BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 3 2014, pages 437–438
doi:10.1093/bioinformatics/btt679

Structural bioinformatics

Advance Access publication November 22, 2013

jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework

Esteban López-Camacho, María Jesús García Godoy, Antonio J. Nebro and José F. Aldana-Montes*

Lenguajes y Ciencias de la Computación, Universidad de Málaga, Bulevar Louis Pasteur 35, 29071, Málaga, Spain

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: Molecular docking is a method for structure-based drug design and structural molecular biology, which attempts to predict the position and orientation of a small molecule (ligand) in relation to a protein (receptor) to produce a stable complex with a minimum binding energy. One of the most widely used software packages for this purpose is AutoDock, which incorporates three metaheuristic techniques. We propose the integration of AutoDock with jMetalCpp, an optimization framework, thereby providing both single- and multi-objective algorithms that can be used to effectively solve docking problems.

Results: The resulting combination of AutoDock + jMetalCpp allows users of the former to easily use the metaheuristics provided by the latter. In this way, biologists have at their disposal a richer set of optimization techniques than those already provided in AutoDock. Moreover, designers of metaheuristic techniques can use molecular docking for case studies, which can lead to more efficient algorithms oriented to solving the target problems.

Availability and implementation: jMetalCpp software adapted to AutoDock is freely available as a C++ source code at http://khaos.uma.es/AutodockjMetal/.

Contact: jfam@lcc.uma.es

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 29, 2013; revised on September 24, 2013; accepted on November 16, 2013

1 INTRODUCTION

The development of techniques such as high-throughput protein purification, X-ray crystallography and nuclear magnetic resonance spectroscopy have contributed in the discovery of many structural details of protein–protein and ligand–protein complexes. Such advances in association with the completion of the human genome sequencing have allowed the identification of many therapeutic drug targets (Meng *et al.*, 2011). In this context, molecular docking is a method of structure-based drug design that has been in development since the early 1980s. This approach is based on two well-defined steps: (i) determining the position and orientation of the ligand within the target receptor and (ii) assessing the affinity between the two structures by analyzing the binding energy scores and selecting those ligand–protein conformations with minimum binding energies.

*To whom correspondence should be addressed.

For the past two decades, there has been an attempt to improve docking molecular techniques. Therefore, many researchers in this field have focused on improving the quality of the docking results. This depends on the energy scoring function for evaluating the results obtained and an optimization algorithm to detect the best state characterized by a minimum of binding energy. Furthermore, another challenge is to develop software solutions, the scoring functions of which allow the rotation around torsional degrees of freedom of the flexible ligands and side chains of receptors. An example of these software packages is AutoDock (Morris *et al.*, 2009), a C++ toolbox, which has become the most cited and one of the most used software packages in molecular modeling studies, and also an efficient tool for virtual drug screening, which has been applied in real cases that involve flexibility (Cosconati *et al.*, 2010). The release of AutoDock 4.2 provides three optimization algorithms: a simulated annealing and two genetic algorithms (GAs), one of which, referred to as the Lamarckian GA, incorporates a local search. These algorithms are metaheuristic techniques, which are high-level techniques that rule a set of underlying simpler methods (typically, heuristics) designed to find optimal or near-optimal solutions to a given optimization problem (Blum and Roli, 2003).

As mentioned, one important research area is to find appropriate search methods capable of producing the most accurate solutions to the docking problems. In this context, this article proposes a software platform oriented to provide a set of metaheuristic techniques allowing molecular docking problems to be optimized. Our approach has been to integrate AutoDock with jMetal, an object-oriented framework for multi-objective optimization with metaheuristics (Durillo and Nebro, 2011), which has become one of the most popular software tools in the field. By multi-objective optimization, we refer to solving problems that have two or more conflicting objectives that have to be maximized or minimized at the same time. If an optimization problem is composed of a unique objective function, we say that it is a single-objective problem. The underlying idea is to run the algorithms in jMetalCpp and to use the scoring function of AutoDock 4.2.

The original jMetal software is written in Java script, so we have built a C++ implementation from scratch called jMetalCpp (http://jmetalcpp.sourceforge.net), the idea being to achieve a clean and efficient integration with AutoDock 4.2. By doing this, the resulting platform can benefit, on the one hand, the biologists interested in applying other algorithms than those provided by AutoDock to solve their target problems. On the

other hand, researchers in metaheuristics frequently search for real-world problems to assess the performance of new algorithms, so our solution could lead to the appearance of more efficient techniques for solving molecular docking problems.

To the best of our knowledge, this platform is the first to integrate a set of single- and multi-objective metaheuristics to solve docking problems.

2 METHODS

Although the original jMetal Java-based framework, jMetalCpp, incorporates a set of multi-objective algorithms, it also includes a number of single-objective techniques that have been assessed using a set of benchmarking problems. The first group of algorithms includes particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), a number of variants of GAs (steady-state, generational and cellular) and a differential evolution solver (DE) (Storn and Price, 1997). As for multi-objective metaheuristics, jMetalCpp provides classic techniques such as Non-dominated Sorting Genetic Algorithm-II (Deb *et al*., 2002) and more recent state-of-the-art algorithms such as the speed-constrained multi-objective PSO (Nebro *et al*., 2009) and Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Zhang and Li, 2007).

To solve the molecular docking problem, our purpose has been to carry out a clean integration of AutoDock 4.2 and jMetalCpp with certain goals in mind: the resulting software should be easy-to-use by AutoDock users, and it should be efficient (Supplementary Fig. S1). The execution parameters of a given experiment are configured using a docking parameter file, which contains the settings of the algorithm to be executed. If a jMetalCpp metaheuristic has been selected, then jMetalCpp takes control. As metaheuristics are iterative algorithms that work with sets of tentative solutions that are modified according to a number of operators (e.g. in the case of GAs the operators are selection, crossover and mutation), whenever a new solution has to be evaluated, it is sent back to AutoDock to apply its scoring function. If the running algorithm is single-objective, AutoDock returns the total free binding energy; in the case of a multi-objective technique, we adopt the approach of Janson *et al*. (2008) and two values are returned: the intermolecular energy E_{inter} that describes the binding affinity of the ligand–receptor conformation, and the intramolecular energy E_{intra} that characterizes the ligand deformation. Once the algorithm iterations have finished, the best solution (or solutions in the case of selecting a multi-objective technique) is returned to the AutoDock code, so the docking log file is created, containing the results obtained.

In this way, we achieve the first of the aforementioned goals, making the software easily usable. The second goal, efficiency, is achieved because of the use of multi-threading. AutoDock and jMetalCpp run in their own thread inside the same process, communication between them is carried out through shared memory and there is a negligible overhead when they synchronize.

3 DISCUSSION

The use of the jMetalCpp framework provides several advantages for those researchers in drug discovery who are interested in improving the docking results with metaheuristics that are different from the searching methods provided by AutoDock 4.2. To demonstrate the benefits of our proposal, we evaluated a set of single-objective metaheuristics implemented in jMetalCpp on a first standard benchmark provided by Morris *et al*. (1998) and a second benchmark that involves flexibility in side chains of arg-8 residue of human immunodeficiency virus-protease receptors (Morris *et al*., 2009). In this article, we have demonstrated that it is possible to improve upon the results obtained by the AutoDock algorithms in some scenarios (López-Camacho *et al*., 2013).

Furthermore, jMetalCpp can be considered a useful tool for advanced users (or developers) with little C++ knowledge, who can implement new metaheuristic techniques that can be integrated into this framework with relative ease. Consequently, these techniques can be quickly integrated inside AutoDock. The Web site of jMetalCpp provides user instructions on how to install and run the jMetalCpp framework.

For future work, additional developments and improvements to be implemented in jMetalCpp are being planned, such as the integration of new single- (e.g. CMA-ES) and multi-objective metaheuristic techniques (e.g. SMS-EMOA). We are also considering integrating the AutoDock Vina docking function (Trott and Olson, 2010), a recently developed energy function, the use of which is spreading throughout the AutoDock user community.

Funding: The Project Grant (TIN2011-25840) (Ministerio de Ciencia e Innovación del Gobierno de España) and (P11-TIC-7529) (Consejería de Economía, Innovación, Ciencia y Empleo de la Junta de Andalucía) have supported this work.

Conflict of interest: none declared.

REFERENCES

Blum,C. and Roli,A. (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput. Surv.*, **35**, 268–308.

Cosconati,S. *et al.* (2010) Virtual screening with AutoDock: theory and practice. *Expert Opin. Drug Discov.*, **5**, 597–607.

Deb,K. *et al.* (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, **6**, 182–197.

Durillo,J.J. and Nebro,A.J. (2011) jMetal: a Java framework for multi-objective optimization. *Adv. Eng. Softw.*, **42**, 760–771.

Janson,S. *et al.* (2008) Molecular docking with multi-objective particle swarm optimization. *Appl. Soft Comput.*, **8**, 666–675.

Kennedy,J. and Eberhart,R. (1995) Particle swarm optimization. In: *IEEE IJCNN*. Vol. 4, IEEE, Perth, WA, Australia, pp. 1942–1948.

López-Camacho,E. *et al.* (2013) Solving molecular flexible docking problems with metaheuristics - a comparative study. *Appl. Soft Comput.*

Meng,X.Y.Y. *et al.* (2011) Molecular docking: a powerful approach for structure-based drug discovery. *Curr. Comput. Aided Drug Des.*, **7**, 146–157.

Morris,G.M. *et al.* (1998) Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.*, **19**, 1639–1662.

Morris,G.M. *et al.* (2009) AutoDock4 and autoDockTools4: automated docking with selective receptor flexibility. *J. Comput. Chem.*, **30**, 2785–2791.

Nebro,A. *et al.* (2009) SMPSO: a new PSO-based metaheuristic for multi-objective optimization. In: *Computational Intelligence in Miulti-criteria Decision-Making, 2009. mcdm'09. IEEE Symposium.* pp. 66–73.

Storn,R. and Price,K. (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, **11**, 341–359.

Trott,O. and Olson,A.J. (2010) AutoDock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.*, **31**, 455–461.

Zhang,Q. and Li,H. (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.*, **11**, 712–731.