Parallel Evolutionary Optimization Algorithms for Peptide-Protein Docking

Sergey Poluyan[1,⋆] and *Nikolay* Ershov[2,⋆⋆]

[1] *Institute of System Analysis and Control, Dubna State University, Universitetskaya str. 19, Dubna 141980, Moscow Region, Russian Federation*
[2] *Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Leninskie Gory 1, Bldg. 52, Moscow 119991, Russian Federation*

Abstract. In this study we examine the possibility of using evolutionary optimization algorithms in protein-peptide docking. We present the main assumptions that reduce the docking problem to a continuous global optimization problem and provide a way of using evolutionary optimization algorithms. The Rosetta all-atom force field was used for structural representation and energy scoring. We describe the parallelization scheme and MPI/OpenMP realization of the considered algorithms. We demonstrate the efficiency and the performance for some algorithms which were applied to a set of benchmark tests.

1 Introduction

Protein-peptide docking (PPeD) aims to predict peptide binding sites on protein surfaces and the associated binding affinities. Traditional experimental methods for binding site determination include crystallography, nuclear magnetic resonance and site-directed mutagenesis and other techniques [1]. Despite their accuracy, efficiency and the huge amount of details they can provide, they are expensive and very labour and skill demanding. Moreover, protein-peptide complexes are more difficult to crystallize than individual proteins. While docking is relatively cheap, it is a prediction method only. Thus, using computational methods have become a popular research endeavor in recent years. Most of them involve stochastic optimization at different stages [2]. The main advantage of using stochastic methods is the possibility of including various knowledge-based information. Furthermore, it is more computationally attractive rather than molecular dynamics simulations.

The current approaches to PPeD are based on Anfinsen's hypothesis [3] that native-like complex conformations represent unique, low-energy, thermodynamically stable conformations. Therefore, the PPeD problem can be considered as a global optimization problem where the objective is to find the complex conformation with the lowest energy. The main motivation of this study is to compare different evolutionary algorithms and identify the most effective strategies within a certain force-field. Solving PPeD problems typically involve the use of combined methods which require a number of various steps and special techniques. However, such approaches are beyond the scope of the current study.

[⋆] e-mail: svpoluyan@gmail.com
[⋆⋆] e-mail: ershovnm@gmail.com

2 Protein-peptide docking

The interaction between peptide and protein can be described by an objective function calculated with respect to three components representing degrees of freedom: (1) the translation of the peptide, involving the three axis values (x, y, z) in cartesian coordinate space; (2) the peptide orientation, modeled as a four variables quaternion; and (3) the flexibilities, represented by the free rotation of torsion (dihedral angles) of the peptide and side-chain of the protein.

As illustrated in Figure 1, the problem solution is encoded by a real-valued vector of $n + 7$ variables. The first three values correspond to the peptide translation. The next four values correspond to the peptide orientation with range of $[-1, 1]^3$ including the angle slope w with range of $[0, \pi]$. The remaining n values are peptide backbone and side-chain dihedral angles (ϕ, ψ, ω, and χ_{1-4}) and protein side-chain dihedral angles within search area. The backbone and side-chain torsion variables are measured in radians and encoded in the range of $[-\pi, \pi]$, except for peptide backbone angle ω with range of $[\pi - \delta, \pi + \delta], \delta = 0.2$ which is locked in the trans-state. The remaining degrees of freedom (valence angles and bond lengths) are idealized with respect to force-field values and do not change during the conformational search.

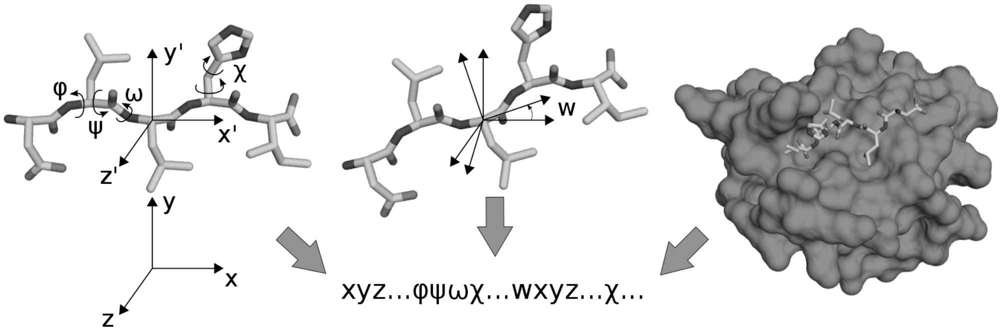

Figure 1. Solution encoding: peptide degrees of freedom, translation, rotation; protein side-chain dihedral angles

The optimization problem is formulated as minimization of the binding energy (BE). This energy (shown in eq. (1)) is defined as the energy of the bound state minus the energy of the unbound state.

$$E_{\mathrm{BE}} = E_{\mathrm{complex}} - (E_{\mathrm{protein}} + E_{\mathrm{peptide}}). \tag{1}$$

The Rosetta 3.8 [4] framework was used for full-atom complex structure representation and scoring (energy evaluation). The Rosetta all-atom force-field is a dedicated structure prediction and docking force-field. It has borrowed much from classical molecular-mechanics force-fields: Lennard-Jones 6-12 potential, E_{vdw}, Lazaridis-Karplus implicit solvation model, E_{sol}, Coulombic electrostatic potential, E_{elec}, etc. The main feature of this force-field is that it uses several knowledge-based terms. For instance, probability of backbone ϕ, ψ angles, E_{rama}. For that reason the energy score has not a direct conversion to physical energy units like kcal/mol. The *talaris2014* energy function is computed from a linear combination of 16 energy terms E_i which are calculated as a function of geometric degrees of freedom, Θ, chemical identities, aa, and scaled by a weight on each term, w, as shown in eq. (2).

$$E_{\mathrm{total}} = \textstyle\sum_i w_i E_i(\Theta_i, aa_i) = \underbrace{w_v E_{\mathrm{vdw}} + \cdots + w_e E_{\mathrm{elec}} + w_s E_{\mathrm{sol}}}_{\text{physics−based terms (kcal/mol)}} + \underbrace{w_r E_{\mathrm{rama}} + \cdots + w_d E_{\mathrm{dunbrack}}}_{\text{knowledge−based terms}}. \tag{2}$$

There are a lot of statistics about degrees of freedom: neighbor-dependent Ramachandran plots [4], backbone-independent [5] and backbone-dependent [6] libraries for side-chain dihedral angles. These libraries have been used to exclude impossible conformations by creating 1–4 dimensional cumulative distribution functions that had been derived from given probability density functions.

3 Algorithms and Implementation

In general, evolutionary algorithms have a similar structure with operators like mutation, recombination, and selection. However, in this study we focus on three algorithms with radically different structure: Competitive Swarm Optimizer (CSO) [7], Particle Swarm Optimization (PSO) [8], and Adaptive Differential Evolution with Optional External Archive (JADE) [9]. These techniques have been selected as they are widely used to solve various real-coded optimization problems, both model and real-world [2].

One of the many advantages of the evolutionary algorithms is that they are easy to parallelize. It is possible to parallelize specific operations, or to parallelize the evolutionary process itself. We considered algorithms where function evaluation for each individual is out from main operators and presented as an independent operation which consists in function evaluation for a whole population. Then the simplest parallelization scheme can be used. First, the population must be divided into groups proportionally to number of cluster nodes. Second, all groups must be sent to cluster nodes using MPI technology according to scheme: one process – one node. Third, objective function must be evaluated for individuals in groups using OpenMP technology within each MPI process.

The calculations were done using the HybriLIT (LIT JINR) cluster with two Intel Xeon twelve-core processor at each node. The effect of parallelization for JADE is presented in table 1. All the presented algorithms have similar to JADE result due to the equal level of complexity.

Table 1. Performance of the applied scheme of parallel calculations

Threads / Nodes	4 / 1	8 / 1	12 / 1	16 / 1	24 / 1	24 / 2	48 / 2
Speed-up	3.31	6.28	9.03	11.84	17.1	15.85	31.1
Efficiency	0.82	0.78	0.75	0.74	0.71	0.66	0.64

4 Results and Discussion

The performance of the selected algorithms has been assessed on the set of structures from [1] which are presented in table 2. There were 10 independent runs for each algorithm with the number of energy evaluations equal to 10^7 per run. Typical execution time for one run using one thread vary from 15 hours to 22 hours. It depends on number of degrees of freedom. In the case of local docking FlexPepDock (FPD) [10] protocol from the Rosetta framework was used with comparable run time. It performs a high-resolution PPeD using a Monte Carlo-Minimization-based approach to refine all the peptide's degrees of freedom (rigid body orientation, backbone and side chain flexibility) as well as the protein receptor side chains conformations. For the case of global docking we compare results with CABS-dock [11] and pepATTRACT [12]. The preparation of structures was performed using the Rosetta Relax protocol.

Table 2. Protein-peptide complexes with their accession codes from the RCSB Protein Data Bank

Protein		Peptide	Problem		
PDB id:Chain	Length	Sequence	Docking type	Search space	Dimension
2CYH:A	164	AP	Local	sphere, $R = 5$Å	25
1JWG:B	140	DLLHI	Local	two spheres $R_1 = R_2 = 5$Å	54
2HO2:A	33	P_9L	Global	sphere with radius of 60Å	93

The FPD protocol requires the initial starting position of the peptide. We considered two starting states relative to the native state: a random 3d rotation with small translation (FPD2) and rotation along one axis in the binding spot (FPD1). It should be noted that in JADE the probability of a mutation and crossover are adaptive parameters. However, at any iteration the mutation probability was high while the crossover probability was low for all the tasks. However, the experiments [13] showed that the crossover operator is a crucial step for the global search. This emphasizes a poor adaptation scheme for the crossover operator.

The obtained docking results are shown in Figure 2. As can be seen, the JADE outperforms other algorithms in two cases and achieves a satisfactory sub-angstrom precision. In case of the 1JWG complex JADE outperforms FPD2, which correspond to blind docking with a random start position. The set of FPD values is achieved with similar to JADE run time. The error is specified in Angstroms.

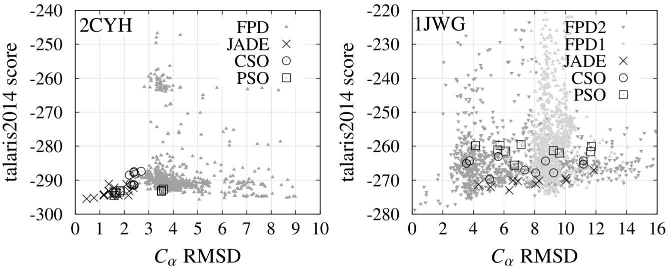

Figure 2. Energy score against alpha carbon Root-Mean-Square Deviation from the native conformation

5 Conclusions

The results of this study show that JADE provides the best overall performance. However, it shows poor results even for search space of about 50 parameters. Unfortunately, it is hard to achieve much better performance due to the heuristic nature of the algorithm. It is important to note that in relevant PPeD tasks it is necessary to consider peptides with lengths of 10–15 amino acid. With prior knowledge of the binding area and peptide structure the number of parameters will grow up to 250–300. This makes impossible to use such algorithms with all-atom resolution. However, using other evolutionary approaches like estimation of distribution algorithms [14], which is presently the current cutting edge, can show better performance. This will be the subject of future studies.

References

[1] R. Rentzsch and B.Y. Renard, Briefings in Bioinformatics **16**, 1045–1056 (2015)

[2] E. Lopez-Camacho, M.J. Garcia Godoy et al., Applied Soft Computing **28**, 379–393 (2015)

[3] C. Anfinsen, Science **181**, 330–331 (1973)

[4] R.F. Alford et al., Journal of Chemical Theory and Computation **13**, 3031–3048 (2017)

[5] B.J. Hintze et al., Proteins: Structure, Function, and Bioinformatics **84**, 1177–1189 (2016)

[6] M. Shapovalov and R.L. Dunbrack, Structure **19**, 844–858 (2011)

[7] R. Cheng and Y. Jin, IEEE Transactions on Cybernetics **45**, 191–204 (2015)

[8] M. Clerc and J. Kennedy, IEEE Transactions on Evolutionary Computation **6**, 58–73 (2002)

[9] J. Zhang and A. Sanderson, IEEE Transactions on Evolutionary Computation **13**, 945–958 (2009)

[10] B. Raveh, N. London, L. Zimmerman, and O. Schueler-Furman, PLoS ONE **6**, (2011)

[11] M. Kurcinski, M. Jamroz et al., Nucleic Acids Research **43**, 419–424 (2015)

[12] S.J. de Vries, J. Rey et al., Nucleic Acids Research **45**, 361–364 (2017)

[13] S.V. Poluyan, N.M. Reinhard, and N.M. Ershov, Vestnik Rossijskogo universiteta druzhby narodov, seriia: Matematika. Informatika. Fizika **2**, 415–418 (2014)

[14] B. Moradabadi and H. Beigy, Genetic Programming and Evolvable Machines **15**, 169–193 (2014)