}$$

In LightDock, these parameters are defined by default as: p ¼ 0:4,c ¼ 0:6, initial luciferin 5:0. Each glowworm gi initially represents a specific position in the translational and rotational space of the ligand (Eq. 2), where tx,ty and tz are the components of the vector vorigin?ligandcenter and qw, qx,qy and qz are the components of the quaternion that represents the ligand rotation in the four dimensional quaternions space. The use of quaternions needs fewer variables than rotation matrices, and avoids the known gimbal lock problem of sampling based on Euler angles or polar coordinates

$$g_i = [t_x, t_y, t_z, q_w, q_x, q_y, q_z] \tag{2}$$

In addition, the framework has the capability of using the anisotropic network model (ANM) to introduce a certain degree of backbone flexibility during the protein‑protein binding process. In this case, each glowworm agent represents, in addition to a translation/rotation ligand position, the extent of deformation along each non-trivial normal mode for the receptor, nr, and the ligand, nl, in the optimization vector(Eq. 3). The number of normal modes is customizable for the recep-tor, R, and the ligand, L.

3.2 Initial receptor/ligand models

The setup of the initial glowworm swarms is as follows. Initially, a fixed number of initial swarm centers Ns (by default 400) are defined around the receptor, by using the spiral method (Rakhmanov et al.,1994), and are projected using a ray-tracing technique to find the closest atom from the receptor at the distance of the maximum radius of the ligand. To guarantee a correct sampling over the surface, a certain density of these centers is needed (Supplementary Methods 1.1). For each initial swarm center, glowworms are defined by randomly positioning the ligands (by default 300) so that their center of coordinates are placed within a 10 A˚ radius sphere from the given swarm center. LightDock framework can also support the use of precalculated ligand poses generated by FTDock (Gabb et al.,1997)

3.3 GSO sampling

As above described, sets of initial receptor/ligand putative models (glowworms) are defined for their use in independent simulations.Each given glowworm gi will move towards the best-scoring (luciferin) neighbor glowworm gj with a given probability pij (Eq. 3)(Krishnanand and Ghose, 2009a),

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \tag{3}$$

where the number of neighbor glowworms (Ni) of glowworm gi is defined by its vision range distance(initially rid ¼ 5:0 ˚A), limited by the maximum number of neighbors (by default Nmax ¼ 5).

The distance in the search space between two receptor/ligand models (glowworms) used to update this list of neighbors (Ni) is computed as that between the centers of the minimum ellipsoids of the ligands (translation and rotation of the receptors does not vary).Other definitions of distance based on RMSD did not improve sampling (Supplementary Methods 1.3). The vision range of each glowworm rid is dynamically updated at each step (Eq. 4) (Krishnanand and Ghose, 2009a) up to a maximum vision range (by default rs ¼ 20:0 ˚A)

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t) + \beta\left(n_t - |N_i(t)|\right)\right\}\right\} \tag{4}$$

where the b parameter indicates how the vision range depends on the number of neighbors in the GSO algorithm (by default b ¼ 0:16).

4 Implementation details

4.1 Comparing with released source codes

The original paper code is available. I used the obtained code to replicate the work and successfully completed the text counter attack.

After finishing the attack on the language model provided in the article, I used the method mentioned in the article to attack the latest model, and found that the result was not as good as the model used in the experiment

4.2 Experimental environment setup

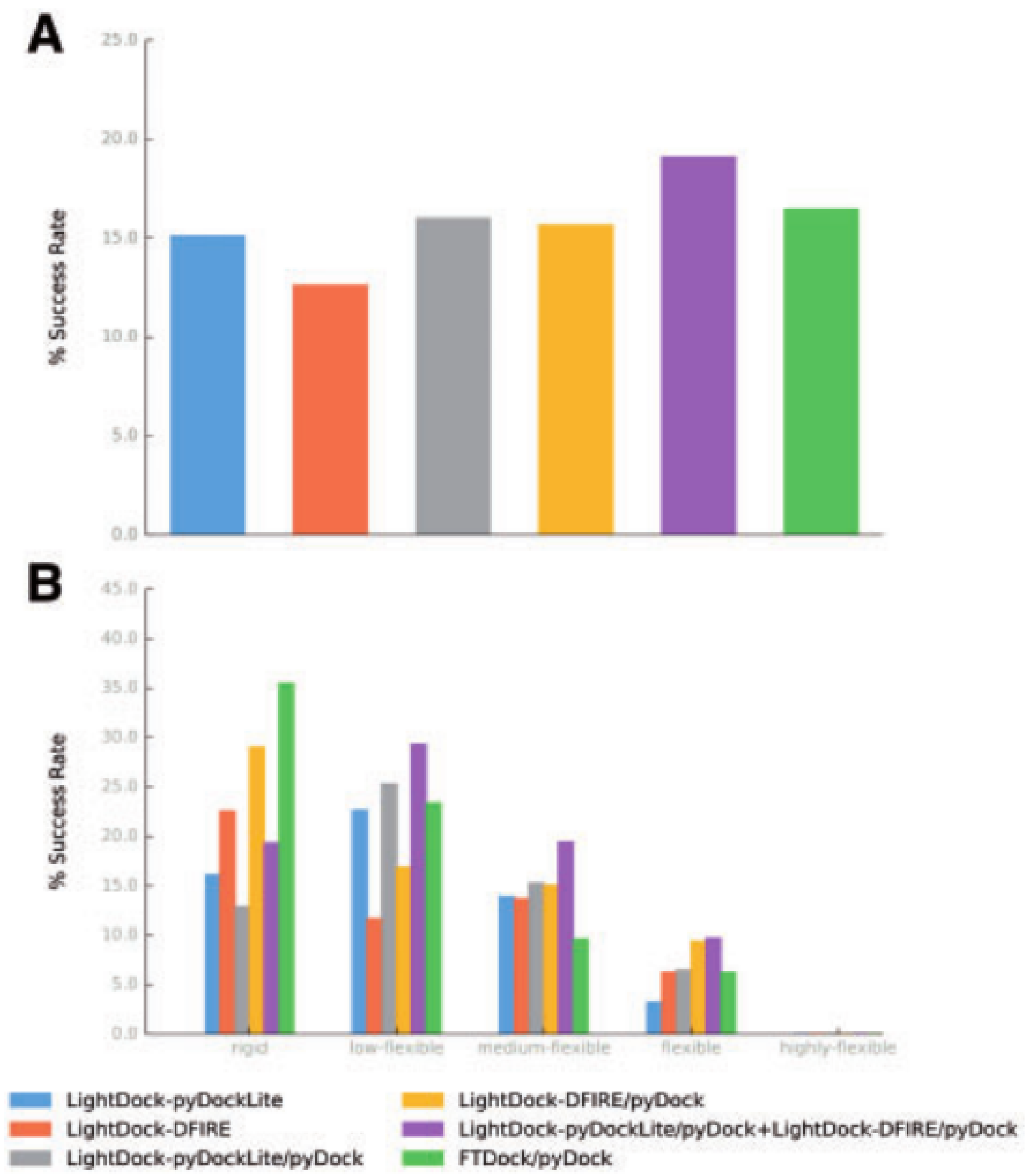
Lightdock:Installing NumPy, Scipy, Cython, Biopython, Nose and MPI4py, libraries are usually available as packages in most of GNU/Linux distributions. In case of using cpydock scoring function or to execute the tests, Freesasa library has to be installed and compiled with the python-binding options. Tested version in LightDock is 1.1 . To install freesasa 1.1. A directory called lightdock-python2.7 is now available. This is the path necessary for setting the enviroment variable LIGHTDOCKHOME in your bash.

Data Set: PDB,Biomolecules are hierarchical structures. For example, proteins are composed of linear chains of amino acids that (often) fold into compact subunits which then can associate into higher level assemblies with other proteins, small molecule ligands, and water or other solvent molecules. Biomolecules in the Protein Data Bank (PDB) archive are organized and represented using this hierarchy to simplify searching and exploration.

5 Results and analysis

The predictive performance of LightDock was tested on the Protein–Protein Docking Benchmark 5.0, composed of a total of 230 complexes. The predictive success rates were based on the percentage of cases in which at least one near-native solution was found within the top N solutions (N ¼ 10, 100), as ranked according to the corresponding scoring function. Near-native solutions were defined as those ones with a ligand RMSD < 10 A° with respect to the ligand position in the reference structure (when receptor molecules are superimposed). We tested the performance of LightDock (using default parameters; see Methods) with DFIRE (Zhou and Zhou, 2002) scoring function (LightDock-DFIRE), as well as that of LightDock with a faster implementation of the pyDock (Cheng et al., 2007) scoring function (Supplementary Methods 1.6) called pyDockLite (LightDock-pyDockLite).

For each docking case, LightDock generated a total of 120 000 poses, which were clustered as described in the Methods section. After clustering, the final number of docking models obtained by LightDock-pyDockLite ranged between 600 (PDB 1CLV) and 6387 (PDB 1DE4), and near-native poses were found in 70 of the cases. In LightDock-DFIRE, the total numberof docking models ranged between 748 (PDB 1CLV) and 6713(PDB 1AKJ), and near-native poses were found in 75 of the cases.As a further test, docking simulations on the same complex using different scoring functions were combined in order to capture different near-native predictions. With this purpose, all the models independently generated by LightDock-DFIRE or by LightDock pyDockLite were merged and re-scored by pyDock scoring function (i.e. combination of LightDock-DFIRE/pyDock and LightDock-pyDockLite/pyDock). The scoring function in pyDock has shown excellent performance in the scorers round of the CAPRI community-wide experiment (Lensink et al., 2016; Pallara et al.,2013), and it is sufficiently fast not to become an overhead in the total computation time of LightDock.



As can be seen in Figure 2A, the use of pyDockLite scoring function within LightDock showed better success rates for the top 10 docking solutions than when using the DFIRE scoring function. The performance of LightDock-pyDockLite is only slightly worse than that of pyDock applied on FTDock docking models (FTDock/pyDock), as in pyDock server (Jime´nez-Garcıá et al., 2013). For the top 100 success rates (Supplementary Fig. S3A), this difference in performance between LightDock-pyDockLite and LightDock- DFIRE scoring functions is higher, and interestingly, LightDock-pyDockLite top 100 success rate is even slightly better than that ofthe standard FTDock/pyDock.

6 Conclusion and future work

We have presented here a new protein－protein docking protocol called LightDock, which is based on the GSO algorithm for sampling the translational and rotational space of protein－protein docking, and ANM representation for the inclusion of flexibility.LightDock aims to be a publicly available framework for testing and developing new scoring strategies for protein－protein docking. The use of pyDockLite scoring function during the search provides comparable success rates to state-of-the-art protocols, and the combination with additional functions, like DFIRE, can further improve the predictions. This multi-scale docking framework has capabilities for the use of many different scoring functions (alone or in combination) and the inclusion of flexibility at different resolution levels.

7 Reference

LightDock: a new multi-scale approach to protein–protein docking Brian Jiménez-García, Jorge Roel-Touris, Miguel Romero-Durana, Miquel Vidal, Daniel Jiménez-González and Juan Fernández-Recio Bioinformatics, Volume 34, Issue 1, 1 January 2018, Pages 49–55

LightDock goes information-driven Jorge Roel-Touris, Alexandre M.J.J. Bonvin, Brian Jiménez-García Bioinformatics, btz642

Integrative Modeling of Membrane-associated Protein Assemblies Jorge Roel-Touris, Brian Jiménez-García Alexandre M.J.J. Bonvin Nat Commun 11, 6210 (2020)