可変長配列パターン抽出における ギブスサンプリングを用いた不要パターンの除去方式

加藤 智之† 森 康真‡ 荒木 康太郎† 黒木 進‡ 北上 始‡

†広島市立大学大学院情報科学研究科 ‡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, mori, kotaro, kuroki, kitakami}@db.its.hiroshima-cu.ac.jp

あらまし 一般に，パターン成長アプローチは，多くの頻出パターンが抽出されるが，それらの多くはモチーフの候補でない不要なパターンである．そこで，筆者らは，不要なパターンを除去するために，パターン成長アプローチとギブスサンプリングを用いた頻出配列パターン抽出法を提案する．ギブスサンプリングを用いて，各配列から長さ k の部分文字列集合を抽出し，抽出された集合に対してパターン成長アプローチを適用する．ギブスサンプリングは，配列データベース中の各配列から，できるだけ互いに類似した部分文字列の組を抽出することができる．これにより，頻出パターン抽出のために利用される配列データベースの参照範囲が限定されるので，不要なパターンを除去することができる．さらに，ギブスサンプリングにより，配列データベースを正と負の部分文字列の集合に分割し，各集合に出現する頻出パターンを考慮したパターンの絞り込みを行う．ギブスサンプリングを用いたパターン成長アプローチの有効性を確認するために，いくつかのデータセットを用いて実験を行った．その結果，抽出される頻出パターン数を 1/3 から 1/6 まで減らすことに成功した．

キーワード 配列パターン抽出，スコープデータベース，ギブスサンプリング

Elimination of Background Patterns using Gibbs Sampling on Flexible Sequence Pattern Extraction

Tomoyuki KATO† Yasuma MORI‡ Kotaro ARAKI† Susumu KUROKI‡ Hajime KITAKAMI‡

†Graduate School of Information Science, Hiroshima City University

‡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 which used pattern-growth approach and Gibbs sampling, in order to extract candidates of a motif from amino acid sequence databases. Although many frequent sequential patterns are extracted in pattern growth approach, those many are unnecessary patterns. Then, in order to remove unnecessary patterns, a Gibbs sampling is used and a set of k-subsequence is extracted from sequences, and pattern-growth approach is applied to the extracted the set of k-subsequence. A Gibbs sampling can extract the group of k-subsequence sets similar as much as possible from each array in an array database. Therefore, the extracted frequent sequential patterns are made strict and unnecessary patterns can be removed. Furthermore, we divided a sequence database into positive and negative subsequences by Gibbs sampling, and narrow down the pattern in consideration of the frequent sequential pattern which appears in each set of subsequences.

Key words sequential pattern extraction, scope database, Gibbs sampling

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ことは，アミノ酸などの生物配列データのモチーフ抽出などの多くの問題解決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]などで見られる生物学的に重要な機能をもつ特徴的なパターンである．

アミノ酸配列や，テキスト情報などを含むデータベースに対する配列データマイニングでは，固定長や可変長のワイルドカード領域をもつ頻出配列パターンを抽出する方法の研究[4][5][6][7][8][9]が進められてきた．し

かしながら,配列データマイニングのアプローチでは,数学的な規則性を漏れなく精密に抽出できるが，明らかに不要と思われる可変長配列パターンが大量に抽出されることや，支持数の低い重要なパターンの欠落などの問題点がある.

本論文では，頻出パターンが大量に抽出されるという問題点を解決することに着目している.そのために,従来の手法に Lawrence らのギブスサンプリング[10][11]の手法を新たに採り入れ，配列データベースを正の部分文字列集合と負の部分文字列集合に分割し，両者を用いて可変長頻出配列パターンを抽出することにより不要パターンを削除する方法について提案する.

本論文の構成は以下の通りである． 2 章で用語と問題の定義を行う. 3 章で従来手法の紹介をする. 4 章では不要パターンの除去法について述べる．5 章で性能評価を行い，6 章でまとめる.

2. 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，各配列は s_{sid} と表現される(n は配列数，sid は配列番号). 各 s_{sid} はアルファベット文字で構成される.

アルファベット文字と記号*で表されるワイルドカード文字(以降，ワイルドカードと呼ぶ)で構成される有限の文字列をストリングと呼ぶ．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．ワイルドカードは任意の 1 文字を表す記号である．ストリングの長さ k はストリングを構成するアルファベット文字数で決まる．例えば，<*FLMA*>は 4-ストリングであり，ワイルドカードを含むストリング<*F*K*A*>は 3-ストリングである.

2.1 パターン

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-ストリングからなる特定の集合に対する表現形式である. 例えば, 2-ストリングの集合{<*F**K*>,<*FK*>,<*F**K*>}を説明する 2-パターン<pat^2>は，<*F*-*x*(0,2)-*K*>と表現される．α_i をアルファベット の要素(1 文字)とすると，k 個のアルファベット文字をもつ k-パターン<pat^k>は以下のように表現される.

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

cnt は支持数を表し，<pat^k>が存在する，異なる配列番号 sid の数を表している．ユーザが与えた最小支持数以上の支持数をもつパターンを頻出パターンと呼ぶ.

$x(i,j)$は, ワイルドカード領域と呼ばれ, 文字 α_i と α_{i+1} の間にワイルドカードが i 個から j 個含まれていることを表している．$x(i,j)$の領域において，$i<j$ のとき，その領域を可変長ワイルドカード領域と呼ぶ．$i = j$ 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 $x(i)$で簡略表現する．また，$\varepsilon = j - i$ を可変長ワイルドカード領域の誤差と呼ぶ．ワイルドカード領域の範囲は，ユーザにより与えられた最大ワイルドカード数 wc_{max} 及び最大誤差数 ε_{max} により制限され，それぞれ $i \leqq wc_{max}$，$\varepsilon = j - i \leqq \varepsilon_{max}$ という関係が成り立つ．ある k-パターンにおいて，全てのワイルドカード領域が固定長である場合，そのパターンを固定長パターンと呼ぶ．一つでも可変長のワイルドカード領域を含むパターンを可変長パターンと呼ぶ.

一般に，各配列には同じ文字で構成される k-ストリングが数多く現れる．ある s_{sid} において，同じ文字で構成される k-ストリングの数の割合を占有度と呼ぶ. 例えば，2-ストリングの集合{<*F**K*>,<*FK*>,<*F*K*>}が同一配列に存在する場合，各 k-ストリングの占有度は 1/3 であり，{<*LA*>,<*L***A*>}が同一配列に存在する場合，各 k-ストリングの占有度は 1/2 である．各 s_{sid} において，専有度の合計は必ず 1 となる.

2.2 正パターンと負パターン

配列データベースに対してギブスサンプリングを適用すると，各配列から指定した長さ k の部分文字列が抽出される．これら k-部分文字列の集合を正の部分文字列集合と呼ぶ．また，ギブスサンプリングで抽出されなかった部分，つまり，正の部分文字列集合以外の集合を負の部分文字列集合と呼ぶ．これらの間の関係を図 1 に示す.

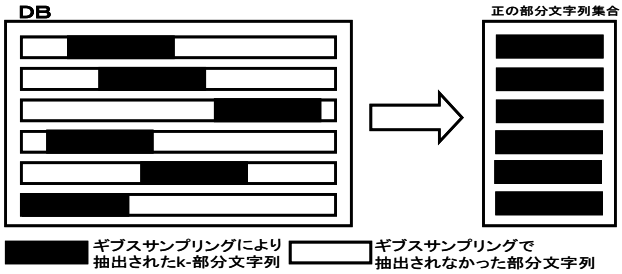

図 1:正の部分文字列集合と負の部分文字列集合

正の部分文字列集合及び負の部分文字列集合からそれぞれ抽出されるパターンを，正及び負のパターンと呼ぶ.

3. 従来手法

射影データベースを用いたパターン成長アプローチにおける可変長パターン抽出法には，頻出パターンを構成する可変長ワイルドカード領域が冗長,または,非極小被覆になる可能性があるという欠点があった．この問題を解決するために，スコープデータベース[8]

という手法が提案されている．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 *k*-ストリングの位置情報から構成される．これにより，極小かつ，非冗長な可変長ワイルドカード領域を求めることができる．

以下で，スコープデータベースによる頻出パターンの抽出手順について述べる．

(1) 入力パラメータとして，最小支持数を与える．さらに，ワイルドカード数を$[0, wc_{max}]$の範囲内から，誤差数を$[0, \varepsilon_{max}]$の範囲内からそれぞれ選択し，与える．
(2) 配列データベース *DB* を 1 回スキャンし，1-頻出パターン$<pat^1>$を全て求め，これらを F_1とする．
(3) $F_k = \phi$ になるまで，各$<pat^k> \in F_k$ 対して，以下の処理を繰り返す．
- $<pat^k>$に対してスコープデータベースを構築する．
- 構築されたスコープデータベースから極小かつ，非冗長な$(k+1)$-パターンを生成する．
- 支持数を計算し，頻出な$(k+1)$-パターンを抽出する．
- $(k+1)$-頻出パターンに含まれる全ての可変長ワイルドカード領域を極小化し，F_{k+1}に追加する．
- $k = k + 1$

(4) $F = F_1 \cup F_2 \cup \cdots \cup F_k$を出力する．

上記の処理手順を表 1 の配列データベースに適用すると，図 2 の列挙木が得られる．

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

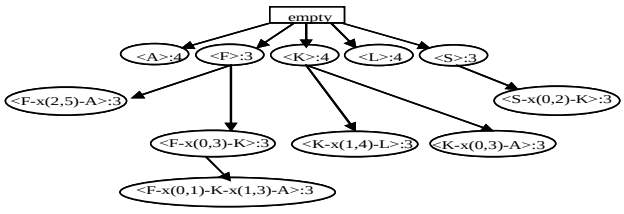

図 2：スコープデータベースにより抽出される頻出パターン

一般に，パターン成長アプローチによる頻出パターン抽出では，精密な表現の頻出パターンを抽出することができるが，一方で，大量の頻出パターンが抽出されるという問題がある．特に可変長のワイルドカード領域をもつモチーフの探索においては，膨大な量の頻出パターンが抽出され，明らかに不必要なパターンが数多く見られる．

4. 不要パターンの除去

パターン成長アプローチにより数多く抽出される不要なパターンを除去するために，統計学手法の一つであるギブスサンプリングを用いて，不要なパターンの除去を行う方法について提案する．

4.1 ギブスサンプリング

ギブスサンプリングは，配列データベース *DB* の各配列から，指定した長さ *k* をもち，お互いにできるだけ類似している部分文字列集合(*k*-部分文字列集合)を求めることができる．ギブスサンプリングのアルゴリズム[10][11]を図 3 に示す．また，ギブスサンプリングの抽出処理の例を図 4 に示す．なお，図 4 中の各番号は図 3 のアルゴリズムの各ステップ番号に対応する．

① *t* 本の配列をもつ *DB* の各配列に対して，*k*-部分文字列の開始点 st_i をランダムに選び，それらを配列順に並べた集合を $S = \{s_1, s_2, \cdots, s_t\}$とする．
② *DB*からランダムに 1 つの配列 *Z* を選択する．
③ 配列 *Z* 以外の残りの $t-1$ 本の配列から取り出される *k*-部分文字列集合から各文字の各位置 *i* での出現確率 A_i を計算する．
④ *DB* の *k*-部分文字列集合に含まれない部分から各文字 α の背景的出現確率 Q_α を計算する．
⑤ *Z* 上の各位置 *i* を *k*-部分文字列を開始点とし，それぞれについて，*k*-部分文字列の確率 P_i を出現確率と背景的出現確率を用いて計算する．
⑥ $\{P_1, P_2, \ldots, P_n\}$の各確率の中から，ランダムに P_j を選択し，P_j に対応する *k*-部分文字列の配列上の新しい開始点 st'_j を選ぶ．ただし，P_j はできるだけ値が大きいものが選ばれるものとする．
⑦ 収束するまで 2～6 を繰り返す．

図 3:ギブスサンプリングのアルゴリズム

4.2 不要パターンの除去法

以下では，不要なパターンの除去のために以下の二つの方法を提案する．

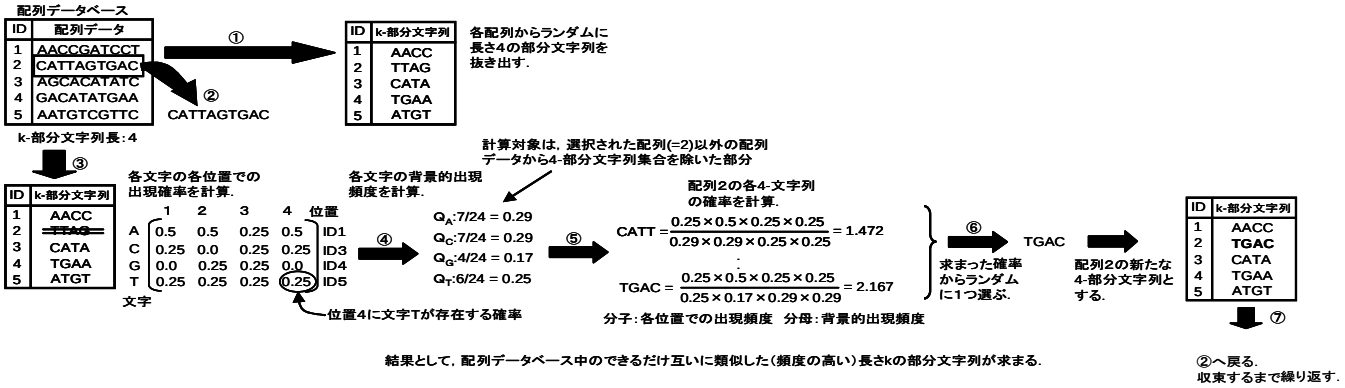

図 4:ギブスサンプリングによるパターン抽出処理の例

4.2.1 正パターン集合の利用

正の部分文字列集合を用いて，従来手法を適用する．これにより,配列データの着目箇所が限定されるので，抽出される頻出パターンの数の減少が期待される．さらに，正の部分文字列集合は，互いにできるだけ類似した配列の集合であるので，正のパターン集合に従来手法を適用することで，よりモチーフの形式に近い頻出パターンの抽出が期待できる．

4.2.2 正と負パターン集合の併用

正の部分文字列集合と負の部分文字列集合の両方の集合を用いて，頻出パターン$<pat^k>$を抽出する．負パターン集合を用いて，正の頻出パターン$<pat^k>$の支持率を負パターン$<pat^k>$の支持率で補正する．ある$<pat^k>$が正パターン集合と負パターン集合の両方に現れるとき，それぞれの支持数を $sup_P(<pat^k>)$，$sup_N(<pat^k>)$とする．このとき，負のパターンにより補正した支持数を $sup(<pat^k>)$とする．

$<pat^k>$が固定長パターンのとき，$<pat^k>$は唯一の k-ストリングを表現しているので，負のパターンにより補正した支持数 $sup(<pat^k>)$の計算法は容易である．しかし，$<pat^k>$が複数の k-ストリングを表現する可変長パターンであるとき，同じ ID を持つ配列上に異なる k-ストリングが複数存在する可能性があるので，これを考慮して $sup(<pat^k>)$を計算しなければならない．従って，少し複雑な計算になる．以下に，$<pat^k>$が固定長パターンのとき及び可変長パターンのときのそれぞれについて，$sup(<pat^k>)$の計算法について述べる．

（１） 固定長パターンの支持数計算法

n を配列数とすると，$(1-sup_N(<pat^k>)/n)$は，$<pat^k>$が負の部分文字列に存在しない割合である.ここでは，この割合を正パターンの支持数 $sup_p(<pat^k>)$に掛けることによって,負の部分文字列の情報を反映し,$<pat^k>$の支持数を補正する．すなわち，支持数 $sup(<pat^k>)$を以下のように計算する.

$$sup(<pat^k>) = sup_p(<pat^k>)\times(1-sup_N(<pat^k>)/n) \qquad (2)$$

負パターンの支持数 $sup_N(<pat^k>)$が大きくなれば，$<pat^k>$の支持数 $sup(<pat^k>)$の支持数が下がる．従って補正により支持数が下がり，最小支持数を満たさなくなれば，$<pat^k>$は頻出パターン集合から除去される．短い正の頻出パターンは負の部分文字列集合に多く含まれるので，補正により求めるべき頻出パターン集合から除去される可能性が高い.

（２） 可変長パターンの支持数計算法

k-ストリング集合 $STS = \{k\text{-}string_1, k\text{-}string_2, \ldots, k\text{-}string_m\}$を表現する可変長パターンを$<pat^k>$とするとき，各配列 i 上に出現する k-ストリング集合 $STS_i = \{k\text{-}string_{i1}, k\text{-}string_{i2}, \ldots, k\text{-}string_{im}\}$の各要素の占有度 $ocp(k\text{-}string_{ij}) = 1/|STS_i|$を計算し,負の部分文字列の情報を各占有度に反映し補正したあと，補正された占有度の合計を支持数 $sup_Q(<pat^k>)$とする．すなわち，支持数 $sup_Q(<pat^k>)$を以下のように計算する.

$$sup(<pat^k>) = \Sigma\, ocp(k\text{-}string_{ij})\times(1-sup_N(k\text{-}string_{ij})/n) \quad [1\leqq i\leqq n]\ [1\leqq j\leqq m_i] \qquad (3)$$

ただし，m_i は配列 i に存在する k-ストリングの数とする.

特に,$STS=\{k\text{-}string\}$の要素数を１とすると,$k\text{-}string$が存在する配列では $ocp(k\text{-}string)=1$ であり,存在しなければ $ocp(k\text{-}string)=0$ であるので，式(3)は，式(2)と一致する.

表 2 を用いて，式(3)の計算例を以下に示す．$STS = \{<AC>, <A*C>, <A**C>\}$ を表現する可変長パターン$<A\text{-}x(0,2)\text{-}C>$の支持数を計算する方法について考えてみよう．各 2-ストリングの支持数は，4, 1, 2 である.

ID が 1 の配列に存在する 2-ストリングは<*AC*>と<*A**C*>だけである．これにより，各 2-ストリングの占有度は 1/2 となる．*ID* が 2 の配列に関しても 2-ストリングは<*AC*>と<*A**C*>だけであり，それぞれ同じ占有度 1/2 をもつ．*ID* が 3 の配列に存在する 2-ストリングは<*AC*>だけであるので，占有度は 1 である．*ID* が 4 の配列についても 2-ストリングは<*AC*>だけであり，同じ占有度 1 をもつ．*ID* が 5 の配列については，<*A*C*>だけが存在するので，占有度は 1 である．以上から，負の文字列集合が存在しないとすれば，<*A*-*x*(0,2)-*C*>の支持数は，1/2+1/2+1/2+1/2+1+1+1=5 となる．これにより，<*A*-*x*(0,2)-*C*>:5 が得られる．また，各ストリングの支持数はそれぞれ{<*AC*>:3，<*A*C*>:1，<*A**C*>:1}となる．もし，負の部分文字列集合に<*A**C*>:5 が存在すれば，補正が必要であり，<*A*-*x*(0,2)-*C*>の支持数は，3(1-0/5)+1(1-0/5)+1(1-5/5)=4 となる．これにより，<*A*-*x*(0,1)-*C*>:4 が得られる．

以下で処理手順について述べる．

(1) ギブスサンプリングを用いて，配列データベースを正負の部分文字列集合に分割する．
(2) $F_k = \phi$ になるまで，各<pat^k>$\in F_k$ 対して，以下の処理を繰り返す．
- <pat^k>に対してスコープデータベースを適用する．
- 構築されたスコープデータベースから，(*k*+1)-ストリングを生成する．
- 生成された各(*k*+1)-ストリングを出現位置により，それぞれ正または負に分類し，占有度から支持数を計算する．
- 式(3)を用いて支持数の補正を行う．
- 頻出な(*k*+1)-パターンを抽出する
- (*k*+1)-頻出パターン中の全ての可変長ワイルドカード領域の再計算を行う．
- $k = k + 1$

(3) $F = F_1 \cup F_2 \cup \cdots \cup F_k$ を出力する．

表 2:配列データベース

ID	sequence
1	ACFAGHC
2	FACGCK
3	RKACK
4	SACLM
5	RAKCFF

5．評価

ここでは，従来手法と提案手法よる頻出パターン抽出の計算結果を比較する．性能評価のために使用したデータセットは，Leucine Zipper モチーフ及び Zinc Finger モチーフを含む配列データベースである．

5.1 正の部分文字列集合を用いた評価

ここでは，ギブスサンプリングにより抽出された正の部分文字列集合に対して，従来手法を適用した結果について述べる．

5.1.1 Leucine Zipper データセット

ここでは，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おり，47250 文字で構成されている．また，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 とした．表 3 は，従来手法による頻出パターン抽出結果を，表 4 は提案手法による頻出パターン抽出結果を表している．ただし，表 4 中の *k* は，ギブスサンプリングで抽出する正の部分文字列集合の文字列の長さ，各表のモチーフ数は，抽出された頻出パターン集合中に含まれるモチーフの数を表している．また，最小支持率は，抽出されるワイルドカード領域を固定長ワイルドカード領域に限定した際にモチーフが抽出される直前付近の支持率から設定した．

表 3:Leucine Zipper の計算処理(従来手法)

比較項目/最小支持数	37%	36%	30%	25%
頻出パターン数(件)	52129	58029	205808	711663
モチーフ数(件)	6	6	11	11
計算時間(秒)	57.46	63.99	341.69	2141.13

表 4:Leucine Zipper の計算処理(提案手法)

比較項目/最小支持数		37%	36%	30%	25%
k = 16	頻出パターン数(件)	1539	1726	3572	5274
	モチーフ数(件)	0	0	0	0
	計算時間(秒)	0.43	0.45	0.72	1.03
k = 32	頻出パターン数(件)	14034	16280	93638	243616
	モチーフ数(件)	6	6	11	11
	計算時間(秒)	16.16	19.31	169.81	466.55
k = 64	頻出パターン数(件)	18878	22037	129736	566330
	モチーフ数(件)	6	6	11	11
	計算時間(秒)	25.07	28.93	262.15	1628.88
k = 128	頻出パターン数(件)	21611	24924	137137	579804
	モチーフ数(件)	6	6	11	11
	計算時間(秒)	26.10	30.53	252.12	1666.38

Leucine Zipper モチーフは，ワイルドカード領域を含めると最大で 16 文字であるため，*k* の値を 16 の倍

数に設定し，実験を行った．

kの値を 16 としたところ，抽出された頻出パターン中にモチーフを見つけることはできなかった．次に k を 32 としたところ，抽出される頻出パターン数が約 1/3 に減少しているにもかかわらず，各支持率において，提案手法と同じ数のモチーフを得ることができた．さらに，kの値を増加させても同様の結果が得られた．

5.1.2 Zinc Finger データセット

ここでは，PROSITE から Zinc Finger 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 PS00028 を用いた．このデータセットには，744 件の配列データが含まれており，426011 文字で構成されている．また，Zinc Finger モチーフの形式は，<*C*-*x*(2,4)-*C*-*x*(3)-[*LIVMFYWC*]-*x*(8)-*H*-*x*(3,5)-*H*>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を 8，最大誤差数を 2 とした．表 5 に従来手法による頻出パターン抽出結果を，表 6 に提案手法による頻出パターン抽出結果を示す．最小支持率は，Leucine Zipper モチーフと同様に，抽出されるワイルドカード領域を固定長ワイルドカード領域に限定した際にモチーフが抽出される直前付近の支持率から設定した．

表 5:Zinc Finger の計算処理(従来手法)

比較項目/最小支持数	90%	80%	70%
頻出パターン数(件)	1042	9673	293218
モチーフ数(件)	9	9	9
計算時間(秒)	115.47	744.41	11572.22

表 6:Zinc Finger の計算処理(提案手法)

比較項目/最小支持数		90%	80%	70%
k = 25	頻出パターン数(件)	44	242	935
	モチーフ数(件)	0	9	9
	計算時間(秒)	0.17	0.87	2.39
k = 50	頻出パターン数(件)	65	463	3633
	モチーフ数(件)	0	9	9
	計算時間(秒)	0.92	5.11	31.62
k = 100	頻出パターン数(件)	168	2143	47566
	モチーフ数(件)	9	9	9
	計算時間(秒)	5.29	55.65	985.15
k = 150	頻出パターン数(件)	233	2559	77727
	モチーフ数(件)	9	9	9
	計算時間(秒)	10.20	93.17	1893.55

Zinc Finger モチーフは，ワイルドカード領域を考慮すると最大で 25 文字であるため，kを 25 及び 50 として実験を行ったところ，どちらの場合も支持率が 90%のときは抽出された頻出パターン中にモチーフを見つけ出すことはできなかったが，支持率 80%以下では提案手法と同じ数のモチーフを見つけることができた．さらに，kを 100 としたところ，支持率 90%でも従来手法と同じ数のモチーフを見つけることができた．このとき，抽出される頻出パターン数は約 1/6 に減少している．

5.2 考察と今後の課題

前述の 2 つのデータセットの実験結果について考察する．両データセットともに，ギブスサンプリングを適用しない場合，頻出パターンの探索範囲はデータセット全体であるため，抽出される頻出パターン数が多くなっている．一方，ギブスサンプリングを適用した場合，データセットの着目範囲が限定される．例えば，Leucine Zipper モチーフを含むデータセットで，k の値が 16 のとき，着目範囲の文字数は 16×125 で，2000 文字となり，ギブスサンプリングを適用しない場合に対して，探索範囲が約 4%に減少しているため，抽出される頻出パターン数が減少している．それにも関わらず，ギブスサンプリングを適用しない場合と同数のモチーフを発見できている．

以上のことから，従来手法にギブスサンプリングを取り入れることで，モチーフの数を減少することなく，抽出される頻出パターン数を減少させることができるため，優れた抽出能力をもっているといえる．

今後の研究課題として，4.2.2 章で示した負パターン集合用いた頻出パターン削減法の有効性を示すことがある．そのためには，配列データベース中のパターンの存在位置の解析が重要である．例えば，１配列中に重要なパターンが繰り返し存在する場合，それらのパターンが正負の両方のパターン集合に存在する場合，負パターン集合を用いて補正を行うと支持数が下がり，頻出パターンとして抽出されない可能性がある．この問題を解決するためには，1 配列中に繰り返し現れるパターンを考慮したパターン抽出法やギブスサンプリングの導入が必要である．

6．まとめ

本論文では，ギブスサンプリングを用いてパターン成長アプローチによって抽出される不要なパターンの除去を行った．Leucine Zipper モチーフ及び Zinc Finger モチーフを含むデータセットを PROSITE から取り出し実験を行った．ギブスサンプリングで抽出された長さ k の部分文字列集合に対して，スコープデータベースを適用した結果，頻出パターン中に含まれるモチーフ数を減少することなく，抽出される頻出パターン数を，Leucine Zipper データセットで 1/3 に，Zinc

Finger データセットで 1/6 まで減少することに成功した．

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：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] 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.

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

[6] 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.

[7] 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.

[8] 加藤 智之，北上 始，森 康真，田村 慶一，黒木 進：極小かつ非冗長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，電子情報通信学会論文誌 D，データ工学特集号，Vol.J90-D，No.2，2007 年 2 月出版

[9] Hiroki Arimura, Takeaki Uno: A Polynomial Space and Polynomial Delay Algorithm for Enumeration of Maximal Motifs in a Sequence, Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005), Vol.3827, pp.724-737, 2005.

[10] Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.N. and Wotton,J.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, Science,263,208-214, 1993.

[11] Liu, J.S., Neuwald, A.N. and Lawrence,C.E.:Bayesian models for multiple local sequence alignment and Gibbs sampling strategies, JASA, 90, 1156-1170, 1995.