ミスマッチクラスタに対する最小汎化パターン抽出方式

Extraction of Least Minimum Generalization from Mismatch Cluster

荒木 康太郎♥ 田村 慶一♦
加藤 智之♥ 北上 始♦

Kotaro ARAKI Keiichi TAMURA
Tomoyuki KATO Hajime KITAKAMI

配列データベースに対する曖昧検索では，非常に多くの類似した部分文字列の集合が検索結果として得られる．我々は，この集合をミスマッチクラスタと呼ぶ．ミスマッチクラスタの全体像の把握を容易にするためには，ミスマッチクラスタから最小汎化された曖昧配列パターン集合を抽出しなければならない．著者らは，曖昧パターン集合を抽出するために，反復精密化法を提案し，その効率的抽出のためにドメイン分割法を提案する．我々が提案する方法の有効性を評価するために，提案方法を実装し，アミノ酸配列からモチーフを抽出する問題に適用した．実験の結果，我々の予測する結果が効率的に得られた．

An ambiguous query in sequence database returns a set of similar subsequences, called a mismatch cluster, to the user. In order to support user comprehension of mismatch clusters, it is important to extract a set of ambiguous sequence patterns with the least minimum generalization in the mismatch cluster. The present paper proposes an iterative refinement to extract ambiguous sequence patterns from mismatch clusters and domain segmentation methods to achieve an effective pattern extraction. Moreover, a prototype implementing the two proposed methods has been applied to three datasets included in PROSITE in order to evaluate their usefulness. The proposed methods resulted in a high capability.

1. はじめに

曖昧検索は，配列データベースから類似する部分文字列の検索をさし， DNAやアミノ酸モチーフの抽出，肺の損傷が類似した進行特性をもつ患者の発見などへの応用として期待されている．モチーフとは，PROSITE[6] やPfamなどで見られる生物学的に重要な機能をもつ特徴的なパターンである．今までに，数多くの曖昧検索の研究[1] [2] [3] [4] [5] が行われてきたが，従来の研究では，長さkの基準文字列と誤差半径r以内にあるk-部分文字列を全て求めるだけにとどまっている．一般に，曖昧検索では，非常に多くの類似した部分文字列の集合が検索結果として得られる．我々は，この集合をミスマッチクラスタと呼ぶ．以下では，ユーザがこのミスマッチクラスタを閲覧し，その全体像を把握することは極めて困難であるという問題に着目する．

本論文では，この問題を解決するために，ミスマッチクラスタから最小汎化された曖昧配列パターンを抽出する方法について提案する．曖昧配列パターン集合を抽出するには，$O(2^{lk})$の汎化パターン集合からミスマッチクラスタを最小被覆する汎化パターンを見つけ出さなければならないため，膨大な計算時間を要する．ただし，kは曖昧配列パターン上の曖昧文字部位の数, lは各曖昧文字の候補となり得る文字数とする．著者らは，曖昧配列パターン集合を効率的に抽出するために，以下の2点を提案する．

(1) 反復精密化法により, 負の汎化パターン集合の算出を経て，最小汎化された正の曖昧パターン集合を抽出する．
(2) 上記の(1)に以下のドメイン分割法を併用する．ドメイン分割法により，文字間の距離行列がユーザの背景知識に基づいて，最汎パターンの各曖昧文字をクラスタ分割し，計算の途中で無意味な曖昧パターン生成を回避する．

2. 用語と記号の定義

Σ_iをアルファベットΣの部分集合とすると，k-配列パターン(k 個の Σ_i を並べたパターン)は以下の形式で表現する．

$$<pat^k>=<\Sigma_1\text{-}x(i_1,j_1)\text{-}\Sigma_2\text{-}x(i_2,j_2)\text{-}\ldots\text{-}\Sigma_{k-1}\text{-}x(i_{k-1},j_{k-1})\text{-}\Sigma_k> \quad (1)$$

式(1)中の， $\Sigma_i\subseteq\Sigma$は，$|\Sigma_i|\geqq 2$のとき，曖昧文字と呼ばれ，Σ_i内に存在する任意の1文字の配置が許されることを示している．また，集合$\sum_i$を曖昧文字ドメインとも呼ぶ．曖昧文字ドメインが1個以上存在するとき，式(1)をk-曖昧配列パターンと呼ぶ．ハイフン“-”は左右の文字の接続を意味するが，以後たびたび省略されることがある．$x(i,j)$は，ワイルドカード領域と呼ばれ，ワイルドカード数がi個からj個までの範囲内であることを示している．$i<j$のとき，$x(i,j)$を可変長ワイルドカード領域と呼び，$i=j$のとき，$x(i,j)$は$x(i)$と書き，$x(i)$は，固定長ワイルドカード領域と呼ぶ．$x(0)$は空文字を意味する．

以下では，曖昧表現に関する本質を浮き彫りにするために，ワイルドカード領域の部分の記述を省略し，k-曖昧配列パターンを以下のように表現する．

$$<pat^k>=<\Sigma_1\text{-}\Sigma_2\text{-}\ldots\text{-}\Sigma_{k-1}\text{-}\Sigma_k> \quad (2)$$

特に， $\Sigma_i=\{\alpha_i\}$のとき，式(2)を以下のように表現する．

$$<pat^k>=<\alpha_1\text{-}\alpha_2\text{-}\ldots\text{-}\alpha_{k-1}\text{-}\alpha_k>$$

2.1 曖昧配列パターンとインスタンスの関係

k-曖昧配列パターン$<\Sigma_1\Sigma_2\ldots\Sigma_{k-1}\Sigma_k>$からインスタンスのすべて(部分文字列の集合)を導出する関数を$EVAL(<\Sigma_1\Sigma_2\ldots\Sigma_{k-1}\Sigma_k>)$と書くことにすると，以下の関係式が成立する．

$$EVAL(<\Sigma_1\Sigma_2\ldots\Sigma_{i-1}(\Sigma_i-\{\alpha_i\})\ldots\Sigma_k>) = EVAL(<\Sigma_1\Sigma_2\ldots\Sigma_{i-1}\Sigma_i\ldots\Sigma_k>)- EVAL(<\Sigma_1\Sigma_2\ldots\Sigma_{i-1}\{\alpha_i\}\ldots\Sigma_k>)$$

以後，関数 $EVAL$ によって，複数の k-インスタンスから成る集合を導出することができる k-曖昧配列パターンを k-汎化パターン(あるいは単に汎化パターン)とよぶ．

2.2 汎化パターン集合

ある k-インスタンス(長さ k の部分文字列)の集合を I^k としよう．$1\leqq j\leqq k$ を満たすアルファベット Σ_j を

$$\{\, inst[\,j\,] \mid inst \in I^k \,\}$$

で定義するとき，$<\Sigma_1\Sigma_2\ldots\Sigma_{i-1}\Sigma_i\ldots\Sigma_k>$を k-インスタンス集合 I^k に対する最汎パターン$<mgpat^k>$と呼ぶ．また I^k を最小被覆する汎化パターン集合を最小汎化パターン集合と呼ぶ．

k-インスタンスを要素とするミスマッチクラスタ MIS =

♥ 学生会員 広島市立大学大学院情報科学研究科
{kotaro,kato}@db.its.hiroshima-cu.ac.jp
♦ 正会員 広島市立大学大学院情報科学研究科
{ktamura,kitakami}@db.its.hiroshima-cu.ac.jp

{<mis_1^k>,<mis_2^k>,...,<mis_m^k>}があるとし，*MIS* に対する最汎パターン<$mgpat^k$>が作成されたとしよう．<$mgpat^k$>から *MIS* を含まない最小汎化パターン集合が計算されたとき，その集合の中に含まれる各要素を負の汎化パターンと呼ぶ．計算方法については 3 章で述べる．また，最汎パターン<$mgpat^k$>から負の汎化パターン集合を含まない最小汎化パターン集合が計算されたとき，それを正の汎化パターン集合と呼ぶ．正の汎化パターン集合は，*MIS* を最小被覆する汎化パターン集合になる．

3. 最小汎化パターンの導出

k-曖昧配列パターンにおいて，各曖昧文字ドメイン Σ_i が $|\Sigma_i|=l$ を満たすとき，長さ *k* の最汎パターン<$mgpat^k$> = < $\Sigma_1 \Sigma_2 ... \Sigma_{i-1} \Sigma_i ... \Sigma_k$>に含まれる汎化パターンやインスタンスは，合計で$(2^k-1)l^k$個存在する．このうち，インスタンスは l^k 個存在するが，汎化パターンの個数は，依然として 2^{lk} のオーダーであるため，この中から最小汎化パターンを探すには膨大な計算時間を要することがわかる．我々は，ミスマッチクラスタを最小被覆する汎化パターンを効率的に抽出するために，以下の処理を行っている．

(1) 基準となる *k*-部分文字列と誤差半径を満たすミスマッチクラスタを配列データベースから見つけ出す．
(2) ミスマッチクラスタから最汎パターンを生成する．
(3) 最汎パターンの曖昧文字のそれぞれに対して，「ドメイン分割法」を適用し，新たなる最汎パターン集合を生成する．
(4) 「反復精密化法」を以下のように適用する．
- 上記(3)の新たなる最汎パターン集合に対して，インスタンス集合を含まない負の最小汎化パターン集合を計算する．
- 上記(3)の新たなる最汎パターン集合に対して，上記で計算済みの負の最小汎化パターン集合を含まない正の最小汎化パターンを計算する．これが，最小汎化パターン集合であり，計算結果として出力する．

以下では，まず、上記の反復精密化法の基本となる最小被覆の計算法を述べた後、反復精密化法とドメイン分割法の 2 つの手法について述べる．

3.1 最小被覆の計算

k-汎化パターン<pat^k>と *k*-インスタンス集合 I^k={<$inst_1^k$>, <$inst_2^k$>,...,<$inst_n^k$>}があるとしよう．*k*-汎化パターン<pat^k> = < $\Sigma_1 \Sigma_2 ... \Sigma_k$>から導出されるインスタンス集合を *EVAL*(<pat^k>)と表現する．これから *k*-インスタンス<$inst^k$>=<α_1-α_2-...-α_{k-1}-α_k>$\in I^k$ を除去したインスタンス集合を最小被覆する *k*-汎化パターンの集合を *COVS*(<pat^k>,<$inst^k$>)で表現し，これを<pat^k>から<$inst^k$>を除去された結果に対する最小汎化パターン集合と呼ぶ．このとき，以下が成立する．

$$COVS(<pat^k>,<inst^k>) = \{ < \Sigma_1 \Sigma_2 ... \Sigma_{i-1}(\Sigma_i - \{\alpha_i\}) ... \Sigma_k > \mid 1 \leqq i \leqq k \} \quad (3)$$

$\alpha_i \in \Sigma_i$ であれば，*COVS*(<pat^k>, <$inst^k$>)は，*k* 個の汎化パターンを要素として持ち，$\alpha_i \notin \Sigma_i$ であれば，*COVS*(<pat^k>, <$inst^k$>)={<pat^k>}である$(1 \leqq i \leqq k)$.

式(3)において，*k*-インスタンス<α_1- α_2 - ... - α_{k-1} - α_k>を k-汎化パターン<$\Gamma_1\Gamma_2 ... \Gamma_k$>に置き換えると，以下のように拡張することができる．ただし，$\Gamma_i \subseteq \sum$とする$(1 \leqq i, j \leqq k)$.

COVS(< $\Sigma_1 \Sigma_2 ... \Sigma_k$>,<$\Gamma_1\Gamma_2 ... \Gamma_k$>)= “*EVAL*(< $\Sigma_1 \Sigma_2 ... \Sigma_k$>) − *EVAL*(<$\Gamma_1\Gamma_2 ... \Gamma_k$>)” の最小汎化パターン集合=

$$\{ < \Sigma_1 \Sigma_2 ... \Sigma_{i-1}(\Sigma_i - \Gamma_i) ... \Sigma_k > \mid 1 \leqq i \leqq k \}$$

$\Gamma_i \neq \Sigma_i$ かつ $\Gamma_i \cap \Sigma_i \neq \varphi$ であれば，*COVS*(< $\Sigma_1 \Sigma_2 ... \Sigma_k$>,<$\Gamma_1\Gamma_2 ... \Gamma_k$>)は，*k* 個の汎化パターンを要素として持ち，$\Gamma_i \cap \Sigma_i = \varphi$ であれば，*COVS*(< $\Sigma_1 \Sigma_2 ... \Sigma_k$>,<$\Gamma_1\Gamma_2 ... \Gamma_k$>)={< $\Sigma_1 \Sigma_2 ... \Sigma_k$>}である$(1 \leqq i \leqq k)$．特に $\Gamma_i = \Sigma_i$ であれば，*COVS*(< $\Sigma_1 \Sigma_2 ... \Sigma_k$>, <$\Gamma_1\Gamma_2 ... \Gamma_k$>)=$\varphi$ となる．

3.2 反復精密化法

k-ミスマッチクラスタ{<mis_1^k>, <mis_2^k>, ..., <mis_m^k>}に対する最小汎化パターン集合の計算は，以下の 2 つの処理手順により行っている．

(1) 最汎パターン<$mgpat^k$>を精密化し，<$mgpat^k$>から *k*-ミスマッチクラスタ *MIS*={<mis_1^k>, <mis_2^k>,...,<mis_m^k>}を除去した結果に対する最小汎化パターン集合 {<neg_1^k>,<neg_2^k>,...,<neg_p^k>}(負の最小汎化パターン)を計算する．
(2) この負の最小汎化パターン集合を用いて，再度，最汎パターン<$mgpat^k$>を精密化し，<$mgpat^k$>から負の最小汎化パターンを除去した結果に対する正の最小汎化パターン集合を計算する．この計算結果は，*k*-ミスマッチクラスタの最小汎化パターン集合となる．

上記の(1)と(2)の方法それぞれの過程で，生成される *k*-汎化パターン集合{<pat_1^k>,<pat_2^k>,...,<pat_m^k>}において，ある *k*-パターン<pat_i^k>が別の *k*-パターン<pat_j^k>を被覆する場合，この集合は冗長である．よって，冗長性を取り除くため，各ステップで集合から<pat_j^k>の除去を行っている．

上記(1)の処理を実施するためには，*k*-ミスマッチクラスタの *i* 番目の要素の分解結果として得られる汎化パターン集合を用いて，*i*+1 番目の要素が含まれないような精密化を行う必要がある$(1 \leqq i \leqq m)$．合計 *m* 回の精密化後に得られる汎化パターン集合は，*k*-ミスマッチクラスタのどの要素<mis^k>も含まない負の汎化パターン集合になる．(2)の処理については，処理すべきデータが異なるだけで(1)の処理と同じである．この計算により，*k*-ミスマッチクラスタの最小汎化パターン集合が得られる．

汎化パターン<pat^k>をミスマッチクラスタのある要素<mis^k>が含まないように分解するには，<pat^k>から<mis^k>を除去した結果に対する最小汎化パターン集合 *COVS*(<pat^k>,<mis^k>)を計算することにより達成できる．また，汎化パターン<pat^k>を負の汎化パターン集合要素<neg^k>が含まないように精密化するには，<pat^k>から<neg^k>を除去した結果に対する最小汎化パターン集合 *COVS*(<pat^k>,<neg^k>)を計算することにより達成できる．

[例 1] 最汎パターンを<[*AD*][*BE*][*CF*]>, *PS*={<*ABF*>, <*AEC*>, <*AEF*>, <*DBF*>, <*DEC*>, <*DEF*> }とするとき，負の最小汎化パターン集合を計算した後に，正の最小汎化パターン集合を計算してみよう．

(1) 最汎パターン<[*AD*][*BE*][*CF*]>から<*ABF*>を除去した結果に対する最小汎化パターン集合は以下のとおりである．
COVS(<[*AD*][*BE*][*CF*]>,<*ABF*>)=
{ <*D*[*BE*][*CF*]>, <[*AD*]*E*[*CF*]>, <[*AD*][*BE*]*C*> }
(2) <[*AD*]*E*[*CF*]>と<[*AD*][*BE*]*C*>から<*AEC*>を除去した結果に対する最小汎化パターン集合はそれぞれ以下のとおりである．
COVS(<[*AD*]*E*[*CF*]>,<*AEC*>)={<*DE*[*CF*]>, <[*AD*]*EF*> }
COVS(<[*AD*][*BE*]*C*>,<*AEC*>)={<*D*[*BE*]*C*>, <[*AD*]*BC*> }
しかしながら，<*DE*[*CF*]>と<*D*[*BE*]*C*>は，上記(1)で得られた<*D*[*BE*][*CF*]>に含まれるので，削除する．
(3) <[*AD*]*EF*>から<*AEF*>の除去した結果に対する最小汎化パターン集合は空集合である．

(4) <*D*[*BE*][*CF*]>から<*DBF*>のを除去した結果に対する最小汎化パターン集合は以下のとおりである.

COVS(<*D*[*BE*][*CF*]>,<*DBF*>)={<*DE*[*CF*]>, <*D*[*BE*]*C*>}

(5) <*DE*[*CF*]>と<*D*[*BE*]*C*>に対する<*DEC*>のを除去した結果に対する最小汎化パターン集合は空集合である.

(6) 残った汎化パターンには<*DEF*> が含まれないので,最小汎化パターン集合の計算は不要である.

以上により，<[*AD*]*BC*>が負の最小汎化パターン集合となる．これを最汎パターン<[*AD*][*BE*][*CF*]>から除去すると，正の最小汎化パターン集合が得られる．(1)～(6)の探索過程を図 1 に示す.

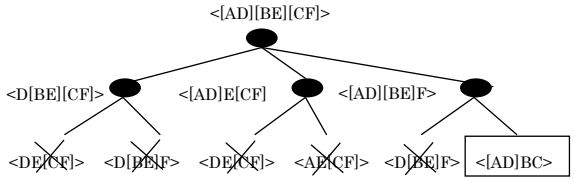

図 1 {<*ABF*>,<*AEC*>,<*AEF*>,<*DBF*>,<*DEC*>,<*DEF*>} に対する負の汎化パターンの計算例

Fig.1 An example of extracting the negative patterns from {<*ABF*>,<*AEC*>,<*AEF*>,<*DBF*>,<*DEC*>,<*DEF*>}

(7) 最汎パターン<[*AD*][*BE*][*CF*]> から<[*AD*]*BC*> を除去した結果に対する最小汎化パターン集合は以下のとおりである.

COVS(<[*AD*][*BE*][*CF*]>,<[*AD*]*BC*>)=
{<[*AD*]*E*[*CF*]>, <[*AD*][*BE*]*F*> }

従って， <[*AD*]*E*[*CF*]>,<[*AD*][*BE*]*F*> が *PS*={<*ABF*>,<*AEC*>, <*AEF*>,<*DBF*>,<*DEC*>,<*DEF*>}に対する正の最小汎化パターン集合となる.

3.3 反復精密化法とドメイン分割法の併用

ここでは，最悪で $O(2^{kl})$の計算量をもつ反復精密化法の計算時間を短縮するために，反復精密化法とドメイン分割法を併用することを提案する．ドメイン分割法は，問題領域特有の経験的知識が文字列間の非類似度を表す距離行列で与えられているとし，この距離行列を用いて最汎パターンに含まれる各曖昧文字の曖昧文字ドメインを分割する方法である．この分割法により,曖昧文字のドメイン(ドメインに含まれる文字要素数が l 個)が小さなサブドメインに分割されるので,最汎パターンを特殊化された複数個の汎化パターンに分割できる．これは,反復精密化法の計算時間の短縮につながる．以下に反復精密化法とドメイン分割法を併用した処理手順を提案する.

(1) 最汎パターン上に存在するそれぞれの曖昧文字に対して,階層併合的クラスタリングより，曖昧文字ドメイン内の文字を分類し，このドメインを複数のサブドメインに分割する.

(2) それぞれの曖昧文字ごとに分割された複数のサブドメインを用いて，最汎パターンから特殊化された複数個の汎化パターンを生成する.

(3) 特殊化された複数個の汎化パターンのそれぞれを反復精密化法に入力する新たな最汎パターンと見なし，それぞれに反復精密化法を適用することによりミスマッチクラスタを最小被覆する最小汎化パターンを抽出する.

4. 実験

反復精密化法とドメイン分割法の有効性を確認するため，PROSITE0[6]に含まれる 3 種類のデータセット，Ribosomal (登録番号:PS00051)，Kringle (登録番号:PS00021)，Zinc Finger (登録番号:PS00028)を用いて評価実験を行った．この評価実験に利用した計算機環境は，Intel Pentium(R)4-3.2GHz，メモリー：2GB，SWAP メモリー：2GB，HDD: 320GB，OS: CentOSv4.5 である．3 種類のデータセットの詳細は表 1 の通りである.

表 1 データセット詳細
Table 1 Characteristics of the three datasets

データセット	データ件数	総長(bytes)	モチーフ表現数	解の数
Ribosomal	44	2330	1728	17
Kringle	88	58453	32	8
Zinc Finger	467	245595	72	41

それぞれのモチーフは以下の形式で知られている.

Ribosomal :<[*KRS*]-[*PTKS*]-*x*(3)-[*LIVMFG*]-*x*(2)-[*NHS*]-*x*(3)-*R*-[*DNHY*]-*W*-*R*-[*RS*]>

Kringle : <[*FY*]-*C*-[*RH*]-[*NS*]-*x*(7,8)-[*WY*]-*C*>

Zinc Finger : <*C*-*x*(2,4)-*C*-*x*(3)-[*LIVMFYWC*]-*x*(8)-*H*-*x*(3,5)-*H*>

表 1 のモチーフ表現数はモチーフに含まれるインスタンス数,解の数はデータセットに含まれるモチーフのインスタンス数を表している.

評価実験を行うため，曖昧パターン検索を行い，配列データベースからミスマッチクラスタを求める必要がある．ここでは，各々のデータセットに対して，DynaCluster[7]と呼ばれるサフィックス木構築アルゴリズムを用いて，ディスク上にサフィックス木を構築し，構築したサフィックス木を深さ優先にスキャンすることでミスマッチクラスタを求める.

以下では，(1)反復精密化法の効果と，(2)ドメイン分割法の効果を検証するため，データセットに対して，曖昧パターン検索を行い，得られたミスマッチクラスタに対して，ドメイン分割法を併用した反復精密化法と，ドメイン分割法を併用しない反復精密化法を実行した． なお，ドメイン分割法ではアミノ酸置換行列 PAM250 を使用し，階層併合的クラスタリングの手法として最長距離法を用いた

4.1 実験結果

(1) 反復精密化法の効果

曖昧パターン検索の検索条件，結果，およびドメイン分割法を併用した反復精密化法の結果を表 2 に示す.

表 2 曖昧パターン検索と反復精密化法の結果
Table 2 Results of ambiguous queries and iterative refinement method

データセット	検索パターン 許容誤差	検索結果 (個)	汎化結果 (個)	減少率 (%)
Ribosomal	<*K*-*P*-*x*(3)-*L*-*x*(2)-*N*-*x*(3)-*R*-*D*-*W*-*R*-*R*> 6	109	48	56
Kringle	<*Y*-*C*-*R*-*N*-*x*(7,8)-*W*-*C*> 4	3552	1152	68
Zinc Finger	<*C*-*x*(2,4)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(3,5)-*H*> 1	2729	102	96

表 2 のパターン数を比較および減少率を見れば，明らかなように，反復精密化法を適用することで大量のミスマッチクラスタを少数の汎化パターン集合で表現することに成功した．なお，ここで得られた汎化パターン集合は曖昧パターン検索によって得られたミスマッチクラスタ内の全ての要素を被覆していることをプログラムで確認している．反復精密化法により,汎化することで,閲覧するデータ数が減少するので,単にミスマッチクラスタを閲覧するより，全体像の把握が比較的容易になる可能性が高まる.

そこで Ribosomal を用いた実験でドメイン分割法を併用した反復精密化法を実行して得られた 48 件の汎化パターンを支持率別にランキングを行った.表 3 に,上位 7 件を示す.表 3 の解の数は，パターンのインスタンスの中でモチーフに含まれるインスタンス数を表している．表 3 より，支持率の

高いパターンがモチーフに関係していることがわかる．反復精密化法により得られた 48 件の汎化パターンから支持率の高い上位にランキングされたパターンに注目することで，モチーフに強く関係したパターンを参照することができ，データセットの性質を知ることができるといえる．ドメイン分割法を併用しないで反復精密化法を実行した場合，また他のデータセットを用いた実験でも同様の結果が得られている．
反復精密化法により，ミスマッチクラスタ集合が汎化され，支持率の高い汎化パターンを参照することで全体像の把握が容易になったといえる．

表 3　汎化パターン支持率別ランキング
Table 3 Support count rate ranking of generalized patterns

ランク	パターン	インスタンス数	解の数	支持率(%)
1	<[KR]-T-x(3)-[IV]-x(2)-N-x(3)-R-[HM]-W-R-R>	8	8	56.10
2	<K-T-x(3)-[LMV]-x(2)-N-x(3)-R-H-W-R-R>	3	3	27.27
3	<[KR]-T-x(3)-V-x(2)-[HN]-x(3)-R-H-W-R-R>	4	4	20.45
4	<K-T-x(3)-V-x(2)-[HNS]-x(3)-R-H-W-R-R>	3	3	20.45
5	<P-[FL]-x(3)-L-x(2)-A-x(3)-K-Q-N-R-R>	2	0	11.36
6	<K-Q-x(3)-V-x(2)-W-x(3)-[KR]-T-N-R-R>	2	0	6.82
7	<R-[ST]-x(3)-I-x(2)-N-x(3)-R-H-W-R-R>	2	2	4.55

(2)ドメイン分割法の効果
以下では，反復精密化法にドメイン分割法を併用した場合とそうでない場合の計算結果を示す．

表 4　ドメイン分割法の併用の有無による比較
Table 4 Comparison by concomitantly-used of the domain segmentation method with or without

	ドメイン分割法(無)		ドメイン分割法(有)	
データセット	パターン数	処理時間(sec)	パターン数	処理時間(sec)
Ribosomal	45	3605.2	48	45
Kringle	---	---	1152	1358.2
Zinc Finger	38	2.03	102	1.05

表 4 よりドメイン分割法を併用することで処理時間が高速になったことがわかる． Kringle を用いた実験では，ドメイン分割法を併用しないで反復精密化法を実行した場合，3 日以上の実験で終了しなかったため測定できなかった．一方，ドメイン分割法を併用した反復精密化法では，高速に結果が得られた．このときに，ミスマッチクラスタから計算された最汎パターンは，全ての曖昧文字ドメイン $\Sigma_i(1 \leqq i \leqq 6)$が 20 種類のアミノ酸文字を持つパターンであった．ドメイン分割法を併用したところ，各曖昧文字ドメインは{*C*}, {*I,L,M,V*}, {*D,E,H,Q*}, {*F,W,Y*}, {*K,N,R,S*}, {*A,G,P,T*}に分割された．これにより，クラスタ間にまたがった文字どうしに関する計算を省くことができるため，高速になったと考えられる．特に，各曖昧文字ドメイン Σ_i が膨大な曖昧文字を持つ場合，ドメイン分割法が処理時間に与える効果が大きいことがわかる．
その他，ドメイン分割法の併用の有無により得られる汎化パターンの違いを調べるために，Zinc Finger を用いて実験を行った．その結果，ドメイン分割法を併用しないときに得られたある汎化パターンには，モチーフに含まれるインスタンスと含まれないインスタンスが混在したが，ドメイン分割法を併用することにより，両インスタンスは 2 つの汎化パターンとして分離された．これより，反復精密化法とドメイン分割法を併用することは，背景知識の利用につながり，モチーフの発見の可能性を高めると期待できる．

5. まとめと今後の課題

配列データベースから曖昧配列パターン集合を効率的に抽出するため，反復精密化法を提案し，それとドメイン分割法を併用する方法を提案した．提案手法の有効性を確認するため，PROSITE にある 3 種類のデータセットを用いて実験を行った．実験の結果，それぞれの実験で，曖昧パターン検索によって得られたミスマッチクラスタを少数の汎化パターンで表現することに成功した．また，ドメイン分割法を併用することで，高速な処理が可能になった．汎化パターンを抽出し，ミスマッチクラスタの全体像を容易に把握することができるほか，支持率別にランキングすることで，データセットに含まれる新たなモチーフの発見に役立つ可能性が高まることが確認できた．

今後は，可変長を考慮した曖昧配列パターンの抽出を行っていく予定である．

[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)(一般)，課題番号:17500097)の支援により行われた．

[文献]
[1] Zvi Galil, Kunsoo Park: An Improved Algorithm for Approximate String Matching, SIAM Journal on Computing, Vo.19, No.6, pp.989-999 (1990).
[2] Sanghyun Park, Wesley W. Chu, Jeehee Yoon, Chihcheng Hsu: Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases, Proceedings of 16th International Conference on Data Engineering (ICDE2000), IEEE Computer Society Press, pp.23-32 (2000).
[3] Marie-France Sagot: Spelling Approximate Repeated or Common Motifs Using a Suffix Tree, Theoretical Informatics, Third Latin American Symposium (LATIN 1998), pp.374-390 (1998).
[4] Pevzner,P.A. and Sze,S.: Combinatorial approaches to finding subtle signals in DNA sequences, Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, pp. 269-278 (2000).
[5] Eleazar Eskin, Pavel A. Pevzner: Finding composite regulatory patterns in DNA sequences, Proceedings of the 10th International Conference on Intelligent Systems for Molecular Biology, pp.269-278, pp.354-363 (2002).
[6] http://kr.expasy.org/prosite/.
[7] Ching-Fung Cheung, Jeffrey Xu Yu, Hongjun Lu: Constructing Suffix Tree for Gigabyte Sequences with Megabyte Memory, IEEE Transaction Knowledge Data Engineering, Vo.17, No.1, pp.90-105 (2005).

荒木 康太郎　Kotaro ARAKI
広島市立大学大学院情報科学研究科博士前期課程在学中．日本データベース学会，情報処理学会，各学生会員．
田村 慶一　Keiichi TAMURA
広島市立大学大学院情報科学研究科助教. 2000　九州大学大学院システム情報科学研究科修士課程修了. 博士(情報科学).
加藤 智之　Tomoyuki KATO
広島市立大学大学院情報科学研究科博士前期課程在学中．日本データベース学会，電子情報通信学会，各学生会員．
北上 始　Hajime KITAKAMI
広島市立大学大学院情報科学研究科教授. 1976　東北大学大学院工学研究科博士前期課程修了．博士(工学)．データベースおよび人工知能などの研究開発に従事．日本データベース学会 BI 研究グループ運営委員，人工知能学会評議員，情報処理学会(TOM)論文誌編集委員, IEEE および ACM 各会員．