複数の蛋白質立体構造データに対するディスク版索引構造の構築方式

高橋 誉文† 黒木進† 田村慶一† 北上始†

†広島市立大学大学院情報科学研究科 〒731-3194 広島市安佐南区大塚東三丁目 4 番 1 号

E-mail: †takahashi@db.its.hiroshima-cu.ac.jp, {kuroki, ktamura, kitakami}@hiroshima-cu.co.jp

あらまし 立体構造が類似する蛋白質どうしは，性質がお互いに類似している．この事情により，複数の配列データ（蛋白質立体構造データ）から性質が類似する蛋白質を高速検索する方法として，索引構造の研究が重要である．しかしながら，従来提案されている索引構造は，1 件の配列データにだけ対処可能な幾何学的なサフィックス木であるため，複数の配列データに使用できないという問題がある．また，それは，メモリ上でのみ動作することを前提としているため，大規模な配列データにも使用できないという問題がある．本稿では，これらの 2 つの問題を解決するために，幾何学的なサフィックス木を拡張し，複数の配列データに対処可能な索引構造をディスク上に構築する方法について提案する．また，この提案方法の特性を確認するために，蛋白質立体構造データベース PDB を用いて，構築性能や検索性能などを評価したので，それらの結果についても報告する．

キーワード データマイニング，バイオインフォマティクス，空間・時空間データベース，情報検索

1. はじめに

蛋白質構造ではアミノ酸配列が異なっていても，3 次元構造が似ているとその蛋白質同士は類似した性質を持つと言われている．現在，類似する蛋白質を発見する方法は，2 つの部分構造を 1 対 1 で比較することにより行われている．しかし，この方法では多くの組み合わせを試す必要があり，多くの時間がかかる．さらに，このような類似構造を蛋白質構造データベースから高速に類似部分構造を検索する方法の研究[8][9]は十分に行われていない．現在までに，サフィックス木を応用して，蛋白質構造データに対する索引構造を構築する方法の研究が行われている[1][2]．これらには，蛋白質の立体構造を表現する座標配列を記号化する方法[1]と座標配列を直接利用する方法[2]がある．しかしながら，どちらの方法においても，1 本の座標配列に対する索引構造の研究にとどまっていると同時にディスク上で管理することを想定していない．本稿では，蛋白質立体構造を表現する座標配列を直接利用する方法と DynaCluster と呼ばれるディスク上のサフィックス木構築法[3][4]とを用いて，複数本の蛋白質立体構造データに対するディスク版サフィックス木を構築する方法について提案する．

以下，本稿の構成を示す．2 章では従来手法として，幾何学的なサフィックス木について述べる．3 章では，提案手法について述べる．4 章では提案手法の有効性を示すための実験による評価を行い，5 章では本稿のまとめを行う．

2. 従来手法

現在提案されている手法は，サフィックス木を拡張して座標配列でサフィックス木を構築する幾何学的なサフィックス木を用いる方法である．幾何学的なサフィックス木では，2 つの座標配列の間の RMSD が閾値以下となる構造は同じ構造とみなして 1 本の枝で表されている．本章では，最初にその基本となるサフィックス木の説明を行った後，RSMD について触れ，幾何学的なサフィックス木の構築法について述べる．

2.1. サフィックス木

サフィックス木とは与えられた文字列のサフィックス(接尾辞)を木構造であらわしたデータ構造である．文字列 $S \in \Sigma^n$ のサフィックス木は，\$が $\$ \notin \Sigma$ のような文字である場合は，$S^+=S\$$のすべてのサフィックスの簡潔なトライ木である．葉はそれぞれ，文字列 S^+のサフィックスを表わし，ノードはそれぞれある部分文字列を表わす．このデータ構造は，シーケンス・パターンマッチング中の様々な問題に非常に役立つ．

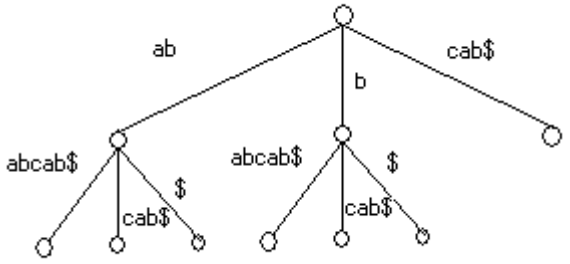

図 2.1. 文字列 *ababcab* でのサフィックス木の例

文字列 *ababcab* を用いてサフィックス木の例を示す．終端記号として文字列の最後に\$を追加する．文字 *a* で始まるものは 3 種類あり，その全てが辺 *ab* まで共通である．その下に辺は，*abcab*\$，*cab*\$，\$の 3 種類があ

り，全てで文字列の先頭に共通部分はない．文字 b で始まるものは 3 種類あり，その全てが辺 b のみ共通である．その下に辺は，$abcab\$$, $cab\$$, $\$$の 3 種類があり，全てで文字列の先頭に共通部分はない．文字 c で始まるものは 1 種類のみなので，1 本の枝のみで表わす．この結果を図 2.1 で示す．

2.2. RSMD

RMSD(平均二乗偏差)とは 2 つの点の集合間の幾何学的な類似度を決定する手段の一つである．p_i と q_j がともに 3 次元領域の座標であり，$i=j$ のとき p_i は q_j に相当する．このときの 2 つの点の集合 $P=\{p_1,p_2,\ldots,p_n\}$ と $Q=\{q_1,q_2,\ldots,q_n\}$を比較する．この比較について図 2.5 で示す．RMSD は適切な回転行列 R および平行移動ベクトル v における $\{(\sum_{i=1}^{n}\|p_i-(R\cdot q_i+v)\|^2/n)\}^{1/2}$ の最小値である．また，$R(P,Q)$および $v(P,Q)$を RMSD を最小化する R および v とする時，$\sum_{i=1}^{n}\|p_i-(R(P,Q)\cdot q_i+v(P,Q))\|^2$ を P と Q の MSSD(最小和二乗距離)と呼ぶ．

$v(P,Q)=\sum_{i=1}^{n}(p_i-R(P,Q)\cdot q_i)/n$，つまり，2 つの点集合の重心が同じ座標点に平行移動される場合，距離が最小化される．従って，それらの重心が座標の原点に位置するように，点集合が両方とも平行移動される場合，RMSD/MSSD 問題は，$f(R)=\sum_{i=1}^{n}\|p_i-R\cdot q_i\|^2$を最小化する Rを見つける問題になる．特異値分解(SVD)を以下のように使用することによって線形の時間で $\hat{R}(P,Q)$ を見つけることができる．$H=\sum_{i=1}^{n}p_iq_i^t$ とする．H の SVD が $U\Lambda V^T$ となり，$R=VU^T$ のとき，$f(R)$を最小化する R となる．RMSD の計算について図 2.2 で示す．

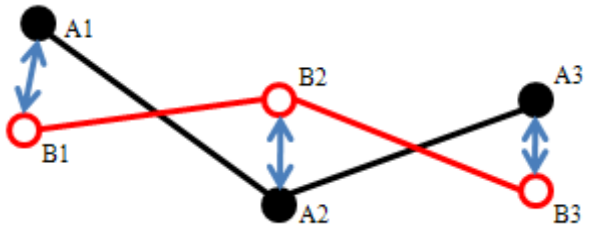

図 2.2. RMSD の例

図 2.2 において配列 A と配列 B の間の RMSD の求め方は，A1 と B1，A2 と B2，A3 と B3 の間の距離の和である．また，MSSD の値は RMSD の最小値である．

2.3. 幾何学的なサフィックス木の構築法

本来のサフィックス木とは違い，幾何学的なサフィックス木を構築するためには，本来のサフィックス木のデータ構造のほかに用いた座標配列の名前，座標配列のサフィックスの始まりの位置，及びサフィックス木の枝の長さをサフィックス木のデータ構造に含めておく必要がある．また，サフィックスの長さが 1 の場合そのサフィックスは辺をもたないので，サフィックスの長さが 2 以上のものでサフィックス木を構築する．

例えば，閾値を 1 とし，座標配列<(1,4)-(4,4)-(4,7)-(5,4)-(5,2)>で幾何学的なサフィックス木を構築してみよう．このときのサフィックスは以下のとおりである．
<(1,4)-(4,4)-(4,7)-(5,4)-(5,2)>, <(4,4)-(4,7)-(5,4)-(5,2)>, <(4,7)-(5,4)-(5,2)>, <(5,4)-(5,2)>, <(5,2)>
このとき，<(5,2)>は辺をもたないので除外する．最初に，サフィックス<(1,4)-(4,4)-(4,7)-(5,4)-(5,2)>と<(4,4)-(4,7)-(5,4)-(5,2)>を比較すると，サフィックス座標配列の間で両者の RMSD が閾値以下となるのは先頭から 3 番目までの部分配列である．従って，両者の部分配列を 1 本の枝で表現することができる．図 2.3 に，このときに構築されるサフィックス木を示す．図 2.4 は，サフィックス<(4,7)-(5,4)-(5,2)>, <(5,4)-(5,2)>を図 2.3 に追加することによって構築された幾何学的サフィックス木である．

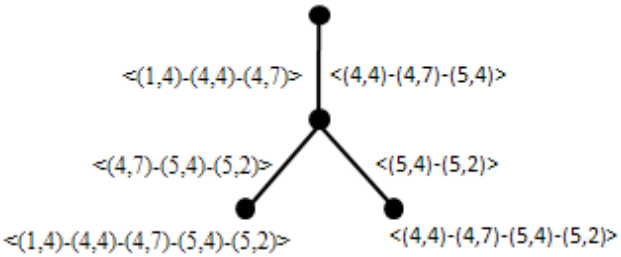

図 2.3. 初期段階の幾何学的なサフィックス木

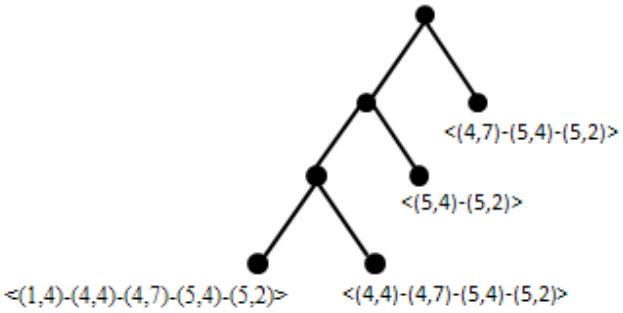

図 2.4. 最終段階の幾何学的なサフィックス木

3. 提案手法

本章では，複数の蛋白質座標配列での幾何学的なサフィックス木の構築する方法，構築したサフィックス木を保存して再利用する方法について提案し，サフィックス木での類似部分構造の検索について触れる．

3.1. 複数本の配列データでのサフィックス木の構築

複数本の座標配列データでのサフィックス木の構

築の方法は，サフィックス木のデータ構造に座標配列を識別する蛋白質の名称を含めておく必要がある．そして，蛋白質の座標配列のサフィックスすべての配列で幾何学的なサフィックス木を構築する．この方法により，複数本の配列データでサフィックス木の構築ができるようにしている．

幾何学的なサフィックス木では，本来のサフィックス木と異なり構築の際に誤差が現れる．この誤差の修正のために，サフィックス木の枝の変更，もしくは検索の際に周辺の枝も含めた検索について考慮しなければならない．本研究では，サフィックス木の検索で誤差を考慮している．

3.2. サフィックス木のディスクへの保存

サフィックス木をディスクへ保存する方法として DynaCluster アルゴリズムを利用する．DynaCluster アルゴリズムは動的クラスタリング法を用いて，サフィックス木の保存を行っている．この方法を応用して，サフィックス木をディスクに保存してメモリ上へのサフィックス木の使用メモリを減らすことにより，大規模なサフィックス木を構築することができるようにしている．

3.3. サフィックス木の検索

サフィックス木の検索を行う前に，あらかじめ検索キーとなる蛋白質をサフィックス木に含めておく．そして，この検索キーを含んだサフィックス木の中から検索キーにかかわる部分に対して検索を行う．

サフィックス木の検索を行うにあたって，検索キーとなる蛋白質のどのくらいの長さが類似するものを見つけるかを決める．そして，その長さ以上のサフィックスを検索キーとして検索する．このことを図 3.1 に示す．

検索では，この検索キーをサフィックス木で辿って行き，検索する蛋白質のサフィックスと類似する部分構造を見つける．

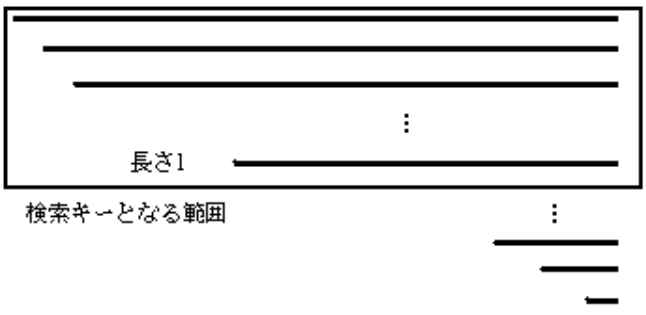

図 3.1. 検索する蛋白質の検索キー

4. 実験

本章では幾何学的なサフィックス木の構築を，メモリのみの場合とメモリとディスクを用いた場合の性能の比較を行う．これにより，ディスクを用いた場合の提案手法の有効性を確認する．

4.1. 実験方法

実験 1

蛋白質の座標配列から幾何学的なサフィックス木の構築をメモリ上のみを用いる場合とメモリとディスクを用いる場合の 2 つの方法で行い，サフィックス木を構築できる配列数の比較，構築時間の計測を行う．

実験 2

幾何学的なサフィックス木の構築をメモリ上のみ用いた場合とメモリとディスクを用いる場合で，検索の精度および検索にかかる時間の比較を行う．

実験環境

実験環境は次の通りである．

OS:Fedore Core 9

CPU:Intel(R) Core(TM)2 Quad CPU×2

メモリ:2GB

HDD:454GB

実験データ

実験データは日本蛋白質構造データバンク(PDBj)に登録されているデータを用いる．また，ここでの蛋白質の名称は PDB ID で表記する．

4.2. 実験結果

実験 1 の結果を表 4.1 に示す．

表 4.1. 実験 1 の結果

	メモリのみ		メモリとディスク	
配列数(本)	58	351	58	351
構築時間(s)	2.165168		6.237834	51.213875
使用ディスク量(GB)				15

この結果から，サフィックス木をディスク上に保存することにより，サフィックス木を構築できる配列数が増加している．この結果，サフィックス木の構築時間が増加している．

また，実験 2 の結果を表 4.2 に示す．

表 4.2. 実験 2 の結果

	メモリのみ	メモリとディスク
配列数(本)	58	351
検索精度(%)	94	92
検索時間(s)	0.884631	9.419635

この結果から，サフィックス木の検索にはディスクを用いると結果が少し悪くなっている．

4.3. 考察

実験 1

実験 1 の結果から，ディスクを用いることによりサフィックス木を構築できる蛋白質数が増加しているので，多くの蛋白質を一度に使用して実験することができる．構築にかかる時間が，ディスクへの書き込みにかかる時間が現れるので，メモリのみの場合よりも多

くなっている．このことから，サフィックス木の構築のために頻繁にディスクの読み書きを行っており，この時間の割合が多くなっていることがわかる．

実験 2

実験 2 の結果から，検索の精度がディスクを用いると下がっている．このことは，実験 1 の結果のように用いた蛋白質の配列数が多くなったことから，他の蛋白質に隠れるものがあったからだと思われる．また，検索時間については検索される蛋白質数の増加，ディスクからの読み込みにかかる時間によってメモリのみの検索時間よりも時間がかかっていることがわかる．

5. まとめ

本研究では，蛋白質立体構造から幾何学的なサフィックス木を構築して類似する蛋白質の検索，および構築したサフィックス木をディスク上に保存することによる構築と検索の精度の確保が確認できた．

今後の課題として，本研究での実験の効率化，検索精度の向上，および類似構造の視覚化などがあげられる．また，蛋白質の座標配列取得方法としてα-カーボンの計算による方法[5][6][7]によって得る方法がある．この方法で得られた蛋白質の座標配列を本研究に用いた場合の違いについての構築結果と検索結果についての研究が考えられる．

謝　辞

本研究の一部は，日本学術振興会，科学研究費補助金(基盤研究(C)，課題番号：20500137)の支援により行われた．

文　献

[1] F. Gao and M.J.Zaki. PSIST: Indexing Protein Structures using Suffix Trees. Proc. IEEE Computational Systems Bioinformatics Conference (CSB), pp.212-222, 2005.

[2] Tetsuo Shibuya. Geometric Suffix Tree: A New Index Structure for Protein 3-D Structures. Combinatorial Pattern Matching 2006, LNCS 4009, pp.84-93, 2006.

[3] Ching-Fung Cheung, Jeffrey Xu Yu, and Hongjun Lu. Constructing suffix tree for gigabyte sequences with megabyte memory. IEEE Trans. Knowl. Data Eng., Vol. 17, No. 1, pp. 90–105, 2005.

[4] 荒木康太郎，田村慶一，加藤智之，黒木進，北上始．曖昧検索に基づく最小汎化パターンの抽出法．DEWS2007.

[5] Mariusz Milik, Andrzej Kolinski, and Jeffrey Skolnick. Algorithm for Rapid Reconstruction of Protein Backbone from Alpha Carbon Coordinates. Journal of Computational Chemistry, Vol 18, No.1, p.80-85, 1997.

[6] A. LIW0, M .R. PINCUS, R.J. WAWAK, S. RACKOVSKY AND H.A. SCHERAGA. Calculation of protein backbone geometry from a-carbon coordinates based on peptide-group dipole alignment. Protein Science, Vol. 2, Issue 10, 1715-1731, October 1993.

[7] Julien Maupetit, R. Gautier and Pierre Tuffe'ry. SABBAC: online Structural Alphabet-based protein BackBone reconstruction from Alpha-Carbon trace. Nucleic Acids Research, 2006, Vol. 34, Web Server issue W147–W151.

[8] A. LIW0, M .R. PINCUS, R.J. WAWAK, S. RACKOVSKY AND H.A. SCHERAGA. Calculation of protein backbone geometry from a-carbon coordinates based on peptide-group dipole alignment. Protein Science, Vol. 2, Issue 10, 1715-1731, October 1993.

[9] Julien Maupetit, R. Gautier and Pierre Tuffe'ry. SABBAC: online Structural Alphabet-based protein BackBone reconstruction from Alpha-Carbon trace. Nucleic Acids Research, 2006, Vol. 34, Web Server issue W147–W151.