ラベル情報と構造情報の相関を用いた三次元点集合マッチングの高速化とそのインフルエンザウイルス解析への応用

佐々木耀一† 渋谷 哲朗†† 大森 亮介††† 伊藤 公人††† 有村 博紀†

† 北海道大学大学院情報科学研究科 〒 060–0814 札幌市北区北 14 条西 9 丁目
†† 東京大学医科学研究所ヒトゲノム解析センター 〒 108–8639 東京都港区白金台 4-6-1
††† 北海道大学人獣共通感染症リサーチセンター 〒 001–0020 札幌市北区北 20 条西 10 丁目
E-mail: †{ysasaki,arim}@ist.hokudai.ac.jp, ††tshibuya@hgc.jp, †††{omori,itok}@czc.hokudai.ac.jp

あらまし 本稿では頂点ラベル付き三次元点集合の照合アルゴリズムを考察する．先行研究において，ラベルなしの点集合データに対して，我々は可能なマッチングの全列挙とスコアの下限見積もりを用いた照合アルゴリズムを提案した． また，タンパク質データなど，生物分野の三次元点集合データでは ラベル情報に基づく距離 (正規化ハミング距離) と構造情報に基づく距離 (RMSD) の間に近似的な相関関係があることが経験的に知られている．本稿では，このような相関関係を用いてラベル情報に基づく距離の下限見積もりを用いた高速化手法を提案する．実験では，インフルエンザウイルスデータを対象として，実際のデータから上記の下限関数を経験的に求め，さらにその有効性を検証した．

キーワード パターン照合，分枝限定法，列挙アルゴリズム，RMSD, 最小二乗和平均平方根距離，正規化ハミング距離，点集合マッチング

1. はじめに

1.1 背　景.

近年，タンパク質や高分子の三次元構造の高速な検索は重要な問題になっている．例えば，あるタンパク質の構造が与えられているが，その性質がわかっていないときに，既に構造と性質が知られているタンパク質との構造的に類似しているかを調べることにより，性質を予想することが行われている [1].

我々は，タンパク質の構造を三次元空間に分布する点集合ととらえて，二つの点集合 P と T の間の三次元点集合マッチング問題を考察し，高速なアルゴリズムを提案した [10] [9]．このアルゴリズムは，点集合の構造の類似距離には平行移動と回転に関する最小二乗和平方距離（RMSD）を採用し，その上限推定を用いた枝刈りにより，高速な照合を実現する．

1.2 本研究の目的.

本稿では，点の座標以外の情報を利用して，RMSD の三次元点集合マッチングを高速化することを考える．近年，分子生物学分野では，二つのタンパク質の間で，それらの RMSD 値と DNA 配列の文字列距離に密接な関係があることが指摘されている [11]．そこで，我々は，この RMSD と文字列距離の関係を用いて，三次元点集合マッチングの高速化を考える．初めに，実際のインフルエンザウイルスのタンパク質データベースを用いて，上記の RMSD と文字列距離の相関関係を実験的に調べる．次に，観察した相関関係に基づいて経験的枝刈り関数 f を定める．最後に，我々の照合アルゴリズムに枝刈り関数を組み込み，提案手法による高速化の有用性を調べる．

1.3 関 連 研 究

本論文の関連研究として，Schwartz と Sharir [3] はデータベースとパターンを共に点列の入力とし，RMSD 指標によるマッチングを考察し，データベースサイズを N としたときに $O(N \log N)$ 時間で実行するアルゴリズムを提案している．また，Shibuya [2] は，同じマッチング問題を，データベースサイズに対して線形時間で実行するアルゴリズムを提案している．その他に，Akutsu [4] らは，ハッシュを用いた手法を提案しており，$O(N)$ 期待計算時間を実現している．また，Shibuya [1] は，幾何接尾辞木 (geometric suffix tree) と呼ばれる，三次元構造に対する索引データ構造を示している．これは，文字列に対する索引構造である接尾辞木 (suffix tree) [6], [7] を基に，タンパク質のような複雑な三次元構造に対して適応させたデータ構造である．このデータ構造は，RMSD 指標による効率よい近似部分構造マッチングを実現し，データベース中に存在するタンパク質構造同士の頻出する類似部分構造の発見にも用いることができる．

また，三次元点集合マッチング問題に対して，分枝限定法を用いた実際的に高速なアルゴリズムが提案されている. [9] [10]

2. 準　備

2.1 基本的定義

本稿では，三次元空間 $\mathbb{R}^3$ を考える．まずはじめに，長さ n の**点列**（point sequence）S は，$S = (s_1, \ldots, s_n) \in (\mathbb{R}^3)^n$ である．ここに，各 $i = 1, \ldots, n$ に対して，$s_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ は i 番目の点の三次元座標を表すベクトルである．ここに，$|S| = n$ で S の**長さ**を表す．任意の $1 \leqq i \leqq j \leqq n$ に対して，$P[i \ldots j] = (s_i, s_{i+1}, \ldots, s_j)$ で S の添え字 i から j までの**部分点列**を表す．点列 S の長さ

$1 \leqq i \leqq n$ の**前綴り**（prefix）とは，最初の i 個からなる列 $S[1..i]$ である．

三次元の回転行列 R によって点列 S を回転した構造は $R \cdot S = (Rs_1, \ldots, Rs_n)$ で与えられる．回転行列 R と移動ベクトル v は三次元空間の点に対する**変換** $f = f_{R,v}(S) = R \cdot S + v$ を与える．ベクトル v の**転置ベクトル**を v^t で表し，行列 A の**転置行列**を A^t で表す．$trace(A)$ は行列 A の**トレース**を与え，$|v|$ はベクトル v の**ノルム**を与える．

次に，長さ n の**点集合**（point set）T とは，$T = T[1, \ldots, n] = T[1] \cdots T[n] = \{t_1, \ldots, t_n\} \in (\mathbb{R}^3)^n$ である．ここに，各 $P[k] = (x_k, y_k, z_k) \in \mathbb{R}^3$ $(k \in [1..k])$ は $\mathbb{R}^3$ の点であり，点の順序は任意に決めておく．

また，長さ n の**ラベル付き点集合** A とは，$A = \{a_1, \ldots, a_n\}$ で，各点 $a_k = (x_k, y_k, z_k, r_k)$ $(k \in [1..k])$ となり，$x_k, y_k, z_k \in (\mathbb{R})$ で,x 座標，y 座標，z 座標を表し，$r_k \in \Sigma$ でラベル (文字) も保持している．

2.2 二つの点集合間の最小 RMSD

本小節では，本稿で問題としている，二つの点集合間の類似度として用いられる**最小平方平均二乗和距離**（最小 RMSD, MinRMSD）を定義する．$P = P[1..n]$ と $Q = Q[1..n]$ を，要素数 n の三次元空間の点集合とする．ただし，これらの点集合は，点列として表されているものとし，要素である点の順番は任意に固定するとする．

順列 $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ に対して，π による長さ n の点列 $P = P[1..n]$ の並べ替えを $\pi(P) = (P[\pi(1)], \ldots, P[\pi(n)])$ と定める．

点列として表されたサイズ n の集合 P と Q 間の最小平方平均二乗和距離を，すべての順列 π と，回転行列 R，移動ベクトル v に関する最小の RMSD 値

$$MinRMSD(P, Q) = \min_{\pi} \min_{R,v} RMSD_{R,v}(\pi(P), Q) \qquad (1)$$

と定義する．すなわち，$MinRMSD(P, Q)$ は，全ての並べ替えに関して点列 P と Q の最小 RMSD スコアをとった最小値である．

2.3 最小 RMSD に関するラベル付き近似点集合マッチング問題

本論文で考察する最小 RMSD スコアに関するラベル付き近似点集合マッチング問題を次のように定義する．三次元空間を考える．

データ点集合とは，n 個の点からなる点集合 $T = \{t_1, \ldots, t_n\}$ である．**パターン点集合**とは，k 個の点からなる点集合 $P = \{p_1, \ldots, p_k\}$ である．T と P の要素を，それぞれ，データ点およびパターン点と呼ぶ．また各点は，三次元の座標とラベルを保持している．ただし，以下では常に $k \leqq n$ であり，データ点集合とパターン点集合は，点列として表されるものと仮定する．

非負実数 $r > 0$ に対して，パターン点集合 P がデータ点集合 T に**最小 RMSD スコア** r **でマッチする**とは，k 個のデータ点からなるある点集合 $Q \subseteq\!\!\!=\, T$ が存在して，

$$MinRMSD(P, Q) \leqq r$$

が成立することである．このとき，T の部分点集合 Q を，T における P **の出現位置**という．

［定義 2.1］ **最小 RMSD スコアに関する点集合マッチング探索問題**は，データ点集合 T と，パターン点集合 P，非負実数 $r > 0$ を受け取り，T 上の最小 RMSD スコア r でのパターン P の出現位置 $Q \subseteq\!\!\!=\, T$ を一つ見つける問題である．

上記の問題の変種として，P が T に最小 RMSD スコア r でマッチするかどうかを YES または NO で答えるものを**点集合マッチングの決定問題**と呼ぶ．また，全ての出現位置 $Q \subseteq\!\!\!=\, T$ をもれなく，かつ重複なしに見つけるものを**点集合マッチングの列挙問題**と呼ぶ．ただし，点列として異なる出現位置 Q は互いに異なるとみなす．

本稿では，とくに断らなければ，点集合マッチングの列挙版問題を考察する．決定版と探索版の問題は，列挙版のアルゴリズムで解くことができる．

2.4 ラベル情報を用いない三次元点集合マッチング問題に対する基本アルゴリズム

本小節では，我々が考案したラベル情報を用いない三次元点集合マッチングのアルゴリズム [9] [10] を紹介する．このアルゴリズムは，最小 RMSD スコアの下界関数を用いた探索の枝刈りによる効率化を行う．

まず初めに，素朴なアルゴリズムとして，再帰的な計算により解となる候補集合を探索し，探索木の葉，すなわち，候補集合のサイズがパターンサイズと同じ k に達したときに初めて，最小 RMSD スコアによる解の検査を行う手順が考えられる．しかし，探索のすべての枝を葉までたどるので効率が悪い．

これを改良するために，前綴りに関する最小 RMSD スコアの下界関数（lower-bound function of minimum RMSD for prefices）を導入し，これを用いた分枝限定法（branch-and-bound method）を取り入れている．

［定義 2.2］（MinRMSD の下界関数） 任意の正整数 $1 \leqq i \leqq k$ に対して，下界関数を次のように定める：

$$LB\_MinRMSD_{i,k}(P[i], Q[i]) = \left(\frac{i}{k}\right)^{1/2} MinRMSD(P[i], Q[i]). \qquad (2)$$

Algorithm 1 に，下界関数による分枝限定を用いた点集合マッチングアルゴリズムを示す．このアルゴリズムは，再帰に基づいて，すべての候補集合を列挙し，最小 RMSD スコアが閾値 r 以下の解を正しく出力する．17 行目から 19 行目で，下界関数を用いた枝刈りを行う．この枝刈りは健全であり，いかなる正解を見落とすこともないことが言える．このアルゴリズムについて次が示されている．

［定理 2.1］ 点集合マッチングアルゴリズム Algorithm 1 は，与えられたデータ点集合 T とパターン点集合 P に対して，最小 RMSD スコアが閾値 r 以下となる P のすべての出現位置を正しく見つける．

アルゴリズムの使用領域は $O(k)$ 領域である．計算時間に関しては，総計算時間が $O(kn^k)$ 時間であること以外はわからない．ただし，素朴なアルゴリズムが入力に無関係に常に $\Theta(kn^k)$ の時間を要するのに対して，枝刈りアルゴリズムは早期終了の

Algorithm 1 ラベル情報を用いない点集合マッチングの基本アルゴリズム BasicMatch

procedure BasicMatch(P, T, r)
入力： パターン点集合 $P[1..k]$，データ点集合 $T[1..k]$，正実数 r.
出力： T 中の最小 RMSD スコア r に関する P の全ての異なる出現位置.
    BasicFind($\varepsilon, 0, P, T, |Q|, |T|$) を呼び出す;
end procedure

procedure BasicFind($Q[1..i], i, P, T, k, n$)
    if $i = k$ **then** ▷ すべての点を選び終わった
        if $MinRMSD(P[1..k], Q[1..k]) \leqq r$ **then**
            マッチング位置の集合 Q を出力する;
        end if
    else
        for $j = 1, \ldots, n$ **do**
            $x = T[j]$;
            if $x \in Q$ **then**
                continue
            end if
            $R := Q$ の末尾に要素 x を加えて得られる点列;
            if $LB\_MinRMSD_{i+1,k}(R, P[1..i+1]) > r$ **then**
                continue
            end if
            BasicFind($R, i+1, P, T, k, n$) を再帰的に呼び出す;
        end for
    end if
end procedure

可能性をもっている点で実際には利点がある.

3. 提 案 手 法

本節では，実データへの応用を考え，具体的にはインフルエンザウイルスにおける類似構造を見つけるという問題に，ラベル付き近似点集合マッチング問題を適応する．この問題において，各点の持つラベル情報はタンパク質におけるアミノ酸の種類に対応させる．生物学で一般的に知られている RMSD と DNA 配列の文字列距離の間の関係 [11] を利用した枝刈り手法を定式化し，これを用いて前節で説明した点集合マッチング問題の高速化を考える．提案手法では，次の文字列距離を用いる．

［定義 3.1］ 点集合 P と Q の間の順列 π の下での**正規化ハミング距離**（NHD）を，対応する点のラベルの不一致数を長さで正規化したもの

$$NHD_\pi(P, Q) = NHD(\pi(P), Q) \tag{3}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \Psi\left[\, \pi(P)[i] \neq Q[i] \,\right] \in [0, 1] \tag{4}$$

と定める．ここに，$\Psi[\cdot]$ は特性関数である.

RMSD による NHD の上限関数（または**上限関数**）とは，任意の実数関数 $f : [0, \infty) \to [0, 1]$ である．多くの点集合の対 $\tau = (P, Q)$ に対して，関係

$$RMSD(\pi(P), Q) \leqq r \Rightarrow NHD(\pi(P), Q) \leqq d = f(r) \tag{5}$$

が成立するならば，与えられた RMSD 値の上限 r に対して NHD 値の上限 $d = f(r)$ を用いて，再帰的なマッチングアルゴリズムの繰り返しにおいて，第 i 番目の点対に対して，制約違反 $NHD(P[\pi(1)] \cdots P[\pi(i)]], Q[1..i]) > d$ が成立したとき，これ以上の探索を行わないという枝刈りが行える．一般には，対上のある同時確率分布 D に従う点集合の対 $(P, Q) \sim D$ に対して高い確率で式 (5) が成立するならば，一定の照合ミスを許した上で，照合の高速化を期待できる．上記のアルゴリズムの擬似コードを 2 に示す.

Algorithm 2 NHD の枝刈りに基づくラベル付き点集合マッチング問題の提案アルゴリズム

procedure Main(P, T, r)
入力： パターン点集合 $P[1..k]$，データ点集合 $T[1..k]$，正実数 r.
出力： T 中の最小 RMSD スコア r に関する P の全ての異なる出現位置.
    Find($\varepsilon, 0, P, T, |Q|, |T|$) を呼び出す;
end procedure

procedure Find($Q[1..i], i, P, T, k, n$)
    if $i = k$ **then** ▷ すべての点を選び終わった
        if $MinRMSD(P[1..k], Q[1..k]) \leqq r$ **then**
            マッチング位置の集合 Q を出力する;
        end if
    else
        for $j = 1, \ldots, n$ **do**
            $x = T[j]$;
            if $x \in Q$ **then**
                continue
            end if
            $R := Q$ の末尾に要素 x を加えて得られる点列;
            if $NHD(R, P[1..]) \leqq d$ **then**
                ▷ NHD による新しい枝刈り
                continue
            end if
            if $UB\_MinRMSD_{i+1,k}(R, P[1..i+1]) > r$ **then**
                continue
            end if
            Find($R, i+1, P, T, k, n$) を再帰的に呼び出す;
        end for
    end if
end procedure

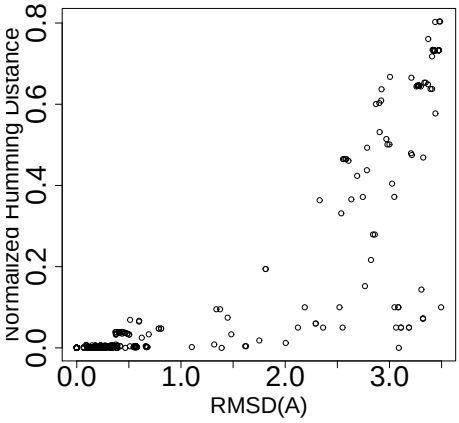

図 1　実験 1：RMSD と NHD の散布図 ($RMSD \leqq 3.5$(Å))

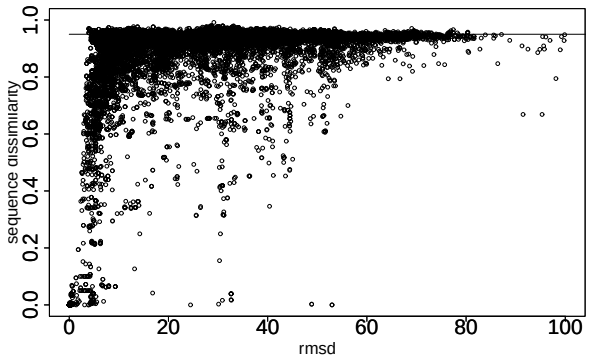

図 2　実験 2：RMSD と NHD の散布図 ($RMSD \geqq 3.5$(Å))

r (A)	0.0~0.1	0.1~0.3	0.3~0.5	0.5~2.0	2.0~3.0	3.0~
d	0.003	0.006	0.04	0.2	0.8	0.1

表 1　今回用いた経験的な上限関数 $d = f(r)$

4. 実　　験

実験 1. 実験 1 では，PDB database 内にある 342 種類のインフルエンザウイルスのヘマグルチニンと呼ばれるタンパク質データに対して，それらの RMSD が 3.5（Å）以下となる組み合わせに対して，RMSD と HND の値を計測した．これより，r の上限と d の上限の間にわれわれが主張するような上記の関係があるかどうかを調べた．

実験 2. 実験 2 では，実験 1 と同様のデータに対して，RMSD が 3.5(Å) 以上では上記のような RMSD と NHD の関係は観察されないことを確認した．

実験 3. 実験 1 の結果のグラフから，経験的な枝刈り関数 f を求めた (表 1)．また，この関数 f を用いて，PDB database(注∗) の一組のインフルエンザウイルスのヘマグルチニン (4bgw.pdb(点の数=486) VS 4bgx.pdb(点の数=485)) に対して，f による枝刈りの正しさを確認した．

(注∗) http://www.rcsb.org/pdb/home/home.do

RMSD r	0.1		0.15		0.2		0.25	
手法	照合	時間	照合	時間	照合	時間	照合	時間
提案手法	0	25	4	166	4	368	4	1532
全解出力	0	67	4	301	4	1463	4	9341

表 2　実験 3．各 RMSD 値毎に，見つけた照合数（照合）と要した計算時間（時間，単位 sec) を示す．

5. 結　　論

本稿では，3 次元空間でのラベル付き点集合マッチングを考察し，RMSD と文字列距離である NHD の関係から考えた関数を用いて枝刈りによる高速化を行うアルゴリズムについて議論した．

実験では，RMSD 値が小さい時に RMSD と NHD に我々の指摘した関係があることを観察した．さらに，観察した相関関係に基づいて経験的枝刈りを定めて，f による枝刈りが安全に行われていることを確認した．最後に，実験より r が小さい時 ($r \leqq 0.25$) において，3 倍から 5 倍程度高速化していることが確認できた．

今後の課題として，実験 3 で与えた枝刈り関数の損失率を定式化し，枝刈り関数のクラスを設計して，要求する損失率に対して，適切な枝刈り関数を与える．

文　　献

[1] Tetsuo Shibuya, ”Geometric Suffix Tree : Indexing Protein 3-D Structures,” Jounal of the ACM, Vol.57, No.3, Article 15, 2010.

[2] Tetsuo Shibuya, ”Searching Protein 3-D Structures in Linear Time,” RECOMB 2009, LNCS 5541, 115, 2009.

[3] Jacob T. Schwartz, Micha Sharir, ”Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves,” Intl. J. of Robotics Res. 6, 2944, 1987.

[4] Tatsuya Akutsu, Kentaro Onizuka, Masato Ishikawa, ”New Hashing Techniques and Their Application to a Protein Strucure Database System,” Proc. Hawaii Int. Conf. System Sciences 5, 197-206, 1995.

[5] Gilbert Strang, ”Introduction to Linear Algebra, 3rd Edition,” Wellesley-Cambridge Press, 2003.

[6] Dan Gusfield, ”Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology,” Cambridge University Press, 1997.

[7] Esko Ukkonen, ”On-line construction of suffix-trees,” Algorithmica, Vol. 13, No. 3, 249-260, 1995.

[8] Mark Pinsky and Samuel Karlin, ”An introduction to stochastic modeling,” Academic press, 2010.

[9] 佐々木 耀一，渋谷 哲朗，伊藤 公人，有村 博紀，”三次元空間における効率良い近似点集合マッチングと分子パターン照合への応用,” 第 42 回バイオ情報学研究会, 2015.

[10] Yoichi Sasaki, Tetsuo Shibuya, Kimihito Ito, Hiroki Arimura, ”Efficient Approximate 3-Dimensional Point Set Matching Using Root-Mean-Square Deviation Score,” SISAP 2015, 191-203, 2015.

[11] Aneerban Bhattacharya, et al. ”Assessing model accuracy using the homology modeling automatically software”, Proteins: Structure, Function, and Bioinformatics, Volume 70, Issue 1, Pages 105118, 2008