USING OF MULTIVARIATE QUANTILE FUNCTION FOR SOLVING BIOINFORMATICS PROBLEMS

S.V. Poluyan [1, a], N.M. Ershov [2, b]

[1] *Dubna State University, 19 Universitetskaya, Dubna, Moscow region, 141982, Russia*

[2] *Lomonosov Moscow State University, the Faculty of Computational Mathematics and Cybernetics MSU, Faculty of Computational Mathematics and Cybernetics, Russia, 119991, Moscow, GSP-1, 1-52, Leninskiye Gory*

E-mail: [a] svpoluyan@gmail.com, [b] ershov@cs.msu.ru

In this work we study the evolutionary optimization algorithms for solving the problems in structural bioinformatics: protein-peptide docking and prediction of three-dimensional peptide structure from amino acid sequence. We describe the main assumptions that reduce these tasks to the continuous global optimization problems. Some special features of the given problem and the difficulties of using evolutionary algorithms are discussed. We propose a way of using evolutionary optimization algorithms based on using grid-based empirical quantile function. The paper describes used schemes for building and using of the quantile function. We describe used scheme for parallel sampling based on flood fill algorithm. The GPU-accelerated approach for quantile function evaluation and the resulting speed-up is presented. We made a comparison with the relevant docking method within a particular force-field and present the results of the experiments.

Keywords: global optimization, evolutionary algorithms, empirical quantile function, docking, flood fill algorithm, parallel computing

1. Introduction

In this study we focus on two problems in structural bioinformatics: prediction of three-dimensional peptide structure from amino acid sequence and protein-peptide docking. The current approaches to peptide structure prediction and protein-peptide docking are based on Anfinsen's hypothesis [1] which demonstrates that native-like conformations represent unique, low-energy, thermodynamically stable conformations. Therefore, the peptide structure prediction and protein-peptide docking can be considered as global optimization problems where the objective is to find the conformation with the lowest energy. The optimization problem for protein-peptide docking is formulated as the minimization of the binding energy.

Problems solving typically involves 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. The main motivation of this study is to create a platform base for comparison of different evolutionary algorithms within a certain force-field. The Rosetta [2] framework was used for full-atom complex structure representation and scoring (energy evaluation). A detailed description of solution encoding for given problems and Rosetta force-field can be found in [2, 3].

Here we focused on a protein-peptide docking with the following features. Firstly, the peptide binding site is known. Secondly, the protein interface has a linear structure (e.g. 2-Helix channel). Thirdly, peptide structure is also linear. The set of these properties is shown in Figure 1. The main feature of this task is to demonstrate that there is no need in searching of peptide beyond the borders of a particular spot (restricted by two spheres) and looking over different peptide conformations. There is also another condition since we study evolutionary optimization algorithms. We must save simple box constraints for continuous search space (i.e. lower and upper limit for each component).

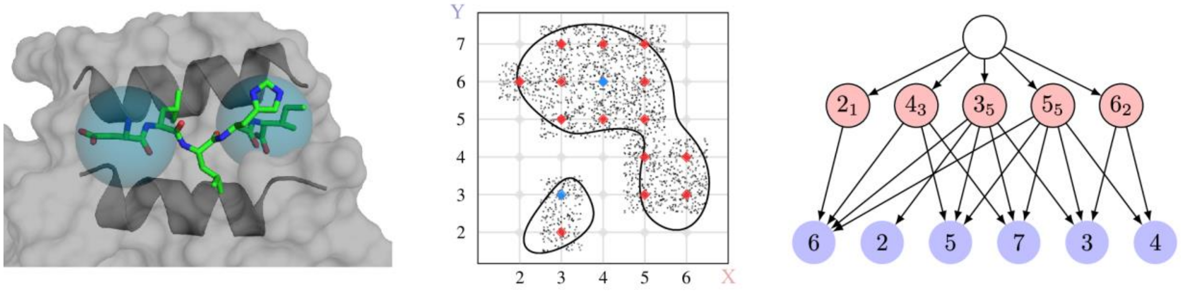

Figure 1. Protein interface and peptide structure (PDB: 1JWG). Search area, sample points and sampled values (black) after quantile transform. Trie-based structure for implicit sample storage

We propose an approach that takes the conditions above into account and avoids rough penalty method. The main idea of this approach is to transform components values responsible for peptide backbone dihedral angles, translation, and rotation by multivariate empirical quantile function [4] into values that correspond to the position of the peptide at the proper binding site. This kind of transformation involves two steps. First of all, we need to know the proper values of each used component, in other words, "sample". Secondly, we need to transform values from unit hypercube to proper component values with a certain procedure that works on a given sample. Thus, the empirical quantile d-dimensional function is $F:[0,1]^d \to \mathbf{R}^d$ which may be defined recursively by using univariate quantile transform [4].

2. Multidimensional Flood Fill Algorithm

As illustrated in the example in Figure 1, it is possible to cover the continuous area by the sample (red and blue points) based on the regular grid. Since we know the binding site it is possible to generate a sample that includes all peptide conformations within a restricted search area.

The sampling procedure is the following. At the first step we place peptide in a search area and determine the first initial sample point that based on a regular grid for all selected components. For instance, there are two initial points (the blue one) in the example in Figure 1. Then we perform Flood fill [5] procedure and determine all possible peptide conformations within a connected area. For this purpose we implement modified parallel queue-based multidimensional Flood fill algorithm with Von Neumann and Moore neighborhood. There are neighborhood points for each sample point starting from initial. Some points may and some may not define the components that place peptide in a proper position. The main idea of neighborhood processing with speed-up is presented in Figure 2. It is important to note that when we check peptide position we do not evaluate energy. Therefore, the check is fast. We use Von Neumann neighborhood (2·*d* neighboring points) for components and Moore neighborhood (3^d-1 neighboring points) for the translation and rotation components.

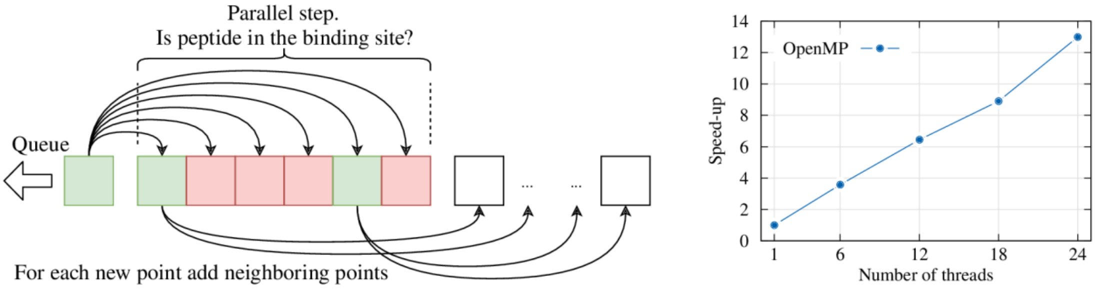

Figure 2. The part of the parallel Flood fill algorithm. The resulting speed-up

3. Multivariate Empirical Quantile Transform

Let suppose we have a sample in explicit (real-valued) form stored in the *n*×*d* matrix, where *n* is the sample size and *d* is the dimension of each vector. We are using multivariate empirical quantile function that is defined [4] recursively. Thus, for a given *d*-dimensional vector from $[0,1]^d$ it is necessary to go *d*-times through the entire sample for grid-based univariate quantile transform. At this procedure it is necessary to compare each sample vector with upper and lower bound vectors that obtained on each transform iteration. It is possible to parallelize selecting procedure that depends on sample size. As illustrated in Figure 3, each component of the vector is compared with the boundary values. The additional matrix contains sign marks obtained after comparison. After sum reduction procedure it is possible to determine whether the current vector is suitable or not.

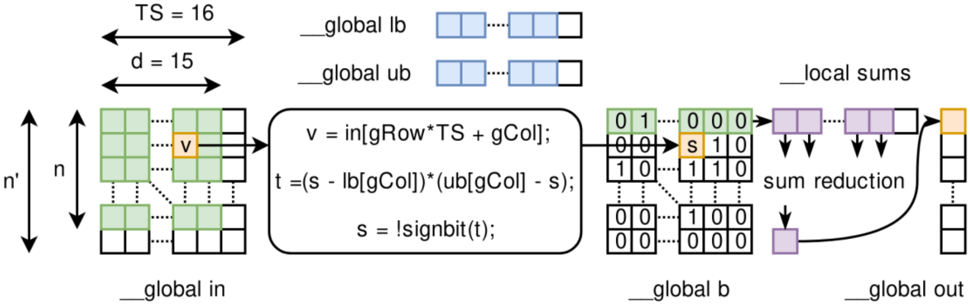
```
v = in[gRow*TS + gCol];
t =(s - lb[gCol])*(ub[gCol] - s);
s = !signbit(t);
```

Figure 3. The structure of the OpenCL procedure GPU2 for parallel data processing

In this work we implement multiple GPU-based approaches using the OpenCL framework with the following names: GPU1 for the procedure without the additional matrix, GPU2 for the procedure with additional matrix and sum reduction, GPU3 for the procedure with sum reduction and without the additional matrix. The obtained results for each approach are shown in Figure 4. The calculations were made with the usage of the Heterogeneous Platform HybriLIT [6] with NVIDIA Tesla K40s.

Since a sample is generated from grid values it is possible to store it in the implicit form at the trie-based structure as shown in Figure 1. The implicit form means that instead of the actual real value the position in the grid is stored. The trie-based structure presented in Figure 1 consists of nodes with a position in a grid. The node index indicates a number of sample vectors in a subtree. It is important to note, that the depth of the trie-based structure is equal to a vector dimension. The implicit univariate quantile transform procedure is similar to an explicit variant.

The multivariate empirical quantile transform allows us to create a continuous search space and use evolutionary algorithms for peptide structure prediction based on fragments. Fragments are short sequence segments that are generated from existing PDB structures. They are a core feature of the Rosetta prediction protocol and are used in the assembly of proteins. This approach substantially cuts down conformational search space.

4. Results and Discussion

In this study we perform docking experiments with three different methods. First is the parallel Adaptive Differential Evolution with Optional External Archive [7] (JADE). The choice of the evolutionary algorithm is founded on the previous studies [3]. The second is the qJADE algorithm which is the JADE algorithm with empirical quantile transform that handles peptide in the restricted binding area that illustrated in Figure 1. The third is the Rosetta FlexPepDock (FPD) [8] protocol from the Rosetta framework. It performs a high-resolution protein-peptide docking 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.

The experiments were done with 1JWG:B (Protein Data Bank id) protein-peptide complex with peptide DLLHI (FASTA format). The problem dimension is 54 parameters. The multivariate quantile transform is used for 15 components. The radius of each sphere illustrated in Figure 1 is equal to four angstrom. It should be noted that for qJADE in the experiments the sample size was about 75 million. Using one node at HybriLIT with two Intel Xeon twelve-core processors the calculations for parallel multidimensional Flood fill algorithm were made in about 5 hours.

The obtained docking results are shown in Figure 4. The set of FPD values is achieved with similar to JADE and qJADE run time. There were 10 independent runs for JADE and qJADE algorithms. The error is specified in Angstroms. Since the Rosetta energy function includes knowledge-based terms [2] energy score presented in Figure 4 has an indirect conversion to physical energy units like kcal/mol.

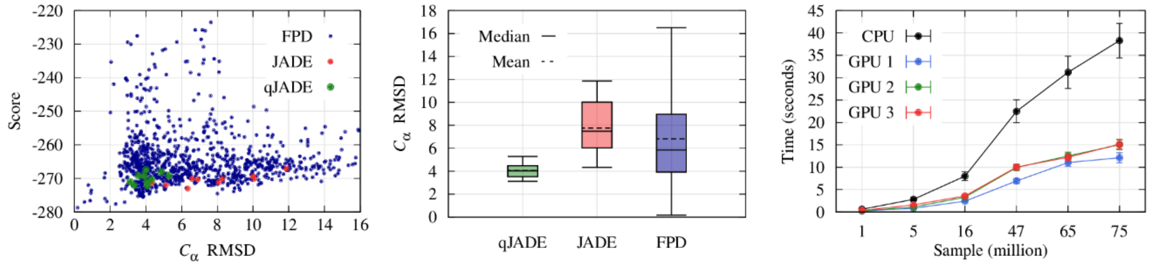

Figure 4. Energy score against alpha Carbon Root-Mean-Square Deviation from the native conformation. Boxplot for each method. Performance of the used schemes of parallel calculations

As it can be seen the FPD outperforms other approaches and achieves a satisfactory sub-angstrom precision. It should be noted that FPD protocol requires the initial starting position of the peptide. Here we considered near-native initial peptide state where a turnover along one axis relative to the native state in the binding spot was made.

The results of the experiments show that qJADE outperforms JADE. However, it shows poor results in comparison to FPD. This indicates the inability of the used evolutionary optimization algorithm to overcome the complex energy landscape.

5. Conclusion

The results of this study show that it is possible to reduce search space for peptide structure prediction and protein-peptide docking to a unit continuous hypercube by using quantile transform. This takes into account the remaining parameters, which undergo a linear interpolation conversion procedure. This formulation of the problem allows us to create a platform for an objective comparison of various global optimization algorithms.

In this study we proposed a grid-based approach for multivariate empirical quantile function and modified multidimensional Flood fill algorithm. We showed the performance of the quantile transform in an explicit form using the OpenCL framework and the speed-up of the parallel Flood fill algorithm. We presented a trie-based structure for implicit sample storage and a trie-based quantile function evaluation.

It is important to note that the proposed approach of using quantile function can be applied to a wide range of tasks with a similar formulation. Implementations of the multidimensional Flood fill algorithm and the multivariate empirical quantile function are available [9] on publicly accessible GitHub repositories.

References

[1] Rentzsch R. et al. Docking small peptides remains a great challenge: an assessment using AutoDock Vina // Briefings in Bioinformatics, 2015, vol. 16, No. 6, pp. 1045–1056. DOI: 10.1093/bib/bbv008.

[2] Alford R.F. et al. The Rosetta all-atom energy function for macromolecular modeling and design. 2017. DOI: 10.1101/106054.

[3] Poluyan S., Ershov N. Parallel evolutionary optimization algorithms for peptide-protein docking. EPJ Web of Conferences, 2018, vol. 173, pp. 06010–06010. DOI: 10.1051/epjconf/201817306010.

[4] Einmahl J.H.J. et al. Generalized Quantile Processes // The Annals of Statistics, 1992, vol. 20, No. 2, pp. 1062–1078. DOI: 10.1214/aos/1176348670.

[5] Vučković V. et al. Generalized N-way iterative scanline fill algorithm for real-time applications // Journal of Real-Time Image Processing, 2017, DOI: 10.1007/s11554-017-0732-1.

[6] Heterogeneous Platform HybriLIT. URL: http://hlit.jinr.ru/en (accessed: 05.11.2018).

[7] Zhang J. et al. JADE: Adaptive differential evolution with optional external archive // IEEE Transactions on Evolutionary Computation, 2009, vol. 13, No. 5, pp. 945–958. DOI: https://doi.org/10.1109/TEVC.2009.2014613.

[8] Raveh B. et al. Rosetta FlexPepDock ab-initio: Simultaneous Folding, Docking and Refinement of Peptides onto Their Receptors // PLoS ONE, 2011, vol. 6, No. 4. DOI: 10.1371/journal.pone.0018934.

[9] GitHub repositories. URL: https://github.com/poluyan (accessed: 05.11.2018).