社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MPS－72 (19)
2008－BIO－15 (19)
2008／12／17

ミスマッチクラスタを表現する最小汎化集合の高速抽出

澤田 祐介，田村 慶一，荒木 康太郎，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

あらまし 配列データベースに対する曖昧な問合せ処理では，非常に多くの類似した部分文字列の集合（ミスマッチクラスタ）が結果として得られる．そのため，ユーザが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困難である．著者らは，ミスマッチクラスタから曖昧な文字表現を含む最小汎化集合を高速に抽出するために，ドメイン分割法を併用した反復精密化法とその並列化方式を提案する．提案手法の有効性を評価するために，アミノ酸モチーフの抽出問題に適用して実験を行ったので，その実験結果についても報告する．

Fast Extracting Minimum Generalized Set of Mismatch Cluster

Yusuke SAWADA, Keiichi TAMURA, Kotaro ARAKI, and Hajime KITAKAMI

Graduate School of Information Sciences, Hiroshima City University, 3-4-1 Ozukahigashi Asaminami-ku, Hiroshima, 731-3194, Japan

Abstract An ambiguous query in sequence database returns a set of similar subsequences, called a mismatch cluster, to the user. Therefore, the user browses the mismatch clusters, and it is difficult to comprehend the mismatch clusters. The authors propose an iterative refinement method to extract the minimum generalized set including the ambiguous sequence patterns from mismatch clusters. In addition, the authors propose the use of both iterative refinement method in combination with the domain segmentation method and parallelization method to achieve a fast extracting the minimum generalized set. Moreover, the authors apply the three proposed methods to the problem of extraction of an amino acid motif and experimented it to evaluate the effectiveness, then we report about the experimental results.

1. はじめに

曖昧な問合せ処理は，テキストデータや配列データからの類似する部分文字列の検索をさし，Web 文書，オンライン文書，分子配列データなどに対する情報検索を初めとして，クラスタリングや配列データマイニングなど多くの分野で重要な要素技術である．曖昧な問合せ処理は，わずかなゆらぎを持つモチーフが含まれる分子配列（DNA 配列やアミノ酸配列）データなどに対する類似検索に有用である．モチーフとは，PROSITE[1]や Pfam などで見られる生物学的に重要な機能を持つ特徴的なパターンである．

今までに，数多くの曖昧な問合せ処理の研究が行われてきたが，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-部分文字列をすべて求めるどけにとどまっている．本論文では，この曖昧な問合せ処理により選択される部分文字列集合をミスマッチクラスタと呼ぶ．そして，ユーザがこのミスマッチクラスタを閲覧し，規則的な特徴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．

我々のゴールは，多数の類似部分文字列を，極大な汎化配列パターンと汎化できなかった類似部分文字列のすべてから成る集合（以後，最小汎化集合と呼ぶ．）で表現するために，ミスマッチクラスタに対する高速な汎化処理を達成することである．そこで，(1)最小汎化集合を抽出する方法として，トップダウン探索のアプローチに基づく反復精密化法[2]を，(2)高速化手法として，ドメイン分割法と反復精密化法の併用方法[2]とその並列化手法を提案する．

それぞれの特徴は，以下のとおりである．

(1) 反復精密化法

反復精密化法では，最初に，負の最小汎化集合を求め，次に，その計算結果をもとに正の最小汎化集合を求めるという2段階で，最小汎化集合を求めていく．ミスマッチクラスタに対して，最も一般的な配列パターン（最汎パターン）を列挙木のルートとみなし，ルートから下方に探索を進める方法となっている．

(2) ドメイン分割法と反復精密化法の併用方法とその並列化手法

反復精密化法を使用することで，最小汎化集合を求めることができるが，解がミスマッチクラスタと大差がない場合，解の探索に多くの時間を費やしてしまう可能性がある．そこで，ドメイン分割法と反復精密化法の併用を行う．ドメイン分割法では，ユーザの背景知識を利用して，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する．そして，特殊化された複数個の汎化パターンごとに，反復精密化法を実行することで最小汎化集合を求める．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することで，無駄な探索を省くことができるので，高速化が期待できる．また，特殊化された汎化パターンごとの反復精密化法

は独立に実行できるため，マスタ・ワーカモデルにより並列化を行うことで，さらなる高速化が可能となる．

2. 用語と記号の定義

2.1. ミスマッチクラスタ

曖昧な問合せでは，ユーザが長さ k を持つある部分文字列（k-部分文字列）と許容誤差 r を検索条件として与える．この問合せにより配列データベースから検索条件を満たす部分文字列<$inst$>の集合が得られる．この集合をミスマッチクラスタ MIS と呼ぶ．以後，ミスマッチクラスタを次の形式で表現する．

$$MIS=\{<inst_1>,<inst_2>,\ldots,<inst_n>\} \quad (1)$$

2.2. 曖昧性を表現する汎化配列パターン

$\sum_i$ をアルファベット$\sum$の部分集合とすると，k-汎化配列パターン（k 個の$\sum_i$ を並べたパターン）を<pat^k>と書き，以下の形式で表現する．

$$<pat^k>=<\sum_1\text{-}x(i_1,j_1)\text{-}\sum_2\text{-}x(i_2,j_2)\text{-}\ldots\text{-}\sum_{k-1}\text{-}x(i_{k-1},j_{k-1})\text{-}\sum_k> \quad (2)$$

ただし，$\sum_i$ は，たびたび括弧[]の中に$\sum_i$ の全要素を列挙した表記をする．式(2)中の，$\sum_i \subseteq \sum$が存在する場所を曖昧文字領域と呼ぶ．また，$|\sum| \geqq 2$ のとき，集合$\sum_i$ は曖昧文字ドメインと呼ばれ，$\sum_i$ 内に存在する任意の 1 文字の配置が許されていることを示している．曖昧文字ドメインが 1 箇所以上存在するとき，式(2)を k-汎化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，$x(i,j)$は $x(i)$と書く．

2.3. インスタンスを導出する関数

k-汎化配列パターン<$\sum_1\sum_2\ldots\sum_{k-1}\sum_k$ >からインスタンスのすべて（部分文字列の集合）を導出する関数を以下のように表記する．

$$EVAL(\{<\sum_1\sum_2\ldots\sum_{k-1}\sum_k>\}) \quad (3)$$

式(3)の括弧{}は<pat^k>の集合を意味し，n 個の<pat^k>から成る集合を $P^k=\{<pat_1^k>,<pat_2^k>,\ldots,<pat_n^k>\}$と書く．例えば，{<$pat^k$>}={<[$AD$]$BC$>}のとき，$EVAL(\{<pat^k>\})$ ={<ABC>,<DBC>}となる．

2.4. 最汎パターンと最小汎化集合

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．$1 \leqq j \leqq k$ に対して，アルファベット$\sum_j$ を以下のように定義する．

$$\sum_j=\{inst[j] | inst \in I^k\} \quad (4)$$

このとき，<$\sum_1\sum_2\ldots\sum_{i-1}\sum_i\ldots\sum_k$>を k-汎化配列パターン集合 I^k に対する最汎パターンと呼び，この最汎パターンを $MGP(I^k)$と表記する．

ある集合 $MGS=\{G_1,G_2,\ldots,G_m\}$が，$k$-汎化配列パターン<$pat^k$>および k-インスタンス<$inst^k$>から構成されているとしよう($1 \leqq m \leqq |MIS|$)．ただし，$EVAL(\{<pat^k>\}) \subseteq MIS$ かつ<$inst^k$>$\in MIS$ を満たすものとする．

この集合 MGS が以下の性質を満たすとき，MGS を MIS に対する最小汎化集合（または，正の最小汎化集合）と呼ぶ．

(1) $EVAL(MGS)=MIS$ が成立する．
(2) MGS の任意の 2 要素 G_i,G_j に対して，G_i と G_j の間には冗長な関係が存在しない($1 \leqq i \neq j \leqq m$).
(3) MGS に含まれるどの要素 G_i も極大（さらに汎化すると MIS に存在しないインスタンスを含んでしまう）である($1 \leqq i \leqq m$).
(4) 上記の(1)～(3)を満たす任意の MGS'に対して，$|MGS'| \leqq |MGS|$が成立する．

3. パターン切除操作

ここでは，単一パターン切除操作とそれを複数回実施する多重パターン切除操作について述べる．

k-汎化配列パターン<pat^k>と k-インスタンス<$inst^k$>があるとしよう．<pat^k>から<$inst^k$>をパターン切除する単一パターン切除操作 $PCUT$ は，以下の関係式が成り立つ．

$$PCUT(<pat^k>,<inst^k>)=EVAL(\{<pat^k>\})-\{<inst^k>\} \quad (5)$$

例えば，$PCUT$(<[AD][BE][CF]>,<ABF>)の計算結果は，{<D[BE][CF]>,<[AD]E[CF]>,<[AD][BE]C>}となる．

<pat^k>から P^k をパターン切除する多重パターン切除操作 $MCUT$ は，以下の関係式が成り立つ．

$$MCUT(<pat^k>,P^k)=EVAL(\{<pat^k>\})-EVAL(P^k) \quad (6)$$

4. 反復精密化法

本節では，反復精密化法の処理手順およびその計算例について述べる．

4.1. 処理手順

反復精密化法は，MIS に対する最小汎化集合を計算する方法であり，以下の処理手順から成る．

(1) ミスマッチクラスタ MIS に対する最汎パターン<$mgpat^k$>=$MGP(MIS)$を計算する．
(2) <$mgpat^k$>から MIS 内のすべてのインスタンスをパターン切除することで，負の最小汎化集合（$EVAL$(<$magpat^k$>)－MIS に対する最小汎化集合）を計算する．
(3) <$mgpat^k$>から(2)で計算した負の最小汎化集合をパターン切除することで，正の最小汎化集合（MIS に対する最小汎化集合）を計算する．この正の最小汎化集合が，MIS を表現する最小汎化集合となる．

ただし，処理(2)，処理(3)のパターン切除では，3 章で述べた多重パターン切除操作を用いる．また，この処理手順では，多重パターン切除の途中で生成される冗長なパターンは削除されるものとする．

4.2. 計算例

以下では，反復精密化法により，ミスマッチクラスタ MIS を表現する最小汎化集合の計算例を示す．6 要素を持つ MIS を{<ABF>,<AEC>,<AEF>,<DBF>,<DEC>,<DEF>}とすると，この最汎パターンは<[AD][BE][CF]>となる．式を用いて，負の最小汎化集合の計算例を示すと，以下の(1)～(6)のようになる．計算結果として得られた負の最小汎化集合を用いて，正の最小汎化集合を計算すると，以下の(7)のようになる．

(1) 最汎パターン<[AD][BE][CF]>から<ABF>をパターン切除する．
$PCUT$(<[AD][BE][CF]>,<ABF>)
={<D[BE][CF]>,<[AD]E[CF]>,<[AD][BE]C>}
(2) 上記の結果に対して，<AEC>の切除を行う．<AEC>をインスタンスとして持つのは<[AD]E[CF]>と<[AD][BE]C>の 2 つであるので，こ

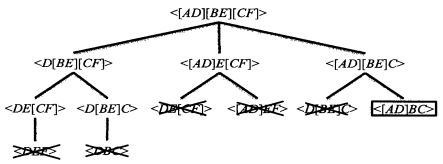

図 1 {<*ABF*>,<*AEC*>,<*AEF*>,<*DBF*>,<*DEC*>,<*DEF*>} に対する負の最小汎化集合の計算例

れらから<*AEC*>の切除を行う．

PCUT(<[*AD*]*E*[*CF*]>,<*AEC*>)
={<*DE*[*CF*]>,<[*AD*]*EF*>}
PCUT(<[*AD*][*BE*]*C*>,<*AEC*>)
={<*D*[*BE*]*C*>,<[*AD*]*BC*>}

しかしながら，<*DE*[*CF*]>と<*D*[*BE*]*C*>は，(1)で得られた<*D*[*BE*][*CF*]>に含まれるので，両者を削除する．

(3) 上記で残ったパターンの集合に対して，<*AEF*>の切除を行う．<*AEF*>をインスタンスとして持つのは<[*AD*]*EF*>であるので，これから<*AEF*>を切除する．しかし，切除した結果は *MIS* の要素<*DEF*>となるため，削除される．

(4) 続いて，<*DBF*>の切除を行う．<*DBF*>をインスタンスとして持つの<*D*[*BE*][*CF*]>であるので，これから<*DBF*>を切除する．

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

(5) 続いて，<*DEC*>の切除を行う．<*DEC*>をインスタンスとして持つのは，<*DE*[*CF*]>と<*D*[*BE*]*C*>であるので，これらから<*DEC*>を切除する．しかし，切除した結果は<*DEF*>∈*MIS* と<*DBC*>*not*∈*MIS* となるが，<*DBC*>は残っている<[*AD*]*BC*>に冗長なため削除される．

(6) 残った<[*AD*]*BC*>には，<*DEF*>が含まれないので，切除は不要である．

以上により，{<[*AD*]*BC*>}が負の最小汎化集合となる．(1)〜(6)の処理を**図 1** に示す．

次に，正の最小汎化集合を多重パターン切除により計算してみよう．

(7) 最汎パターン<[*AD*][*BE*][*CF*]>から負の汎化配列パターン<[*AD*]*BC*>を切除する．

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

したがって，{<[*AD*]*E*[*CF*]>,<[*AD*][*BE*]*F*>}が *MIS* を表現する最小汎化集合となる．

5. 最小汎化集合の高速抽出

本章では，最小汎化集合を高速に抽出するための，ドメイン分割法を併用した反復精密化法とその並列化方式について説明する．

5.1. ドメイン分割法を併用した反復精密化法

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領域の曖昧文字ドメインを分割する方法である．本研究では，距離行列として，アミノ酸文字間の類似度を表す PAM250 を各文字間の距離（非類似度）を表す距離行列へ変換して用いた．

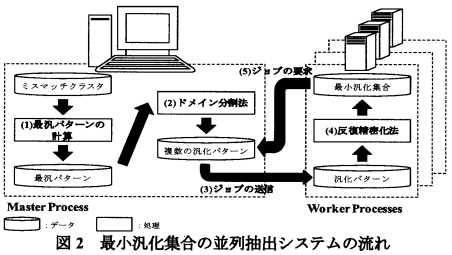

図 2 最小汎化集合の並列抽出システムの流れ

階層併合的クラスタリングにより，最汎パターン中の各曖昧文字ドメイン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することができる．ドメイン分割法と反復精密化法の併用方法は，この特殊化された複数個の汎化パターンそれぞれに対して反復精密化法を行うことで達成される．それぞれの汎化パターンで抽出された最小汎化集合の和がミスマッチクラスタを表現する最小汎化集合となる．これにより，クラスタ間にまたがった文字に関する計算を省くことができるため，反復精密化法の計算時間の短縮につながる．

5.2. 最小汎化集合の並列抽出方式

ドメイン分割法と反復精密化法の併用方法の並列化は，連続した反復精密化法の総タスクを複数のプロセッサ上で分散並列計算することで達成される．

最小汎化集合を並列抽出するために，マスタ・ワーカモデルを適用する．本方式のシステム構成を**図 2** に示す．マスタプロセスは，ジョブの生成と並列処理の管理を担当する．ここで，ジョブとは，ドメイン分割法によって生成された複数の汎化パターンのそれぞれを指す．各ワーカプロセスは，マスタプロセスから受け取ったジョブを入力データとして反復精密化法を独立して実行する．これらの処理は，ジョブプールが空になるまで繰り返し実行される．すべてのジョブに対する反復精密化法が終了したとき，各ワーカプロセスで抽出された最小汎化集合の和が，ミスマッチクラスタを表現する最小汎化集合となる．

マスタプロセスとワーカプロセスのデータの送受信には，ソケット通信を利用している．また，ジョブを動的に各ワーカプロセスに配布するために，マルチスレッド方式を適用している．

本方式では，ドメイン分割法を併用した反復精密化法の総タスクを複数台のワーカプロセス上に均等に分散することで，ワーカプロセス台数倍の計算時間の短縮が期待できる．

6. 評価実験

提案手法の有効性を確認するため，PROSITE に含まれるデータセット *Zinc Finger*, *Leucine*, *Kringle* を用いて評価実験を行った．実験は，最初に，それぞれのデータセットに対して，ディスクベースサフィックス木を構築し，曖昧な問合せを行うことでミスマッチクラスタを求める．次に，反復精密化法，ドメイン分割法を併用した反復精密化法，最小汎化集合の並列抽出の 3 つの手法を用いて，ミスマッチクラスタの最小汎化集合を計算する．これにより，3 つの提案手法の有効性を汎化結果と計算時間の 2 つの観点から考察を行う．

表 1　曖昧な問合せ結果汎化処理の結果

データセット	問合せ文字列 許容誤差	問合せ 結果(件)	汎化結果 (件)
Zinc Finger	<*Cx*(2,4)*Cx*(3)*Lx*(8)*Hx*(3,5)*H*> 1	2729	38(74)
Leucine	<*PLxLx*(2)*LAx*(2)*HxSTLSR*> 5	31	14(14)
Kringle	<*YCRNx*(7,8)*WC*> 4	3552	(1097)

表 2　ドメイン分割法の併用の有無による計算時間

データセット	ドメイン分割法（無）の計算時間	ドメイン分割法（有）の計算時間
Zinc Finger	0.228	0.264
Leucine	361.5	0.345
Kringle	-	42.40

実験で使用した PC は，CPU: Intel Core Duo E6600(2.4GHz)，Memory: 1GB×2，HDD: 250GB を搭載している．最小汎化集合の並列抽出の実験では，分散並列環境として 4 台の PC から構成される PC クラスタを使用した．

6.1. 反復精密化法の有効性

実験に用いた曖昧な問合せに対する検索結果，汎化処理の結果を**表 1** に示す．表 1 の汎化結果件数の括弧内の数字は，ドメイン分割法を併用した反復精密化法によって得られた要素数を示している．

表 2 の類似部分文字列の数と最小汎化集合の要素数を比較すると，反復精密化法によって大量のミスマッチクラスタを少数の最小汎化集合で表現することに成功している．

ここで，それぞれのデータセットに対する反復精密化法によって抽出された最小汎化集合の一部を以下に記載する．

(a) *Zinc Finger*: <*Cx*(2)*Cx*(3)*Lx*(8)*Hx*(3)[*GHLNQTV*]>
(b) *Leucine*: <*PLxLx*(2)*VAx*(2)[*ILV*]*x*(2)*HxSTVSR*>
(c) *Kringle*: <*YCRNx*(7)[*WY*]>

抽出されたパターンを閲覧すると，一つの汎化パターンで，複数の部分文字列を表現していることがわかる．よって，全体像の把握が比較的容易になる可能性が高まる．

6.2. ドメイン分割法の有効性

計算時間に関する結果を**表 2** に示す．それぞれのデータセットにおける実験に注目すると，*Zinc Finger* は，ほとんど計算時間に変化が見られなかった．*Leucine* は，1048 倍の高速化に成功している．*Kringle* は，反復精密化法では計算を終えることができなかったのが，計算を終了することに成功している．

Zinc Finger は，ドメイン分割法によって，最汎パターンが 48 個の汎化パターンに分割される．これにより，クラスタ間にまたがった計算を省くことができるが，曖昧文字ドメイン数が少ないため，計算時間の短縮には至らなかったといえる．*Leucne*，*Kringle* は，それぞれ 8 個，46656 個の汎化パターンに分割される．これにより，クラスタ間にまたがった計算を省くことができ，計算時間を短縮することができたといえる．特に，*Kringle* は，最汎パターンの各曖昧文字ドメインに多くのアミノ酸文字が存在するため，クラスタ間にまたがった計算を大幅に削減することができた．

次に，ドメイン分割法併用の有無によって得られる最小汎化集合の内容を比較するために，*Zinc Finger* の実験で得られた最小汎化集合を比較し検証した．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 つの汎化パターンに分離された．これにより，ドメイン分割法と反復精密化法の併用方法は，背景知識の利用につながり，モチーフ発見の可能性を高める．

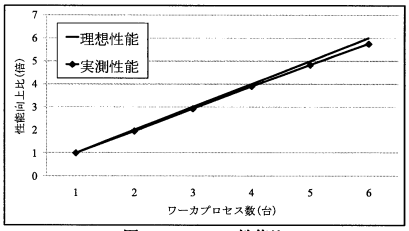

図 3　*Kringle* の性能比

6.3. 並列抽出の有効性

最小汎化集合をさらに高速に抽出するために，本実験では，*Kringle* の最小汎化集合を並列抽出した．本実験で扱うデータは，PC1 台のメモリ領域で十分計算が可能である．よって，マルチプロセッサの特性から，3 台の PC にそれぞれ 2 つのワーカプロセスを割り当てた．残りの 1 台はマスタプロセスを割り当てた．

実験結果を**図 3** に示す．図 3 のワーカプロセス数 1 台のときの計算時間は，表 2 の計算時間である．また，抽出された最小汎化集合は，ワーカプロセス数 1 台から 6 台まで等しい結果が得られている．

図 3 から，理想値に近い並列性能が得られていることがわかる．これは，ドメイン分割法を併用した反復精密化法の総タスクが各ワーカプロセス上で均等に分割できているからである．

7. おわりに

本稿では，曖昧な問合せ処理結果として返されるミスマッチクラスタを表現する最小汎化集合を抽出ために反復精密化法を提案した．また，高速に最小汎化集合を抽出するために，ドメイン分割法を併用した反復精密化法とその並列化方式を提案した．

提案手法の有効性を確認するため，3 種類のデータセットを用いて実験を行った．その結果，反復精密化法によって，ミスマッチクラスタを表現する最小汎化集合の抽出に成功した．さらに，ドメイン分割法と反復精密化法の併用方法とその並列化によって，計算時間の高速化に成功した．

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:20500137），および，文部科学省・科学研究費補助金（課題番号:20700095）の支援により行われた．

参考文献

[1] http://kr.expasy.org/prosite/.

[2] K. Araki., K. Tamura., T. Kato., Y. Mori., and H. Kitakami.: Extraction of ambiguous sequential patterns with least minimum generalization from mismatch clusters, *The third international conference of SITIS'2007*, *IEEE CSP*, pp.32-39 (2007).