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

加藤 智之　北上 始　森 康真　田村 慶一　黒木 進

広島市立大学情報科学部 〒731-3194 広島市安佐南区大塚東三丁目４番１号

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

あらまし 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．そのために，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．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．これにより，次に抽出される可変長の(k+1)-頻出パターンが過度に説明する表現を含むのを回避することができる．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．

キーワード 配列パターン抽出，可変長ワイルドカード領域，オカーレンス，スコープデータベース

Extraction for Frequent Sequential Patterns with Minimum Variable-Wildcard Regions

Tomoyuki KATO　Hajime KITAKAMI　Yasuma MORI　Keiichi TAMURA　Susumu KUROKI

Faculty of Information Sciences, Hiroshima City University

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

Abstract. 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 for each frequent k-length pattern is defined as the extension of Projected Database that generate frequent (k+1)-length patterns from a frequent k-length pattern in pattern growth approach. The scope database for frequent k-length pattern consists of not only the existing projected database of the pattern but also the range of scan and occurrences of the pattern. Moreover, we report experimental results that our extended method was evaluated using a dataset that includes the Leucine Zipper motif.

Key words sequential pattern extraction, variable length wildcard region, occurrence, scope database

1. はじめに

配列データベースから頻出パターンを抽出する方法は，DNA やアミノ酸などの生物学的配列データのモチーフ抽出だけでなく，顧客の購買履歴，Web のアクセス履歴などの分析に有効であるといわれている．モチーフとは，PROSITE[1][2]や Pfam[3]などで見られるアミノ酸配列上に存在する特徴的なパターンである．近年，データマイニングの立場から，アミノ酸配列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いる．そのため，アミノ酸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．

頻出パターン抽出法として，パターン成長アプローチを用いた PrefixSpan 法[4]が提案されている．しかしながら，PrefixSpan 法は，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，配列データベースに PrefixSpan 法を適用すると，余分なパターンが多く抽出され，計算時間が大変かかってしまう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[5]が提案されている．この方法は，パターン成長アプローチを用いた 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ている．また，可変長パターン抽出法[6]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．しかしながら，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，抽出された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なっていない．

(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している．可変長ワイルドカード領域を持つ頻出パターンの抽出方法は，2 ステップから成る．最初に，長さ k のパターンに対するスコープデータベースを用いて，長さ k のパターンから長さ(k+1)のパターンを求める．このとき，長さ k のパターンの最右端文字に追加すべき極小な可変長ワイルドカード領域とアルファベット文字を見つけ出す．次に，第 1 段階の処理以前に計算されている可変長ワイルドカード領域を再計算し，極小化する．頻出パターンが抽出されなくなるまでこの 2 ステップは繰り返し適用される．

本論文の構成は以下の通りである．2 章で関連研究の紹介をする．3 章で用語と問題の定義を行い，4 章で従来の手法の紹介をする．5 章で提案する手法の説明をする．6 章で性能評価を行い，7 章でまとめる．

2. 関連研究

現在までに，配列データベースからモチーフを発見する問題を解くために，多くの研究が行われてきた[7][8][9][10][11]．モチーフの候補となる頻出パターンには，固定長ワイルドカード領域や可変長ワイルドカード領域が含まれている．しかしながら，このような頻出パターンを全て抽出するには，膨大な計算量を必要とするため，ユーザが制限した部分文字列をもつ頻出パターンの抽出[7][10][11]や，特殊な長さの頻出パターンだけの抽出[9]に努力が注がれてきた．

多くの研究で抽出される頻出パターンには，固定長ワイルドカード領域だけしか含まれていない．その中で，文献[7][10]では，配列データから一定の長さの部分文字列をすべて切り出した集合（ブロックデータ構造[8]）を用いて，頻出な部分文字列を抽出している．このため，切り出された部分文字列の長さよりも，長い部分文字列を持つ頻出パターンを見落としてしまうだけではなく，処理速度が遅いという問題がある．また，文献[11]では，頻出パターンの抽出に先立ち，配列データベースからサフィックス木を構築している．このため，多大な構築時間を要するだけでなく，サフィックス木の蓄積容量が膨大になるという問題がある．

文献[9]では,スキャンフェーズと畳み込みフェーズの処理により，頻出パターンを抽出している．スキャンフェーズでは，ユーザが与えたパターン長とワイルドカード文字数をもつ基本頻出パターンをすべて抽出する．畳み込みフェーズではそれらを組み合せて最長の頻出パターンをすべて作成している．このため，抽出される頻出パターンに含まれる固定長ワイルドカード領域の長さが限定されてしまうという問題がある．

以上の問題点を解決するために，パターン成長アプローチ[4]を基本にしたパターン抽出法[5][6]が提案されている．文献[5]では，射影データベースを応用し，固定長ワイルドカード領域を含む頻出パターンが抽出されている．そこでは，長さ(k+1)の頻出パターンの抽出は，長さ k の頻出パターンに追加すべき固定長ワイルドカード領域とアルファベット文字を配列データベースから見つけ出すことにより達成される.文献[6]では,上記の追加すべき固定長ワイルドカード領域を可変長ワイルドカード領域に置き換えることだけで，可変長ワイルドカード領域をもつ頻出パターンを抽出している．このため，頻出パターンに含まれる各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，抽出された頻出パターンの集合が冗長であるという問題もある．本論文では，これらの問題を解決するために，頻出パターンの可変長ワイルドカード領域などを管理するためのスコープデータベースを導入し，それを用いた新しいパターン抽出法を提案している．

3. 用語と問題の定義

配列データベースは，DB= $\{t_1,t_2,\cdots t_n\}$ と表現される．また，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 を持つ配列と定義する．配列 s_{sid} の先頭から j 番目の文字は，$s_{sid}[j]$として表

される．表 1 は，DB={t_1,t_2,t_3,t_4,t_5},Ω={1,2,3,4,5}, t_1=<1,FKYAKWLCDN>,t_2=<2,SFVKTAEHNQC>,t_3=<3,ALR>, t_4=<4,MSKPL>,t_5=<5,FSKFLMAWEH>であることを示している．表 1 の例で言うと，アルファベット文字 A は，t_1,t_2,t_3,t_5に含まれているので，s_1[4]=s_2[6]=s_3[1]= s_5[7]=“A”と表現することができる．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．例えば，表 1 の最初の配列は 10-配列データである．

表 1:配列データベース

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

アルファベット文字及び，＊と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）を用いて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．例えば，ワイルドカードを含むストリング<F*K*A>は 3-ストリングであり，ワイルドカードを含まないストリング<FLMA>は 4-ストリングである．以下，アルファベット文字のことを単に文字と呼ぶ．

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 k-ストリングの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．k-string に，それ自身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 3 項関係{(k-string,i,j)| k-string は s_i の j 番目に存在，<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を指し，以下の形式で表現する．

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

式(1)中の，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．また，ε=j−i をワイルドカード領域 x(i,j)の誤差と呼ぶ．ある k-パターン中の全てのワイルドカード領域が固定長であるとき，つまり，誤差を含まないパターンを固定長パターンと呼ぶ．少なくとも 1 つの可変長ワイルドカードを持つパターン，つまり，誤差を含むパターンを，可変長パターンと呼ぶ．

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，ユーザが与えた最小支持数，最大ワイルドカード数，最大誤差数といった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.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最大の誤差数と定義する．頻出な k-パターンの集合 P_k が，m 個の要素を持っているなら，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．

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

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，接頭辞を射影しながらパターンを成長させていくアプローチを用いており，配列データベースから，k-頻出パターンから(k+1)-ストリングを生成するときは常に，配列データベースから，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，頻出な k-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[4]を作成しておくと便利である．k-パターン<pat^k>の射影データベース 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 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[6]. 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.

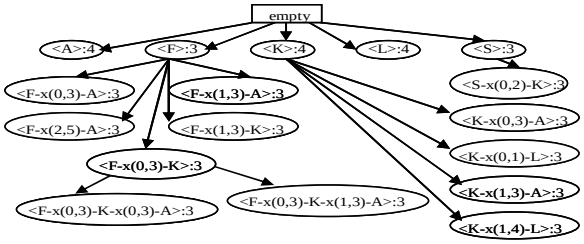

図1：抽出されるパターン(可変長パターン抽出法)

図1中の<F-x(0,3)-K-x(0,3)-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には，<F-x(0,3)-A>:3が存在するが，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.

5. 提案手法

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ための方法として,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，その(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-パターンに関するスコープデータベースによって求めることができる．k-パターン<pat^k>のスコープデータベースは以下の通り定義される.

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

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

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 $\varepsilon\le\varepsilon_{max}$.
また，Append(X,Y)は，リストXにリストYを追加することを指す.

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

$k\ge2$ 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がないからである．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便利であり，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>のスコープデータベースを構築し，そのスコープデータベースを用いて<pat^{k+1}>を作成する必要がある．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，$[0,wc_{max}]$の範囲内から選択されるとき，式(4)(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ために，SDB_α(<pat^k>,[r,r+ε_{max}])を用いて,{wc|(List,i,j,wc,α)∈SDB_α(<pat^k>,[r,r+ε_{max}])}の最小値 r_{min} と最大値 r_{max} の計算から始める．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-x(r_{min}, r_{max})-α>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 として，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_{min}=0, r_{max}=3 を得ることができる．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+ε_{max}])の部分集合である，SDB_α(<pat^k>,[r,r+ε_{max}])を使用する．<pat^{k+1}>=<pat^k-x(r,r+ε)-α>の支持数は SDB_α(<pat^k>,[r,r+ε_{max}]) 中の識別子の集合{i|(List,i,j,wc,α)∈SDB_α(<pat^k>,[r,r+ε_{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，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，支持数=3 で 2-頻出パターン<F-x(0,3)-K>:3 を抽出することができることを意味している．

(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-x(r,r+ε)-α>（$\varepsilon \le \varepsilon_{max}$）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’≤j の関係が成立する．もし，i’≠i の関係が成立する可変長ワイルドカード領域が 1 つでも存在すれば，冗長性の発生を防止するために，その(k+1)-頻出パターンを除去する．

(例 4-1) 3-頻出パターン<F-x(0,3)-K-x(1,3)-A>:3 が例 1 の SDB_A(<F-x(0,3)-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-x(0,3)-K>,[1,4])に含まれる各 $List^2$ の値[0],[1],[1]は，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．

表 2：例の配列データベース

sid	配列データ
1	AVCMUEJ
2	ALCXXKLEWRGH
3	AZDCPSPETGYYH

(例 4-2)表 2 に対して，入力パラメータを全て 2 としたとき，3-頻出パターンとして<A-x(1,2)-C-x(2,4)-E>が得られる．次にこの 3-頻出パターンから 4-頻出パターン<A-x(1,2)-C-x(2,4)-E-x(1,2)-G>が得られる．この

とき，SDB_G(<A-x(1,2)-C-x(2,4)-E>,[1,3])={([2,3],3,8,1,<G>),([1,4],3,9,2,<G>)}の，<C>と<E>とのワイルドカード数を示す $List^3$ の値が 3,4 であるので，可変長ワイルドカード領域は x(3,4)となる．しかし，2≠3 であるので，極小な 4-頻出パターンは抽出されない．

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．

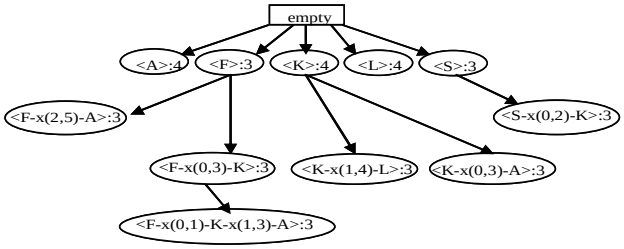

図 2：抽出されるパターン(スコープデータベース)

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∈[0,wc_{max}]に対して，wc_{max}+1 個の SDB(<pat^k>,[r,r+ε_{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∈[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}+ε_{max}])を構築する．また，これをハッシュテーブルにより管理し，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 ごとにグルーピングする．グルーピングされた要素の集合は SDB_α(<pat^k>,[0,wc_{max}+ε_{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（以後，可変長領域列挙リストと呼ぶ）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>が得られる．

以下，これらの 2 点の最適化に重要なハッシュテーブルおよび可変長領域列挙リストについて，それぞれの例を用いて説明する．

5.2.1. ハッシュテーブル

最大ワイルドカード数，最大誤差数を 3 とし，式(7)について考える．最適化をしない場合，[r,r+ε_{max}]の範囲は，[0,0+3]，[1,1+3]，[2,2+3]，[3,3+3]の 4 つの範囲に対して SDB(<pat^k>,[r,r+ε_{max}])をそれぞれ構築することになる．最適化する場合，それらをまとめた 1 つのスコープデータベースを構築することになる．すなわち，[0,0+3]∪[1,1+3]∪[2,2+3]∪[3,3+3]が[0,3+3]となるので，構築すべきスコープデータベース SDB(<F-x(0,3)-K>,[0,3+3])は以下のようになる

SDB(<F-x(0,3)-K>,[0,3+3])=
{([0],1,3,0,<Y>),([0],1,3,1,<A>),([0],1,3,2,<K>),
([0],1,3,3,<W>),([0],1,3,4,<L>),([0],1,3,5,<C>),
([0],1,3,6,<D>),([3],1,6,0,<W>),([3],1,6,1,<L>),
([3],1,6,2,<C>),([3],1,6,3,<D>),([3],1,6,4,<N>),
([1],2,5,0,<T>),([1],2,5,1,<A>),([1],2,5,2,<E>),
([1],2,5,3,<H>),([1],2,5,4,<N>),([1],2,5,5,<Q>),
([1],2,5,6,<C>),([1],5,4,0,<F>),([1],5,4,1,<L>),
([1],5,4,2,<M>),([1],5,4,3,<A>),([1],5,4,4,<W>),
([1],5,4,5,<E>),([1],5,4,6,<H>)}　　(8)

一般にスコープデータベースは大規模になる傾向があるため，このままの状態で，計算を進めるよりも，ハッシュテーブルで構造化した方が処理の効率化を達成できる．スコープデータベースの各要素が持っている追加候補文字 α をハッシュテーブルのハッシュキーとすることによって，各要素を同じ α でグルーピングすることができる．図 3 は，式(8)をハッシュテーブルで構造化した例である．α の値が<A>のエントリーに接続されている要素の集合は，以下の SDB_A(<F-x(0,3)-K>,[0,3+3])の内容に相当する．

SDB_A(<F-x(0,3)-K>,[0,3+3])=
{([0],1,3,1,<A>),([1],2,5,1,<A>),([1],5,4,3,<A>)}
(9)

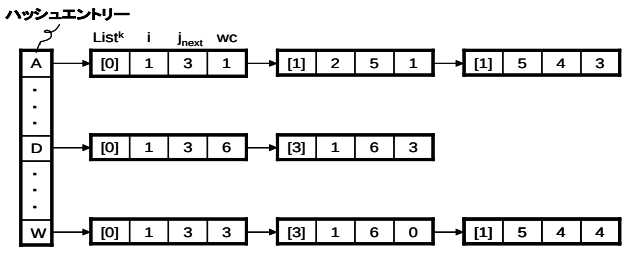

図 3：SDB(<F-x(0,3)-K >,[0,3+3])に対するハッシュテーブルの概略

5.2.2. 可変長領域列挙リスト

<pat^k>に対する SDB_α(<pat^k>,[0,$wc_{max}+\varepsilon_{max}$])を用いて，すべての<pat^{k+1}>=<pat^k-x(r,j)-α>を作成するとき,x(r,j)を極小な可変長ワイルドカード領域として SDB_α(<pat^k>,[0,$wc_{max}+\varepsilon_{max}$])から計算する方法について考える．このスコープデータベース内の各要素が持っているワイルドカード数 wc と求めるべき領域 x(r,j)の間には，j=wc=r+ε が成立する．この性質を利用すると，r とεの組み合わせ(r,ε)の集合 R（以下では r-ε集合と呼ぶ）は以下の通りである．

$$R=\{(r',\varepsilon')\mid wc=r'+\varepsilon',(List^k,i,j,wc,\alpha)\in SDB_\alpha(<pat^k>,[0,wc_{max}+\varepsilon_{max}]),\ r'\in[0,wc_{max}],\ \varepsilon'\in[0,\varepsilon_{max}]\} \quad (10)$$

この r-ε集合を r の値ごとに分類すると,最大で wc_{max}+1 個のグループ R_rが作られる.各グループから可変長ワイルドカード領域 x(r,j_{max})を計算することができる(j_{max}=r+$max_\varepsilon(R_r)$)．以下では，各グループについて，グループ内の各要素をεの値に関して昇順にならべたリストを可変長領域列挙リストと呼ぶ．

図 4 は，SDB_A(<F-x(0,3)-K>,[0,3+3])に対する 4 本の可変長領域列挙リストである．r=1 に関するグループには，$(1,0)_{\#1,\#2}$，$(1,2)_{\#5}$が接続されている．それらは，このグループに r=1 で，ε=0 をもつワイルドカード領域が配列データの識別子 1 と 2 に存在し，ε=2 をもつワイルドカード領域が配列データの識別子 5 に存在することを表している．したがって，このグループの極小な可変長ワイルドカード領域は x(1,1+2)として表現することができる．また，x(1,3)の支持数は，最小支持数を満たすので，<F-x(0,3)-K-x(1,3)-A>:3 が得られるが,x(0,3)は,前述の再計算処理によって，x(0,1)に変更される．r=0,2,3 の場合については，最小支持数を満たさないので，それらから可変長ワイルドカード領域を得ることができない．

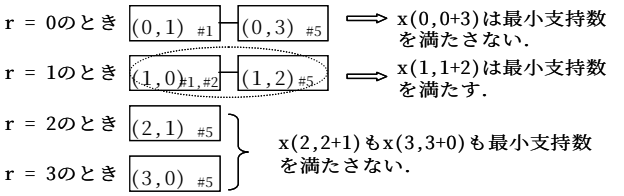

図 4: SDB_A(<F-x(0,3)-K>,[0,3+3])に対する 4 本の可変長領域列挙リスト

ここで，仮に，最小支持数を 2 としてみよう．すると，r=0 のグループも最小支持数を満たすことになる．したがって，このグループの極小な可変長ワイルドカード領域として x(0,0+3)を得ることができる．よって，頻出で極小な可変長ワイルドカード領域として，r=0 のグループの x(0,3)と r=1 のグループの x(1,3)が得られる．しかし，[1,3]⊂[0,3]であるので，x(1,3)は，x(0,3)に対して，冗長な領域である．よって冗長なワイルドカード領域 x(1,3)は除去される．

このように，可変長領域列挙リストは，冗長なワイルドカード領域を効果的に除去することができる．

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.この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 3 に示す．表 3 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，及び，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．例えば，図 2 の場合は，総ワイルドカード領域数は 7 個で，そのうち，1 箇所が変更されている(<F-x(0,1)-K-x(1,3)-A>の<F>と<K>との間のワイルドカード領域が x(0,3)から x(0,1)に変更されている)ので,表 1 のワイルドカード領域変更率は,0.14%である．

表 3:Leucine Zipper の計算処理

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，より小さい支持数でも頻出パターンを抽出することができ，かつ，極小なワイルドカード領域を持つパターンを抽出することができるので，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．

今後の課題として，ギブスサンプラーの導入や，「曖昧要素」を含む頻出パターンの抽出をする方法の研究があげられる．

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）（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] Nicolas Hulo, Christian J. A. Sigrist, Virgine Le Saux, Petra S. Langendijk-Genevaux, Lorenza Bordoli, Alexandre Gattiker, Edouard De Castro, Philipp Bucher and Amos Bairoch:Nucleic Acid Research, Vol.32, pp.134-137, 2004

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

[3] Erick L.L. Sonnhamer, Sean R. Eddy, and Richard Durbin: Pfam: A Comprehensive Database of Proteins, Vol. 28, pp.405-420, 1997

[4] 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.

[5] 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.

[6] 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.

[7] Alvis Brazma, Inge Jonassen, Ingvar Eidhammer, and David AndGilbert: Approaches to the Automatic Discovery of Patterns in Biosequences, Technical Report, Department of Informatics, University of Bergen, Norway, 1995.

[8] 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, pp28-36, 1994.

[9] Isidore Rigoutsos and Aris Floratos: Combinatorial Pattern Discovery in Biological Sequences: the TEIRESIAS Algorithm, Bioinformatics, Vol. 14 No. 1, p.55-67, 1998.

[10] Inge Jonassen, John F. Collins, and Desmond G. Higgins: Finding Flexible Patterns in Unaligned Protein Sequences, Protein Science, Cambridge University Press, pp.1587-1595, 1995.

[11] Laurent Marsan and Marie-France Sagot: Extracting Structured Motif Using a Suffix Tree Algorithms and Application to Promoter Consensus Identification, RECOMB2000, pp.210-219, ACM-Press, Tokyo in Japan, 2000.