知能と複雑系　132－6
(2003. 3. 13)

配列からの頻出パターン抽出のためのWebシステム

徳山 文範　北上 始　森 康真

広島市立大学情報科学部

〒731-3194 広島市安佐南区大塚東 3-4-1

E-mail: {fuminori, kitakami, mori}@db.its.hiroshima-cu.ac.jp

あらまし　本稿では，すでに著者らによって開発されているModified PrefixSpan法により大規模なアミノ酸配列データから極めて多量に抽出される頻出パターンに対して，それらから知識発見を行うときに生じる利用者の手間を軽減するためのWebシステムを提案する．本システムでは，抽出された頻出パターンを構造化し，それを視覚的に表示する機能，頻出パターンのワイルドカード領域を特殊化する機能，頻出パターンの配列上の所在位置を視覚的に表示する機能などを持たせた．実装した機能により利用者は視覚的に多量のデータの情報を得る事ができる．今後，この試作システムの機能をさらに強化し，インタフェース等の評価を行う予定である．

キーワード　配列データ，データマイニング，頻出パターン，視覚化

Web System for Extracting Frequent Patterns from Sequences

Fuminori TOKUYAMA　Hajime KITAKAMI　and　Yasuma MORI

Faculty of Information Sciences, Hiroshima City University

3-4-1 Ozuka-Higashi, Asa-Minami-Ku, Hiroshima, 731-3194 Japan

E-mail: {fuminori, kitakami, mori}@db.its.hiroshima-cu.ac.jp

Abstract　The paper proposes a Web system to reduce the users' workload caused when knowledge is discovered from frequent patterns extracted from a large-scale amino acid sequence database using Modified PrefixSpan method that has already been developed by authors. This system structurizes the set of extracted frequent patterns and visualizes it, and provides such functions as displaying the whereabouts position in the sequences and specializing the wild-card area. The user can visually obtain information on a large amount of data using the prototype system. In the future, we will enhance the prototype system and evaluate the system interface.

Keyword　Seqences Database, Data Mining, Frequent pattern, Visualization

1. はじめに

モチーフは，アミノ酸配列上における特徴的なパターンであり，生物の進化の過程で保存されてきた蛋白質の機能や構造に深く関係していると考えられている．これにより，分子生物学分野の様々な専門家が，自分達の研究に関連するアミノ酸配列を多数集め様々なモチーフが見出されてきた．それらは，PROSITEやPfamといったモチーフライブラリーとして整備されている[1]．我々は，モチーフ発見の支援を目指しアミノ酸配列からモチーフの候補となる頻出パターンを従来の抽出方法[2]よりも高速に抽出するためにModified PrefixSpan法[3],[4]を提案してきた．しかし，配列データベースから抽出される頻出パターンの数は極めて多量であるため，利用者がそれらを知識発見のために直接閲覧するには大変な手間がかかるという問題がある．

本稿では，この問題を解決するために，Modified PrefixSpan法による頻出パターン抽出の機能をもつWebシステムの構成法を提案する．本システムでは，抽出された頻出パターンを構造化しそれを視覚的に表示する機能，頻出パターンのワイルドカード領域を特殊化する機能，頻出パターンの配列上の所在位置を視覚的に表示する機能などをもち，それらはJavaアプレット，PHP, PostgreSQLなどで実装されている．

2. 頻出パターンの抽出法についての関連研究

マルチプルアライメントを用いて，あるアルファベット集合Σの上で定義されているN本の配列集合$\{S_1,S_2,...S_N\}$の平均長をLとすると，その配列集合からモチーフを見つけ出す方法は，時間計算量が$o(L^N)$となるので膨大な計算時間を要する．それにもかかわらず，極めて限定された頻出パターンしか得られない．

これに対して，MEME [5]では，アライメントされていない N 本の配列集合からおおまかな情報としてモチーフの長さ W と各配列中の位置を利用者に指定させ，その部分に対してマルチプルアライメントを行った後，統計的手法である期待値最大化アルゴリズムをM回繰り返すことにより，複数の頻出パターンを抽出している．これにより，MEME では，時間計算量を $o((NM)^2W)$に抑えているが，抽出される頻出パターン中にワイルドカードを含まないため，PROSITE や Pfam などで見られるモチーフとは直接結びつきにくい．

これに対して，Jonassen らが開発した Pratt というシステム[6]では，利用者に抽出パターン P の最大長 W を指定させ，各配列から長さ Wの部分配列（以後，セグメントと呼ぶ）を全てとりだし，それらの集まりとして定義されるセグメント集合 B_Wからワイルドカードを含む頻出パターン P を全て抽出することができる．ただし，配列の終端付近からも頻出パターンを抽出するために，長さ Wの部分配列を取り出す前に各配列の後ろに $W-1$ 個のダミー記号（Σには存在しない）が予め追加されている．全ての頻出パターンは k 文字長のパターン集合から $k+1$文字長のパターン集合を再帰的に構成する方法により抽出される．この処理は，最大長が W文字になるまで繰り返し計算が行われる．計算量の見積もりは難しいが，効率的な抽出のために，セグメント集合 B_Wは各構成要素 a が i 番目にもつセグメントの集合 $b_{i,a}(\subseteq B_W)$の集まりとして構造化されている．例えば，パターン P から $P'=P\text{-}x(j)\text{-}a$ を構成する場合，P' は P が前方一致するセグメント集合 M_p と要素 a が $L(P)+j+1$ 番目に位置するセグメント集合 $b_{L(P)+j+1,a}$ との共通セグメントの全てと前方一致する（ただし，j は 0 以上の整数だが，$L(P)+j+1\leqq W$ を満たす）．これにより，P'が存在する配列数を数え，その数が利用者によって与えられた最小支持数 N_{min} 以上であれば，P'を頻出パターンとしている．ただし，$L(P)$はワイルドカード領域を含むパターン P の長さを表す．

しかし，著者らが開発した Modified PrefixSpan 法[3],[4]は，抽出される頻出パターン P の最大長 Wの利用者による指定が不要である．その代りに，利用者にワイルドカード領域の最大長 Vを指定させている．これにより，著者らの方法では，利用者に予期していなかった長さの頻出パターンを発見する可能性を与えている．また，この方法も，Pratt のように，長さが k の頻出パターンから長さが $k+1$の頻出パターンを再帰的に作成しながら全ての頻出パターンを抽出しているが，その処理過程においては，頻出パターン長の制限につながるセグメント集合を作らずに，直接 N本の配列集合から頻出パターンを抽出している．効率的な処理を行うために，N 本から成る配列集合$\{S_1,S_2,\ldots S_N\}$の構造化は，配列の構成要素 a と配列 S_iとの組みごとに，配列 S_iの先頭から何番目に a が存在するかを示す位置情報の集合 $T_{a,i}$を作成することにより達成している．著者らはこれをアドレステーブル[4]と呼んでいる．例えば，パターン P から $P'=P\text{-}x(j)\text{-}a$ かつ $0\leqq j\leqq V$ を構成する場合を考えてみよう．各配列 S_i において $last(P,i)$を P の最後尾にある要素の位置情報とすると，位置情報の集合 $T_{a,j}$中に $last(P,i)+1$から $last(P,i)+V+1$ までの区間に当該要素 a の位置情報があるかどうか調べ，その位置情報が存在する配列 S_iの数が最小支持数 N_{min}以上であるものを P'としている．

3. パターンの定義

N本の配列集合$\{S_1,S_2,\ldots S_N\}$の夫々は，あるアルファベット集合Σ上で定義されているとする．この集合から抽出するパターン P の形式は以下のとおりである．

$P=A_1\text{-}x(i_1,j_1)\text{-}A_2\text{-}x(i_2,j_2)\text{-}\ldots\text{-}x(i_{p-1},j_{p-1})\text{-}A_p$

A_iを文字要素または単に要素と呼ぶ（記号 pは要素数である）．また，文字要素 A_iが，1文字で表現されるとき単一要素，2文字以上で表現（たとえば[ILVF]など）されるとき曖昧要素と呼ぶ．$X(i_k,j_k)$の領域において，$i_k<j_k$のとき，その領域を可変長ワイルドカード領域と呼ぶ．$i_k=j_k$のとき，それを固定長ワイルドカード領域と呼び，この領域を $x(i_k)$で簡略表現することがある．パターン Pにマッチする部分配列の最大長を $L(P)$で表現すると，$L(P)$は，$L(P)=p+\Sigma j_k[1\leqq k\leqq p-1]$で表される．

このパターン Pが利用者によって与えられた最小支持数 N_{min}以上の配列にマッチするとき，そのパターンは頻出パターンと呼ばれる．本稿では，ワイルドカード領域の最大長 Vを満足する頻出パターンだけを抽出する方法を採用しており，この方法でも N本の配列集合からは膨大な数の頻出パターンが抽出される．以下では，抽出された頻出パターンの集合を閲覧する際に便利な構造化について述べる．

4. パターン集合の構造化

膨大な量のパターン集合 R を閲覧する際，パターン集合をある基準で分類階層化し，そこから利用者に興味ある分類ノードを指定させ，そのノード以下に含まれる類似したパターンを一度に閲覧できるようにすれば，パターン集合の絞り込み検索に利用できると考えている．ここでは，頻出パターン集合の構造化として，トライ構造と中間マッチ構造に着目し，各々の効果について述べる．

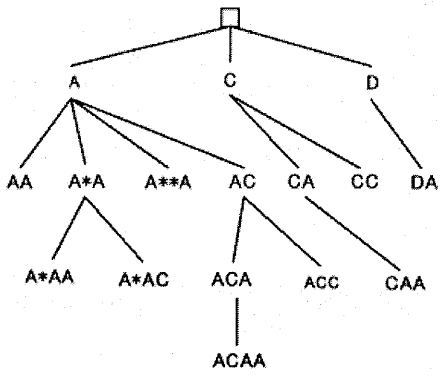

図 1.　トライ構造の例

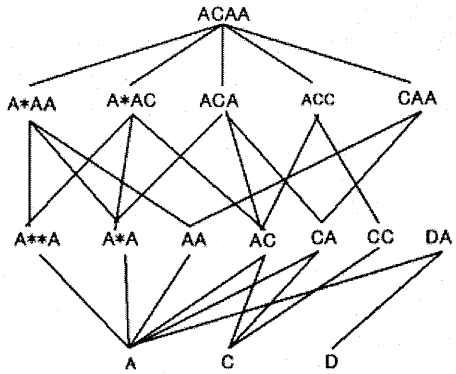

図 2.　中間マッチ構造の例

（1）　トライ構造

パターン中に含まれるワイルドカードをアルファベット文字の 1 つと見做し，頻出パターン集合 R をトライ構造で表す方法である．トライ構造は，パターンの前方文字列が一致するパターンを寄せ集め，一致する前方文字列の長さに基づいて作成される木構造である．一致する長さが短いパターンは木構造の根の近くに配置され，その一致する長さが長くなるにつれ深さ方向にパターンが配置される．従って，このトライ構造を辿ることにより，前方文字列が同じパターンの集まりを容易に閲覧することが可能になる．

（2）中間マッチ構造

頻出パターンの集合 R をワイルドカードが含まれる文字列 $P=A_1,A_2,...A_n$ の集合と見なす，R に含まれる文字列間の中間マッチングによって定められる順序関係で頻出パターン集合を順序付けし，それによって定義される半順序集合 $(R,\leqq)$ をグラフ表現した構造を中間マッチ構造と呼ぶ．ここでは，パターン文字列 $P_A=A_1,A_2,...A_n$ がテキスト文字列 $P_B=B_1,B_2,...,B_m$ に中間マッチするとき，即ち，$A_1=B_j,A_2=B_{j+1},...,A_n=B_{j+n-1}$ が成立するとき，$P_A \leqq P_B$ と定義する $(n \leqq m, 1 \leqq j \leqq m-n+1)$．ただし，この中間マッチでは，$A_i=B_{j+n-1}$ が成立するのは以下の条件のどちらかに限る $(2 \leqq i \leqq n-1))$．

- パターン文字列側の A_i がワイルドカードの場合，テキスト文字列側の B_{j+i-1} はワイルドカードまたは文字要素でなければならない．
- パターン文字列側の A_i が文字要素の場合，テキスト文字列側の B_{j+i-1} は同一文字要素でなければならない．

頻出パターン集合 R を{ *A, C, D, A**A, A*A, AA, AC, CA, CC, DA, A*AA, A*AC, ACA, ACC, CAA, ACAA* }とし，トライ構造および，中間マッチ構造について考えてみよう．

図 1 はトライ構造の例である．根から下の方向に進むにしたがい，支持数が低くなるという性質がある．図中のパターン *A*A* を前方文字列としてもつパターン *A*AA* と *A*AC* はトライ構造の下をたどることにより，ただちに見つけることができる．図 2 は中間マッチ構造の例である．この場合は下の方向に進むにしたがい支持数が高くなるという性質がある．図中のパターン *A*AC* に中間マッチするパターンの集合{*A**A, A*A, AC, A, C*}は中間マッチ構造の下をたどることにより，ただちに見つけることができる。

5. 実装

図 3 に本研究で試作したシステム構成を示す．本システムは以下の手順で動作する．

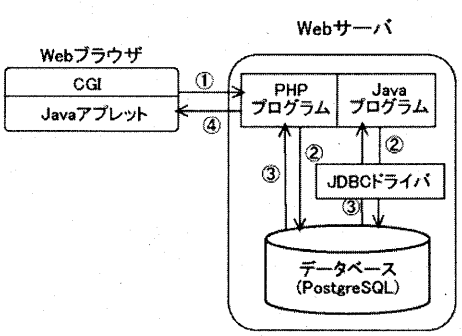

図 3.　システム構成

①Web ブラウザから CGI，Java アプレットを通じて処理要求を送る．

②処理要求を受け取ったサーバ側のプログラムが処理を実行する．必要があればデータベースに接続して処理を実行する．Java アプレットはデータベースに接続する際に JDBC ドライバという特殊なプログラムを使用する．

③データベースから必要なデータを取ってきてそれを処理要求にしたがって処理する．

④処理結果を CGI，Java アプレットを通じて Web ブラウザに出力する．

5.1. データ処理の流れ

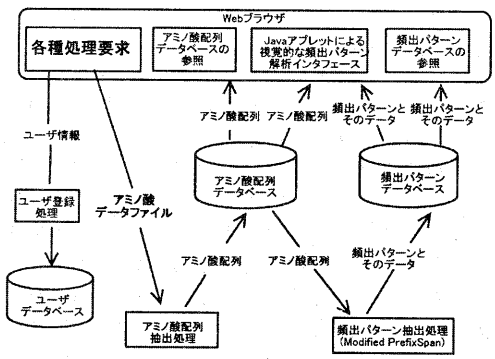

図 4．データ処理の流れ

図 4 は，本システムにより処理されるデータの流れを表している．アミノ酸配列から頻出パターンを抽出するまでには以下のような処理ステップが適用される．

- アミノ酸配列データが格納されているデータファイルを準備する．
- データファイルからアミノ酸配列を選択する．
- 選択したアミノ酸配列の集合に対する利用者管理を行う．
- アミノ酸配列からの Modified PrefixSpan 法による頻出パターンを抽出する．
- 抽出した頻出パターンの集合に対する利用者管理を行う．

本システムでは，以上の処理を統合し利用者に扱いやすい Web インタフェースを持つシステムを実装している．また，本研究の目的である抽出された大規模な頻出パターンデータベースから知識発見の支援を行うために視覚的な情報提供を行うためのインタフェースを作成した．以下では，そのインタフェースが有する頻出パターンの構造化，頻出パターンのワイルドカード領域の特殊化，頻出パターンの配列上の所在位置を視覚的に表示する機能について述べる．

5.2. インタフェース画面

図 5 は前述したトライ構造により抽出された頻出パターンを構造化したものである．各頻出パターンの横に付加している数字は支持率である．利用者は目的とする頻出パターンや支持率まで表示することにより，その周辺の類似するパターンなども同時に把握することができる．これにより頻出パターン集合をかたまりとして捉えることができる．

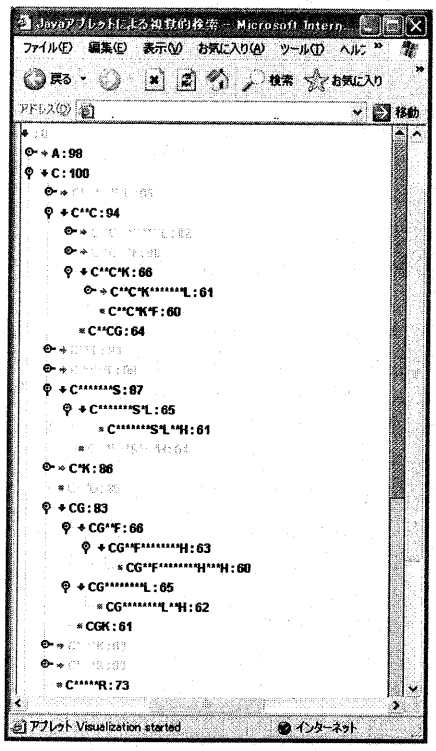

図 5．　頻出パターンのトライ構造による構造化

図 6 はワイルドカード領域の特殊化に利用する Web ページの画面である．Modified PrefixSpan 法により抽出される頻出パターンはほとんどの場合，ワイルドカード領域を含む．ワイルドカード領域の特定の部分にはある同じ要素しか存在しない，またはある数種類の要素が存在したときそれを同じ頻出パターンとみなすということがあるため，このワイルドカード領域の特殊化にはそのような頻出パターンの発見に有効であると考えられる．たとえば図 6 中の 5 番目のワイルドカードは，曖昧要素*[FWY]*であることがわかる．

図 7 は指定した頻出パターンが配列上のどの位置に存在するかという情報を視覚的に表示するものである．配列は 10 文字ごとに区切りが入っており，また

60 文字ごとに改行されている．このように表示することにより配列上の頻出パターンが何文字目から何番目まで出現するか，また頻出パターン同士の位置関係などの情報を視覚的に得る事ができる．

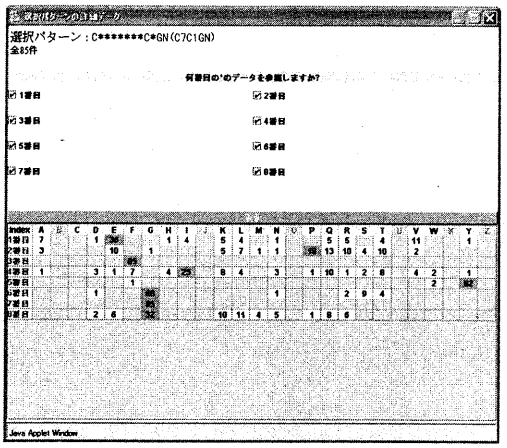

図 6.　ワイルドカード領域の特殊化

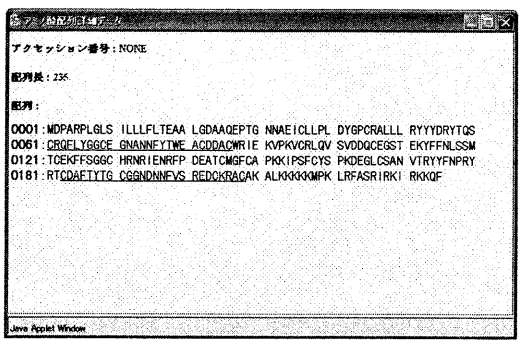

図 7.　頻出パターンの配列上の所在位置表示

6. まとめ

本研究では，アミノ酸配列から Modified PrefixSpan 法により抽出された極めて大規模な頻出パターンに対して，利用者がそこから知識発見の支援をするための Web システムを提案した．そして，視覚的に頻出パターンやその集合を捉えることのできる機能を一部実装した．今後の課題は以下のとおりである．

- 曖昧要素の抽出機能の強化
  Kringle というアミノ酸配列には F3GC6[FY]5C というモチーフが存在する．このモチーフには曖昧要素が含まれている．試作したシステムでは，F3GC6Y5C は抽出されているが F3GC6F5C は最小支持数に達していないため抽出されていない．このような同等のものとして扱われる要素を含む頻出パターンの抽出機能の実装が重要である．
- 頻出パターンの構造化の改良
  今回試作したシステムでは，頻出パターンをトライ構造により構造化しただけである．今後は，中間マッチ構造の実装が残されている．これにより，頻出パターンと部分一致するパターンが周辺に集まるため，一目で類似パターンを検索，把握することも可能になると考えられる．
- 専門家の観点からの評価
  このシステムをさらに強化し，インタフェース上どの程度有用かの評価をすることが重要である．

参 考 文 献

[1] 金久 實:ポストゲノム情報への招待，共立出版社，2001 年．

[2] J. Pei, J. Han, B. Mortazavi-Asl, and H. Pinto: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, Proceedings of the 7th International Conference on Data Engineering (ICDE2001), IEEE Computer Society Press, pp.215-224, 2001.

[3] Tomoki Kanbara, Yasuma Mori, Hajime Kitakami, Susumu Kuroki and Yukiko Yamazaki: Discovering Motifs in Amino Acid Sequences using a Modified PrefixSpan Method, Currents in Computational Molecular Biology 2002, ACM-SIGACT, Washington DC, pp.96-97, April 2002.

[4] Hajime Kitakami, Tomoki Kanbara, Yasuma Mori, Susumu Kuroki, and Yukiko Yamazaki: Modified PrefixSpan Method for Motif Discovery in Sequence Databases, Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI2002), pp.482-491, Springer-Verlag, August 2002.

[5] Timothy L. Bailey and Charles Elkan, Fitting a mixture model by expectation maximization to discover motifs in biopolymers, Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, pp. 28-36, AAAI Press, Menlo Park, California, 1994.

[6] Inge Jonassen, John F. Collins, and Desmond G. Higgins: Finding flexible patterns in unaligned protein sequences, Protein Science, pp.1587-1595, Cambridge University Press, 1995.