数理モデル化と問題解決　39－6
（２００２．　５．１０）

効率的クリークアルゴリズムに基づく電気泳動画像マッチング、タンパク質構造マッチング、および、タンパク質側鎖パッキング

ケーシー　ドゥッカ　バハドゥル[1,2]　阿久津　達也[1,2]
富田　悦次[3]　関　友和[3]　藤山　秋佐夫[4]

[1] 京都大学 化学研究所 バイオインフォマティクスセンター
[2] 京都大学大学院 情報学研究科 知能情報学専攻
[3] 電気通信大学大学院　電気通信学研究科　電子情報学専攻　[4] 国立情報学研究所

電気泳動画像マッチング、タンパク質構造マッチング、タンパク質側鎖パッキングはいずれもバイオインフォマティクスにおいて有用なNP困難問題である。これらの問題に対して効率的な最大クリークアルゴリズムを用いた一連のアルゴリズムを開発し、計算機実験によりその性能を検証した。その結果。直接クリーク問題に変換するアルゴリズムを用いた場合は、50残基（点）から150残基（点）からなる入力に対して1 CPUで最適解を現実的な時間で計算することができた。また、いくつかのヒューリスティクスを組み合わせることにより、より現実的なサイズの問題に対して、良好な解を得ることができた。

Point matching under non-uniform distortions and protein side chain packing based on efficient maximum clique algorithms

Dukka Bahadur K. C.[1,2]　Tatsuya Akutsu[1,2]
Etsuji Tomita[3]　Tomokazu Seki[3]　Asao Fujiyama[4]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University,
Gokasho, Uji, Kyoto-Fu 611-0011, Japan.

[2]Graduate School of Informatics, Kyoto University, Sakyo-ku, Kyoto 606-8501, Japan.

[3]Graduate School of Electro-Communications, The University of Electro-Communications,
Chofu-city, Tokyo 182-8585, Japan.

[4]National Institute of Informatics, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.

{dukka,takutsu}@kuicr.kyoto-u.ac.jp　{tomita,seki-t}@ice.uec.ac.jp

We developed maximum clique-based algorithms for spot matching for two-dimensional gel electrophoresis images, protein structure alignment and protein side-chain packing, where these problems are known to be NP-hard. Algorithms based on direct reductions to the maximum clique can find optimal solutions for instances of size (the number of points or residues) up to 50∼150 using a standard PC. We developed pre-processing techniques to reduce the size of graphs. Combined with some heuristics, many realistic instances can be solved approximately.

1 Introduction

Many important computational problems in Bioinformatics are NP-hard. There are two major approaches to cope with these NP-hard problems. One approach is to use heuristic optimization techniques such as genetic algorithms, simulated annealing and neural networks. But, the crucial drawback of this approach is that an optimal solution is not guaranteed to be output. If an optimal solution is not guaranteed, one may doubt whether there is a much better solution. The other approach is to use the branch-and-bound technique. But, it is difficult to design branch-and-bound algorithms which can be applied to realistic size instances. Deep insight into the target problem is usually required in order to design an efficient branch-and-bound algorithm.

On the other hand, two of the authors have been developing efficient algorithms for the *maximum clique* problem [6, 7], where the maximum clique problem is a well-known NP-hard graph theoretic problem (the maximum clique problem is, given an undirected graph, to find a complete subgraph with the maximum number of vertices). Different from the standard approach based on IP (Integer Programming) [4], these algorithms employ very simple branch-and-bound techniques and are much more efficient than existing maximum clique algorithms [6]. Some of the algorithms can find the maximum cliques of a graph with more than 10,000 vertices if the graph is sparse. Some of the algorithms can find the maximum cliques of a graph with more than 1,000 vertices even if the graph is dense. Using these algorithms and reductions to the maximum clique problem, we develop algorithms for three important problems in Bioinformatics: *spot matching for two-dimensional gel electrophoresis images* [2], *protein structure alignment* [4] and *protein side-chain packing* [1].

Maximum clique algorithms have been already applied to Bioinformatics [3, 4] and pattern matching [5]. But, most of the previous methods have at least one of the following drawbacks: (i) algorithms can not be applied to realistic size instances [5], (ii) the original problem is too simplified by using heuristics so that (non-efficient) maximum clique algorithms can be applied [3], (iii) a domain-specific and complicated branch-and-bound procedure is required [4]. Our algorithms based on direct reductions can find optimal solutions for instances of size up to 50∼150 (spot matching for images with 100 spots, structure alignment for proteins with 50 residues and side-chain packing for proteins with 150 residues). To solve larger instances, we developed pre-processing techniques that reduce the size of graphs. Combined with some heuristics, most of realistic instances can be solved approximately, where details of the pre-processing techniques and heuristics are omitted in this article because of the page limit.

2 Algorithms for Point Matching

In this article, both spot matching and structure alignment are defined as the point matching problem under non-uniform distortions. Point matching under non-uniform distortions is defined as follows [2]. Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m\}$ and $Q = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$ be point sets in d-dimensions, respectively. For two points $\boldsymbol{x}$ and $\boldsymbol{y}$, $|\boldsymbol{x} - \boldsymbol{y}|$ denotes the Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{y}$. Let $f(x)$ be a function from the set of non-negative reals to the set of reals no less than 1.0. We call a set of pairs $M = \{(\boldsymbol{p}_{i_1}, \boldsymbol{q}_{j_1}), \ldots, (\boldsymbol{p}_{i_l}, \boldsymbol{q}_{j_l})\}$ a *matching* if $(\forall h \neq k)(\boldsymbol{p}_{i_h} \neq \boldsymbol{p}_{i_k}$ and $\boldsymbol{q}_{j_h} \neq \boldsymbol{q}_{j_k})$.

Then, *point matching under non-uniform distortions* is, given f, P and Q, to find a maximum matching M (i.e., a matching M with the maximum cardinality) satisfying

$$(\forall k)(\forall h \neq k)\left(\frac{1}{f(r)} < \frac{|\boldsymbol{q}_{j_h} - \boldsymbol{q}_{j_k}|}{|\boldsymbol{p}_{i_h} - \boldsymbol{p}_{i_k}|} < f(r) \right),$$

where $r = \min\{|\boldsymbol{q}_{j_h} - \boldsymbol{q}_{j_k}|, |\boldsymbol{p}_{i_h} - \boldsymbol{p}_{i_k}|\}$. It is proven in [2] that this problem is NP-hard for any $d \geq 2$ even if $f(x)$ is a constant.

It should be noted that P and Q can be exchanged in the above definition because $\frac{1}{f(r)} < \frac{x}{y} < f(r)$ if and only if $\frac{1}{f(r)} < \frac{y}{x} < f(r)$. It should also be noted that, if $f(r)$ is a constant close to 1.0, local similarity must be preserved because the error for two point pairs must be small if the distances between points in the pairs are small. But, global similarity need not be strictly preserved since the error can be large if the distances are large.

Reduction of the matching problem to the maximum clique problem is very simple (see Fig. 1). Basically, the definition of the problem itself gives the reduction. We construct a graph $G =$

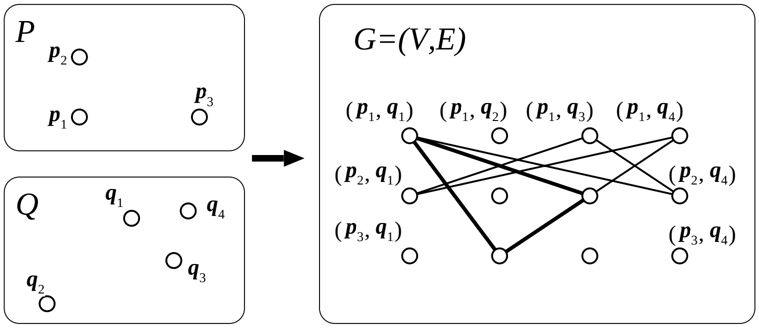

Figure 1: Reduction from point matching to maximum clique. The maximum clique $\{(\boldsymbol{p}_1, \boldsymbol{q}_1), (\boldsymbol{p}_2, \boldsymbol{q}_3), (\boldsymbol{p}_3, \boldsymbol{q}_2)\}$ of G corresponds to the maximum match between P and Q.

(V, E) by letting $V = \{(\boldsymbol{p}_i, \boldsymbol{q}_j) | i = 1, \ldots, m, j = 1, \ldots, n\}$ and letting $\{(\boldsymbol{p}_{i_h}, \boldsymbol{q}_{j_h}), (\boldsymbol{p}_{i_k}, \boldsymbol{q}_{j_k})\} \in E$ if and only if $\frac{1}{f(r)} < \frac{|\boldsymbol{q}_{j_h} - \boldsymbol{q}_{j_k}|}{|\boldsymbol{p}_{i_h} - \boldsymbol{p}_{i_k}|} < f(r)$ holds. Then, the maximum clique corresponds to the maximum point match. It should be noted that $O(mn)$ vertices. and $O(m^2 n^2)$ edges will be created by means of this reduction. However, this graph is usually sparse and thus the maximum clique algorithms can be directly applied if n, m are less than 50∼100.

3 An Algorithm for Protein Side-chain Packing

Application to side-chain packing is preliminary. We consider a simple geometric condition for side-chain packing [1]. We only mind whether or not each side chain collides with the main chain or the other side chains. In this case, we simply consider rotations (χ_1 angles) around the vector defined by Cα and Cβ atoms, though extensions to more general cases are straightforward. We use discrete rotation angles. Currently, the set of rotation angles is defined by $\{(2\pi k)/K | k = 0, \ldots, K-1\}$, where $K = 20$.

Let R be the set of residues. Each residue consists of positions of atoms in the side chain, where hydrogen atoms are ignored in the current implementation. Glycine residues are ignored and the positions of atoms in Proline residues are fixed in the current implementation. The positions of atoms in a side-chain are rotated around the χ_1 axis. Let $r_{i,k}$ be the i-th residue whose side-chain atoms are rotated by $(2\pi k)/K$ radian. We say that *residue $r_{i,k}$ collides with the main chain* if the minimum distance between the atoms in $r_{i,k}$ and the atoms in the main chain is less than $L_1 \mathring{A}$. We say that *residue $r_{i,k}$ collides with residue* $r_{j,h}$ if the minimum distance between the atoms in $r_{i,k}$ and the atoms in $r_{j,h}$ is less than $L_2 \mathring{A}$. We currently use $L_1 = 1.0$ and $L_2 = 4.0$. Then, graph $G = (V, E)$ is defined by $V = \{r_{i,k} |\ r_{i,k}$ does not collide with the main chain $\}$, $E = \{\{r_{i,k}, r_{j,h}\} |\ i \neq j,\ r_{i,k}$ does not collide with $r_{j,h}\}$. It is easy to see that a maximum clique of size n corresponds to a consistent side-chain configuration (i.e., a configuration in which any two atoms do not collide). It should be noted that the graphs created as above are usually dense because two (geometrically) remote residues do not collide.

4 Computational Experiments

We conducted computational experiments mainly for assessing the scalability of the algorithms, using real protein and electrophoresis image data. We used a standard PC with an AMD Athlon 700MHz CPU and 512MB main memory. In all cases of the experiments, MCQ [6] was used for

Table 1: CPU times for spot matching, where P and Q are input point sets.

$\|P\|$	$\|Q\|$	DMATCH	PMATCH
74	83	23.0 sec.	8.5 sec.
97	108	71.8 sec.	39.7 sec.
134	152	N/A	334.6 sec.
215	228	N/A	1041.7 sec.

Table 2: CPU times for side-chain packing.

PDB code	#residues	CPU time (sec.)
3fxc	98	12.2
5cpv	108	15.5
4hhb(A)	141	32.7
4i1b	151	53.7

finding maximum cliques. Because of the page limit, we only show the results on spot matching and side-chain packing.

For spot matching, we examined the CPU time (sec.) of the algorithm without preprocessing (denoted by DMATCH) and the algorithm with preprocessing (denoted by PMATCH) varying the number of points in a point set, where we used subsets of the original point sets derived from real electrophoresis images [2]. Table 1 shows the results. In Table 1, N/A means that matching could not be found because of memory allocation error. PMATCH found the maximum matches in the first three cases, but it is unclear whether or not the computed clique was the maximum in the last case (PMATCH does not necessarily output an optimal solution if the size of the optimal solution is not large enough).

We conducted a preliminary computational experiment on side-chain packing, where we used PDB (Protein Data Bank) data. Table 2 shows CPU times required for side-chain packing for several protein data.

These results suggest that maximum clique-based algorithms may be useful for solving realistic problems in Bioinformatics.

References

[1] Akutsu, T.: NP-Hardness results for protein side-chain packing, *Genome Informatics*, **8**, 180-186, 1997.

[2] Akutsu, T., *et al.*: Matching of spots in 2D electrophoresis images. Point matching under non-uniform distortions, *LNCS*, **1645**, 212-222, 1999.

[3] Kato, H. and Takahashi, Y.: SS3D-P2: a three-dimensional substructure search program for protein motifs based on secondary structure elements, *CABIOS*, **13**, 593-600, 1998.

[4] Lancia, G., *et al.*: 101 optimal PDB structure alignment: a branch-and-cut algorithm for the maximum contact map overlap problem, *Proc. RECOMB 2001*, 193-202, 2001.

[5] Ogawa, H.: Labeled point pattern matching by Delaunay triangulation and maximal cliques, *Pattern Recognition*, **19**, 35-40, 1986.

[6] Seki, T. and Tomita, E.: Efficient branch-and-bound algorithms for finding a maximum clique, *Technical Report of IEICE*, **COMP2001-50**, 101-108, 2001.

[7] Tomita, E., Kohata, Y. and Takahashi, H.: A simple algorithm for finding a maximum clique, *Technical Report*, **UEC-TR-C5**, The University of Electro-Communications, 1988.