極小な可変長ワイルドカード領域を持つ頻出配列パターンの抽出

Extraction for Frequent Sequential Patterns with Minimum Variable-Wildcard Regions

加藤 智之† 北上 始‡ 森 康真‡
田村 慶一‡ 黒木 進‡

Tomoyuki KATO Hajime KITAKAMI
Yasuma MORI Keiichi TAMURA
Susumu KUROKI

著者らは，アミノ酸配列データベースからモチーフの候補である頻出パターンを抽出するために，極小な可変長ワイルドカード領域を持つ頻出パターンを導き出す方法を提案する．この方法では，k-頻出パターン(長さ k の頻出パターン)から(k+1)-頻出パターンを生成するパターン成長アプローチを拡張し，k-頻出パターンごとにスコープデータベースを作成する．スコープデータベースの有効性を示すために，PROSITE から Leucine Zipper モチーフを含むデータセットを取り出し，可変長の頻出パターンを抽出する能力の評価を行ったので，その結果について報告する．

We propose a method for extracting frequent sequential patterns with minimum variable-wildcard regions, in order to extract candidates of a motif from amino acid sequence databases. A scope database defined by each frequent k-length pattern is constructed by the extension of Projected Database that generate frequent (k+1)-length patterns from a frequent k-length pattern in pattern growth approach. Moreover, we report experimental results that our extended method was evaluated using a dataset that includes the Leucine Zipper motif.

1. はじめに

近年，データマイニングの立場から，DNA やアミノ酸などの分子配列データから取り出した特徴的なパターンによりモチーフを発見する手法が注目されており，分子配列データベースから頻出パターンを正確に抽出することが重要な課題となっている．モチーフとは，生物進化の過程で保存されてきた生物学的に重要な機能をもつパターンである．

本論文では，モチーフ発見を支援するために，アミノ酸配列データベースからさまざまなワイルドカード領域を含む頻出パターンの抽出法を提案する．既に可変長ワイルドカード領域を含む頻出パターン抽出法として，PrefixSpan法[2]を拡張した可変長パターン抽出法[4]が提案されている．しかし，可変長パターン抽出法には以下の問題が生じている．

(1) 可変長ワイルドカード領域の非極小性
可変長パターン抽出法では，頻出パターンに含まれる可変長ワイルドカード領域は，それ自身の証拠に該当するオカーレンス集合を過度に説明してしまう表現になることがあるので，その集合を極小被覆する表現に必ずしもならない．

(2) 可変長ワイルドカード領域の冗長性
可変長パターン抽出法では，パターン上の同じ位置関係にある可変長ワイルドカード領域だけが異なる冗長なパターンが複数抽出されるが，冗長なパターンが除去されていない．例えば，2 つの領域間が$[i_1,j_1]\subset[i_2,j_2]$の関係にあるとき，パターン<A-x($i_1$,$j_1$)-F>は，パターン<A-x($i_2$,$j_2$)-F>に冗長であるにもかかわらず，除去されていない．

これらの問題を解決するために，本論文では，頻出パターンの成長させる度に，追加すべき，極小で非冗長な可変長ワイルドカード領域を計算するだけでなく，それまでに生成されている可変長ワイルドカード領域を再計算する機能を組み込んだ新しい手法を提案する．

2. 関連研究

頻出パターンを抽出する方法であるが，パターン成長アプローチを用いたPrefixSpan法[2]は，モチーフ中に存在するワイルドカード領域の抽出を考慮していないので，余分なパターンが多く抽出され、多大な計算時間を要するという問題があった．この問題を解決するために，PrefixSpan法にワイルドカード領域を抽出する仕組みを導入した固定長パターン抽出法[3]が提案された．この方法は，PrefixSpan法に，最大ワイルドカード数と呼ばれる入力パラメータを追加したもので，固定長ワイルドカード領域を持つ頻出パターンの抽出を可能とした．また，可変長パターン抽出法[4]は,固定長パターン抽出法に最大誤差数と呼ばれる入力パラメータを付加することで,可変長ワイルドカード領域を持つ頻出パターンの抽出を可能としている．しかし，可変長パターン抽出法には，頻出パターンに含まれる可変長ワイルドカード領域がオカーレンス集合を過度に説明する表現として求まってしまうという問題がある．さらに，抽出される頻出パターンの集合が冗長であるという問題もある．

3. 用語と問題の定義

配列データベースDB=$\{t_1,t_2,\cdots t_n\}$において，各要素は，<sid,s_{sid}>と表現される（nは要素数，sidは配列識別子）．配列データベースの配列識別子の集合は$\Omega=\{1,2,3,\cdots,n\}$と表現する．各$s_{sid}$は，配列識別子の値としてsidを持つ配列データと定義する．配列データs_{sid}の先頭からｊ番目の文字は，$s_{sid}[j]$として表される．ある部分配列データのアルファベット文字数がk個であるなら，その配列データをｋ-配列データと呼ぶ．アルファベット文字及び，記号＊で表現されるワイルドカード文字（以下,ワイルドカードと呼ぶ）の両者から作成される有限列は，ストリングと呼ばれる．ただし，ストリングの両端は，アルファベット文字に限定する．ワイルドカードとは，任意の１文字を表す記号である．ストリングに対するアルファベット文字の数がk個であれば，そのストリングは，k-ストリングと呼ばれる．

† 学生会員 広島市立大学大学院情報科学研究科
kato@db.its.hiroshima-cu.ac.jp
‡ 正会員 広島市立大学情報科学部
{kitakami,mori,ktamura,kuroki}@its.hiroshima-cu.ac.jp

3.1 パターンとオカーレンスの関係

パターンとは，複数の配列データに共通に含まれている，あるk-ストリング集合に対する表現形式である(k≥1)．このk-ストリングをk-stringと表記しよう．あるk-stringが配列データベースDB上に存在する位置情報を追加した三項関係{(k-string,i,j)|k-stringはs_iのj番目に存在し，<i,s_i>∈DB}をk-オカーレンス集合と呼ぶ．k個の文字を持つk-パターン<pat^k>は，あるk-オカーレンス集合を表現するために与えられる形式であり，以下の形式で表現される．

$<pat^k>=<\alpha_1\text{-}x(i_1,j_1)\text{-}\alpha_2\text{-}x(i_2,j_2)\text{-}\cdots\text{-}x(i_{k-1},j_{k-1})\text{-}\alpha_k>$ (1)

式(1)の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指す．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)を，固定長ワイルドカード領域と呼ぶ．ε=j−iをワイルドカード領域x(i,j)の誤差と呼ぶ．また，あるパターンが，あるオカーレンス集合から得られるとするとき，その集合の異なる配列識別子sidの数をパターンの支持数と呼び，ユーザが与えた最小支持数以上の支持数を持つパターンを頻出パターンと呼ぶ．

3.2 問題の定義

ここで扱う問題は，配列データベースからユーザが与えた３つの入力パラメータを満たす全ての頻出パターンを抽出することである．入力パラメータには，最小支持数，最大ワイルドカード数，最大誤差数がある．最大ワイルドカード数は，頻出パターンの各ワイルドカード領域の最大の長さと定義する．最大誤差数は，配列データベースから抽出される頻出な可変長パターンの各ワイルドカード領域の中で最大の誤差数と定義する．m個の要素を持つ頻出なk-パターン集合P_kは以下のように表される．cntはそのパターンの支持数を意味する．

$P_k=\{<pat_1^k>:cnt_1, <pat_2^k>:cnt_2,\cdots,<pat_m^k>:cnt_m\}$ (2)

集合Pを，配列データベースから抽出された全ての頻出パターンと定義するとき，集合Pは$P_1\cup P_2\cup\cdots\cup P_q$と表される．ただし，qは抽出された頻出パターンの最大長と一致する．

4. 従来の方法

既存の手法は，パターンを成長させていくアプローチを用いており，k-頻出パターンから(k+1)-パターンを生成するときは常に，配列データベースの各配列をスキャンし，k-頻出パターンに追加される文字と，ワイルドカード領域を見つけ出す．無駄なスキャンを避けるため，(k+1)-頻出パターンの抽出のためのスキャン開始位置は，k-頻出パターンの最右端の文字のすぐ右隣の位置でなければならない．このスキャン開始位置を効率的に見つけ出すには，k-パターンごとに射影データベース[2]を作成しておくと便利である．k-パターン<pat^k>により定義される射影データベースPDB(<pat^k>)は以下の通りである．

PDB(<pat^k>)={(i,j)|<pat^k>のオカーレンス集合に含まれる各k-ストリングの最右端の文字は，配列データベース中の配列データs_iの(j-1)番目に存在する} (3)

表１の配列データベースに対して，可変長パターン抽出法を適用し，可変長ワイルドカード領域を持つ頻出パターンの抽出をすると図１の列挙木が得られている[4]．ただし，最小支持数,最大ワイルドカード数,最大誤差数を各々3とする．

図１中の<F-x(0,3)-K-x(1,3)-A>に注目してみよう．<F>と<K>との間の可変長ワイルドカード領域がx(0,3)となっているが，<F>と<K>との間に配置すべき極小な可変長ワイルドカード領域は，表１のs_1,s_2,s_5からわかるように，x(0,1)である．これは，射影データベースには，k-頻出パターンのk番目と(k-1)番目の文字間の可変長ワイルドカード領域だけを計算する情報しか含まれていないことが原因である．k-頻出パターンの接頭辞<pat^{k-1}>中の可変長ワイルドカード領域を極小化するためには，射影データベースの情報だけでは不十分である．また，図１中の，<F-x(0,3)-A>:3は，<F-A>,<F-x(1)-A>,<F-x(2)-A>,<F-x(3)-A>といった2-オカーレンスが存在することを表現しているが，表１に2-ストリング<F-A>を持つ2-オカーレンスが存在しない．これにより，可変長ワイルドカード領域x(0,3)は極小ではない．さらに，図１中の<F>と<K>との間の可変長ワイルドカード領域に注目すると，表１からわかるように，<F>と<K>との間のワイルドカード数は，配列データs_1では0と3，s_2では1，s_5では１である．つまり，極小の可変長ワイルドカード領域はx(0,3)である．図１中の<F-x(1,3)-K>は，<F-x(0,3)-K>に冗長な表現である．

表1:配列データベース
Table1: An example of a sequence database

sid	配列データ
1	FKYAKWLCDN
2	SFVKTAEHNQC
3	ALR
4	MSKPL
5	FSKFLMAWEH

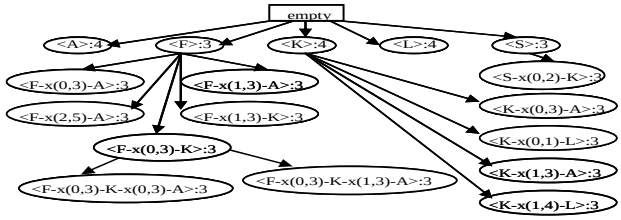

図１：抽出されるパターン(可変長パターン抽出法)
Fig1：Patterns extracted by the variable-length pattern extraction method.

5. 提案方法

射影データベースを用いたパターン成長アプローチの問題点を改善する方法として,スコープデータベースという新たな手法を提案する．(k+1)-パターンの集合が，k-パターンの成長によって生成されるとき，各(k+1)-パターンに含まれる極小かつ非冗長な可変長ワイルドカード領域のすべてと，そのパターンの支持数は，k-パターンにより定義されるスコープデータベースから計算できる．k-パターン<pat^k>により定義されるスコープデータベースは以下の通りである．

SDB(<pat^1>,[r,r+ε_{max}])={(nil,i,j,wc,$s_i[j_{new}]$)|<pat^1>のある文字が，配列データs_iの先頭からj番目に存在する．
$wc\in[r,r+\varepsilon_{max}]$, $j_{new}=j+wc$, $1\le j\le||s_i||, j_{new}\le||s_i||\}$ (4)

$SDB(<pat^k>,[r,r+\varepsilon_{max}])=\{(List^k,i,j_{next},wc_{next},s_i[j_{new}])|(List^{k-1},i,j,wc,\alpha)\in SDB_\alpha(<pat^{k-1}>,[r,r+\varepsilon_{max}]),$
$List^k=Append(List^{k-1},[wc]), wc_{next}\in[r,r+\varepsilon_{max}], s_i\in DB, j_{next}=j+wc+1, j_{new}=j_{next}+wc_{next}, 1\le j\le||s_i||, j_{next}\le||s_i||, j_{new}\le||s_i||\}$ (5)

ただし以下が成立するものとする．
$k\ge2, <pat^k>=<pat^{k-1}\text{-}x(r,r+\varepsilon)\text{-}\alpha>$, かつ$\varepsilon\le\varepsilon_{max}$.
Append(X,Y)は，リストXにリストYを追加することを指す．

$SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])=\{(List^k,i,j,wc,\alpha)|(List^k,i,j,wc,\alpha)\in SDB(<pat^k>,[r,r+\varepsilon_{max}])\}$ (6)

k≥2 のとき, $List^k$は, k-ストリング中の隣接文字間に(k-1)個配置されているワイルドカード数を表している. $List^1$が空のリストであるのは, 1-ストリングは隣り合う文字をもたないからである. 例えば, 3-ストリング<F*K***A>の$List^3$は, [1,3]と表される.

頻出なk-パターンと配列データベースを用いて構成されるスコープデータベース$SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])$には, 最右端の文字としてαを持つ(k+1)-頻出パターンの証拠としての(k+1)-オカーレンス集合が含まれている($0 \le r \le wc_{max}$). 従って, k-パターンのスコープデータベースは, 極小なk-パターンから極小な(k+1)-パターンを導き出すために有効であり, ユーザが指定した範囲内にある文字とその文字までの極小な可変長ワイルドカード領域を全て求めることができる.

5.1 スコープデータベースに対する操作

$<pat^k>$から全ての$<pat^{k+1}>$を作成するためには, $<pat^k>$によりスコープデータベースを定義し, そのスコープデータベースを用いて$<pat^{k+1}>$を構築する必要がある. 前者に対しては1つの操作, 後者に対しては3つの操作を用意している. 以下, それぞれの操作について述べる.

(操作 1) 入力パラメータとしてのワイルドカード数rが, $[0,wc_{max}]$の範囲内から順に選択されるとき, 式(4)又は(5)から, $SDB(<pat^k>,[r,r+\varepsilon_{max}])$を構築する.

(例 1) 表1に, 操作1を適用してみよう. ただし, 最小支持数, 最大ワイルドカード数, 最大誤差数は3とする. k=1のとき, 1-頻出パターンは, {<A>:4,<F>:3, <K>:4,<L>:4,<S>:3}である. 次に, r=0 とし, 接頭辞<F>から頻出な 2-パターンを生成するならば, 範囲は, [0,0+3]となる. よって, 1-パターン<F>のスコープデータベースは以下のようになる.

SDB(<F>,[0,3])=
{(nil,1,2,0,<K>),(nil,1,2,1,<Y>),(nil,1,2,2,<A>),
(nil,1,2,3,<K>),(nil,2,3,0,<V>),(nil,2,3,1,<K>),
(nil,2,3,2,<T>),(nil,2,3,3,<A>),(nil,5,2,0,<S>),
(nil,5,2,1,<K>),(nil,5,2,2,<F>),(nil,5,2,3,<L>),
(nil,5,5,0,<L>),(nil,5,5,1,<M>),(nil,5,5,2,<A>),
(nil,5,5,3,<W>)} (7)

(操作 2) 列挙木の親ノードに対応するk-頻出パターン$<pat^k>$の成長により, その子ノードに対応する全ての候補(k+1)-パターン$<pat^{k+1}>$を生成する. 区間$[0,wc_{max}]$からある値rを順に選択するとき, 最右端文字αを持つ候補(k+1)-パターン内に存在するk番目の可変長ワイルドカード領域の計算をする. このために, まず, $SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])$を用いて, $\{wc|(List,i,j,wc,\alpha)\in SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])\}$の最小値$r_{min}$と最大値$r_{max}$を計算する. もし, r_{min}とrが等しいならば, 候補(k+1)-パターン$<pat^{k+1}>=<pat^k\text{-}x(r_{min},r_{max})\text{-}\alpha>$を生成し, これが同じαを末尾に持つ(k+1)-パターンに冗長であれば, (k+1)-パターンを削除する. もし, $r_{min}>r$ならば, (k+1)-パターンを生成しない.

(例 2) r=0 とし, 例1と同じ条件下で, SDB(<F>,[0,3])にこの操作を適用してみよう. $SDB_K(<F>,[0,3])$={(nil,1,2,0,<K>),(nil,1,2,3,<K>),(nil,2,3,1,<K>),(nil,5,2,1,<K>)}から, $r=r_{min}=0$, $r_{max}=3$ を得ることができ, これより, 接頭辞<F>を持つ候補 2-パターン<F-x(0,3)-K>を得ることができる.

(操作 3) 各候補(k+1)-パターン$<pat^{k+1}>$の支持数を計算するため, $SDB(<pat^k>,[r,r+\varepsilon_{max}])$の部分集合である, $SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])$を使用する. $<pat^{k+1}>=<pat^k\text{-}x(r,r+\varepsilon)\text{-}\alpha>$の支持数は$SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])$中の識別子の集合$\{i|(List,i,j,wc,\alpha)\in SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])\}$であり, その集合の要素数を数えあげることで計算される. $<pat^{k+1}>$の支持数がユーザによって与えられた最小の支持数を満たすなら, $<pat^{k+1}>$は(k+1)-頻出パターンになる.

(例 3) 例2と同じ条件で, SDB(<F>,[0,3])にこの操作を適用してみよう. 表1から可変長ワイルドカード領域x(0,3)と文字“K”で, 候補の 2-パターン<F-x(0,3)-K>を抽出するために用いられた$SDB_K(<F>,[0,3])$の集合から, 識別子の集合{1, 2, 5}を得ることができる. この結果は, 最小支持数が 3 以下のとき, 2-頻出パターン<F-x(0,3)-K>:3 となる.

(操作 4) k≥2 のとき, 接頭辞$<pat^k>$を持つ(k+1)-頻出パターン$<pat^{k+1}>$が$<pat^k\text{-}x(r,r+\varepsilon)\text{-}\alpha>$($\varepsilon \le \varepsilon_{max}$)と表されるとき, $SDB_\alpha(<pat^k>,[r,r+\varepsilon_{max}])$の$List^k$を使用して, $<pat^{k+1}>$のk-接頭辞$<pat^k>$に存在するそれぞれの可変長ワイルドカード領域x(i,j)を再計算し, 極小な可変長ワイルドカード領域x(i', j')を求める. 再計算後の可変長ワイルドカード領域には, $j' \le j$ の関係が成立する. もし, $i' \neq i$の関係が成立する可変長ワイルドカード領域が1つでもパターン中に存在すれば, 冗長性発生防止のために, その(k+1)-頻出パターンを除去する.

(例 4) 3-頻出パターン<F-x(0,3)-K-x(1,3)-A>:3 が例1の$SDB_A(<F\text{-}x(0,3)\text{-}K>,[1,4])$={([0],1,3,1,<A>),([1],2,5,1,<A>),([1],5,4,3,<A>)}から生成されるとき, 操作 4 を適用することで 3-頻出パターンの可変長ワイルドカード領域x(0,3)を更新することができる. $SDB_A(<F\text{-}x(0,3)\text{-}K>,[1,4])$に含まれる各$List^2$の値は[0],[1],[1]であり, 文字“F”と“K”の間のワイルドカード数を表している. よって, 頻出な 3-パターン<F-x(0,3)-K-x(1,3)-A>: 3 における最初のワイルドカード領域x(0,3)はx(0,1)に更新される. この結果から, 極小な 3-頻出パターン<F-x(0,1)-K-x(1,3)-A>:3 を得ることができる.

上記の 4 つの操作を表 1 に繰り返し適用した結果, 抽出される頻出パターンは図2のようになる.

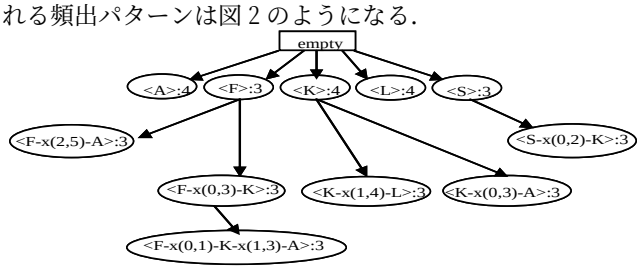

図 2：抽出されるパターン(スコープデータベース)
Fig2：Patterns extracted by the scope database.

5.2 アルゴリズムの最適化

k-頻出パターン$<pat^k>$に対するスコープデータベースは, それぞれの $r \in [0,wc_{max}]$ に対して, $wc_{max}+1$ 個の$SDB(<pat^k>,[r,r+\varepsilon_{max}])$が構築されていた. このとき, 参照される各配列データ上において, 同じ文字位置を最大で$wc_{max}+1$回もスキャンするという無駄が発生する. この無駄を低減するために, 以下の2点に着目した最適化を行っている.

（1）スコープデータベースの構造化

k-頻出パターン$<pat^k>$に対するスコープデータベースは, $r \in [0,wc_{max}]$ごとには構築せず, これらを1つにまとめ, $SDB(<pat^k>,[0,wc_{max}+\varepsilon_{max}])$を構築する. また, これをハッシュテーブルで管理し, スコープデータベースの全要素に対して, 追加候補文字αごとに分類する. 分類された要素の集合は$SDB_\alpha(<pat^k>,[0,wc_{max}+\varepsilon_{max}])$の内容と一致する.

（2）可変長ワイルドカード領域の効率的計算

どの追加候補文字αをもつグループについても, 可変長ワイ

ルドカード領域x(i,j)を計算することは容易である．この計算を効率化するために，ワイルドカード領域列挙リストと呼ばれるリストを構築する．この計算結果により，(k+1)-頻出パターン<pat^k-x(i,j)-α>が効率良く得ることができる．

6. 性能評価

ここでは,従来手法の可変長パターン抽出法と，提案手法のスコープデータベースの計算結果を比較する．このために利用した計算機環境は，Intel PIV-2.4GHz，メモリ：2GB，SWAP メモリ：2GB，HDD:74.5GB，OS：Microsoft Windows XP Professional である．性能評価のために使用したデータセットは，Leucine Zipper モチーフを含む配列データベースである．PROSITE から Leucine Zipper モチーフを含むデータセットを選択するために,登録番号として PS00036 を用いた．このデータセットには，125 件の配列データが含まれている．Leucine Zipper モチーフの形式は,<[KR]-x(1,3)-[RKSAQ]-N-x(2)-[SAQ](2)-x-[RKTAENQ]-x-R-x-[RK]>である．配列データベースから上記のモチーフを抽出するために，入力パラメータを，最大ワイルドカード数，最大誤差数をともに 2 とし，計算をした．その結果を表 2 に示す．表 2 は，最小支持率別の従来手法と提案手法の頻出パターン数と，提案手法によるワイルドカード領域の変更率を示している．ワイルドカード領域変更率とは，全ての頻出パターンのワイルドカード領域に対する，パターンの成長過程において，変更されたワイルドカード領域の割合のことである．

表 2:Leucine Zipper の計算処理
Table2: Leucine Zipper Evaluation.

比較項目/最小支持率		37%	36%	30%	25%
従来手法	頻出パターン	56821	64182	－	－
	計算時間(sec)	11.7	12.8	－	－
提案手法	頻出パターン	51739	57540	203077	700845
	計算時間(sec)	92.6	101.5	468.4	2378.0
	領域変更率(%)	4.6	5.0	11.3	11.9

提案手法は従来手法に比べ，抽出される頻出パターンが少ないことがわかる．これは，提案手法は，従来手法で抽出されていた可変長ワイルドカード領域が冗長なパターンや非極小被覆なパターンを抽出しないためである．また，他の理由として，パターン成長過程において，可変長ワイルドカード領域を極小化するための再計算を行った結果，冗長性が発生したため，パターンが削除されたことも考えられる．

また，従来手法では，支持率を 36%未満にすると，メモリ不足で計算を打ち切ってしまう．提案手法では，抽出されるパターンが少なくなることもあり，支持率 25%まで頻出パターンを抽出することに成功している．以上のことから，提案手法は，従来手法に比べ，優れた抽出能力を持っていることを確認することができた．

7. まとめ

本論文では，アミノ酸配列データベースからモチーフの候補である極小な可変長ワイルドカード領域を持つと同時に非冗長な頻出パターンを見つける方法を提案した．パターン成長アプローチに基づく射影データベースを拡張し，k-頻出パターンごとにスコープデータベースを作成している．k-頻出パターンのスコープデータベースは，従来の射影データベースに含まれるスキャン開始位置の情報に加えて，ユーザにより定められた参照範囲の情報と，それまでに求まった可変長の k-頻出パターンに対する全てのオカーレンスの情報から構成される．スコープデータベースの有効性を示すために，Leucine Zipper モチーフを含むデータセットを PROSITE から取り出し，可変長の頻出パターンを抽出する能力の評価を行った．その結果，従来手法に比べ，より極小な可変長ワイルドカード領域を持つと同時に非冗長な頻出パターンを抽出することができた．

[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：17500097），広島市立大学・特定研究費（一般研究費（コード番号:31006））の支援により行われた．

[文献]

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

[2] Jan Pei, Jiawei Han, Behzad Mortazavi-Asl, and Helen Pinto: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, Proc. of International Conference on Data Engineering (ICDE 2001), IEEE Computer Society Press, p.215-224, 2001.

[3] 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), Springer-Verlag, pp.482-491 , August 2002.

[4] 塔野 薫隆，北上 始，田村 慶一，森 康真，黒木 進：Modified PrefixSpan 法を用いた頻出正規パターンの抽出をめざして，日本データベース学会 Letters，Vol.3, No.1，pp.61-64，2004 年 6 月．

加藤 智之 Tomoyuki KATO
広島市立大学情報科学研究科博士前期課程在学中．2006 広島市立大学情報科学部卒業．日本データベース学会学生会員．

北上 始 Hajime KITAKAMI
広島市立大学情報科学部教授．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)．データベースシステムの研究・開発に従事．日本データベース学会ビジネスインテリジェンス研究グループ運営委員など．

森 康真 Yasuma MORI
広島市立大学情報科学部助手．1994 北陸先端科学技術大学院大学情報科学研究科博士前期課程修了．データベースシステムの研究・開発に従事．日本データベース学会会員．

田村 慶一 Keiichi TAMURA
広島市立大学情報科学部助手．2000 九州大学大学院システム情報科学府修士課程修了．データベース並列処理の研究に従事．日本データベース学会，情報処理学会 各会員

黒木 進 Susumu KUROKI
広島市立大学情報科学部助教授．1990 東京大学大学院工学系研究科修士課程修了．博士(工学)．空間データベースの研究に従事．日本データベース学会，情報処理学会，電子情報通信学会，ACM, IEEE 各会員．