異なる尺度を持つ類似文字列検索が最小汎化集合抽出に与える影響

森近　昭紀　北上　始　森　康真
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
E-mail: mw67033@edu.ipc.hiroshima-cu.ac.jp, {kitakami,mori}@hiroshima-cu.ac.jp

あらまし
配列データベースに対して類似文字列検索を行うと，類似する文字列の集合，すなわち，ミスマッチクラスタが得られるが，ミスマッチクラスタから抽出される規則性の精度等は，類似文字列検索で利用される尺度（文字列間の距離）の定義の違いにより左右される．本研究では，この尺度として，編集距離とハミング距離の 2 種類をとりあげ，最小汎化集合（正規表現）を規則性として抽出し，その精度等を評価する．具体的には，それぞれの尺度をもつ類似文字列検索手法に対する最小汎化集合を抽出し，支持数に関して上位にランキングされた汎化配列パターンの精度等を評価し，2 種類の尺度の違いによる影響を比較・考察する．

キーワード
データマイニング，配列データベース，バイオインフォマティクス，正規表現，ランキング

1. はじめに

類似文字列検索は，テキストデータベースや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，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はじめとし，クラスタリングや配列データマイニングなどの多くの分野で重要な要素技術である．

類似文字列検索を行うことで，検索結果として，非常に多くの類似した部分文字列を得ることができる．この得られた部分文字列の集合をミスマッチクラスタと呼ぶ．このミスマッチクラスタをユーザーが目で直接見て，規則的な特徴を見つけ出し把握することは，膨大なコストがかかってしまい，極めて困難なことである．そこで，ミスマッチクラスタから最小汎化集合と呼ばれる，極大な汎化配列パターンの集合と汎化できなかった部分文字列の集合とを抽出する研究が行われている．最小汎化集合を抽出できれば，ユーザーは，(1)曖昧な問合せ処理の結果として得られたすべての部分文字列を閲覧，分析の手間から解放され，(2)部分文字列の規則的な変化の様子を理解できる．例えば，アミノ酸配列データを用いた実験において，ミスマッチクラスタから最小汎化集合を抽出することで，汎化配列パターンを支持数でランキングすると，生物学的な機能をもつモチーフと呼ばれる配列パターンが，上位にランキングされる傾向があることが確認されている．

このように，ミスマッチクラスタから最小汎化集合を抽出することは，規則的な特徴を把握することに有効である．また，ミスマッチクラスタを構成する部分文字列の文字列長が等しくない場合でも，研究[1]により，異なる文字列長の部分文字列で構成されたミスマッチクラスタに対してアラインメント処理を行い，アラインメント結果から最小汎化集合を抽出することが可能となった．

そこで，本研究では，類似性検索で利用する非類似度の尺度として，編集距離とハミング距離の 2 つをとりあげ，それぞれが最小汎化集合の計算結果に与える精度面の影響について考察する．なお，精度を検証するために，*ZincFinger* データセットと *Kringle* データセットの 2 種類のデータセットを利用している．

本論文の構成は以下の通りである．第 2 章では関連研究について述べる．第 3 章で本論文に扱う記号の定義を示し，第 4 章では，検索手法について説明する．第 5 章で評価実験について説明する．第 6 章で本論文のまとめを行う．

2. 関連研究

ミスマッチクラスタから最小汎化集合を効率よく抽出する方法がいくつか研究されており，反復精密化法，ドメイン分割法と反復精密化法の併用方法，段階的一般化法が提案されている．反復精密化法は，ミスマッチクラスタから最汎パターン

と呼ばれるミスマッチクラスタを表現する最も一般的な汎化配列パターンの起点を探索木のルートとし，パターン切除に基づき，ルートから下方向に探索を進める．段階的一般化法は，ミスマッチクラスタを構成する部分文字列の部分集合に対する汎化配列パターンを段階的に列挙することで探索を進める．反復精密化法や段階的一般化法を用いることで，ミスマッチクラスタから最小汎化集合を求めることができるが，これらはミスマッチクラスタを構成するすべての部分文字列の長さが等しいことを前提に作成されている．

ミスマッチクラスタを構成する部分文字列の長さがそれぞれ異なる場合では，アラインメント処理に基づくミスマッチクラスタからの最小汎化集合の抽出[1]として提案されているものがある．これはミスマッチクラスタに対して，階層的に部分文字列をクラスタリングしながら，アラインメントを行っていく累進法を用いてアラインメント処理を行い，ミスマッチクラスタを構成するインスタンスを列ごとに比較し，列についてまとまりのある部分文字列を$(*n*)と表記する記号に変換する，この変換を用いて，汎化処理を行い，最小汎化集合を抽出する．

本論文では，このアラインメント処理に基づくミスマッチクラスタからの最小汎化集合の抽出において，ミスマッチクラスタを計算する曖昧検索の段階で編集距離を用いると，抽出精度にどの程度向上するのかについて検証している．

3. 記号の定義

本章では，本論文で使用する記号と編集距離の定義を行う．

3.1 編集距離

ある 2 つの文字列 X と Y があるとき，X に対して文字の削除, 挿入, 置換などをすることにより，X を Y に変換する操作は，編集操作と呼ばれる．

編集距離とは，この 2 つの文字列どちらかに空白文字を挿入（この挿入が Y 側ならば X 側の対応する文字は削除を意味する）して長さを同じにそろえ，編集操作の最小数を指す．また，文字列 X から文字列 Y に変換する編集距離を $d(X, Y)$と表す．

また，この $d(X, Y)$の値を出すために編集グラフというものが用いられる．文字列 X の長さを n，文字列 Y の長さを m とする．また，編集グラフの i 列 j 行目を $D(i, j)$と表す．編集グラフの作成にあたり，まず i 列に 0 から n，j 行に 0 から m までの数字を入れる．その後以下の式に従い，値を入れていく．

$$D(i,\ j) = min\begin{Bmatrix} D(i,\ j-1) \\ D(i-1,\ j) \\ D(i-1,\ j-1) + t(X_i,\ Y_j) \end{Bmatrix}$$

$$for\ \ t(i,\ j) = \begin{Bmatrix} 1 & (X_i \neq Y_j) \\ 0 & (X_i = Y_j) \end{Bmatrix} \cdots (1)$$

この式により検出された $D(n, m)$の値が編集距離の値である．

また，トレースバックというものを行い，空白文字の入るところも確認できる．トレースバックの方法として，作成した編集グラフの $D(n, m)$から $D(0, 0)$に向かって，左側，上側，左斜め上側のどの値の最小値をとってきたかをたどっていくことにより，トレースバック経路が生成される．もし $D(i, j)$でトレースバック経路を左側に行くと，その時に文字列 X に空白文字が入り，上側に行くと，文字列 Y に空白文字が入る．またトレースバック経路は複数ある時もある．

また，トレースバックを行う際，検索文字の中にあるワイルドカード以外の文字が空白文字に置き換わり，うまくクラスタリングができない．このため，本論文では検索文字側には文字の削除は許さないとし，式(1)を以下のように変更して行う．

$$D(i,\ j) = min\begin{Bmatrix} D(i,\ j-1) + 100 \\ D(i-1,\ j) \\ D(i-1,\ j-1) + t(X_i,\ Y_j) \end{Bmatrix}$$

$$for\ \ t(i,\ j) = \begin{Bmatrix} 1 & (X_i \neq Y_j) \\ 0 & (X_i = Y_j) \end{Bmatrix}$$

本研究ではこの編集距離を用いてデータベースに対し，類似性検索を行うと最小汎化集合にどのような影響があるのかを研究することがゴールである．

3.2 ミスマッチクラスタ

ミスマッチクラスタ MIS を本論文では，以下の形式で表記する．

$$MIS = \{< inst_1 >, < inst_2 >, \ \ldots \ \ , < inst_k >\}$$

また，編集距離による類似性検索により，データベースから得られた部分文字列$< inst_k >$をインスタンスと呼ぶ．各インスタンス$< inst_k >$の長さは$|< inst_k >|$と表し，インスタンス$< inst_k >$の第 i 文字目を$< inst_k >[i\ ]$と表記する．

3.3 汎化配列パターン

記号Σ^iを任意の1文字Σの部分集合とするとき，k 個の Σ^i を並べたパターンを k-汎化配列パターンと呼び，次の式のように$< pat^k >$と表記する．

$$< pat^k >= < \Sigma^1\Sigma^2 \ldots \Sigma^{k-1}\Sigma^k >$$

だだし，Σ_iはたびたび括弧[ ]の中に Σ の全要素を列挙した表記する．$\Sigma_i \subseteq \Sigma$ が存在する場所を曖昧文字領域と呼ぶ．また，$|\Sigma_i| \leqq 2$ のとき，集合 Σ_i は曖昧文字ドメインと呼ばれ，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．曖昧文字ドメインが 1 個所以上存在するとき，$< pat^k >$を k-汎化配列パターンと呼ぶ．

3.4 最小汎化集合

最小汎化集合 $MGS = \{G_1, G_2, \ldots, G_m\}$ ($1 \leqq m \leqq |MIS|$)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する．ただし，$EVAL(< pat^k >) \subseteq MIS$ かつ，$< inst^k > \in MIS$ を満たすものとする．また，関数 $EVAL$ は汎化配列パターンに含まれるすべてのインスタンスを求める関数とする．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と呼ぶ．

(1) $EVAL(MGS) = MIS$
(2) MGS の任意の 2 つの要素 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．
(4) 上記の(1) ~ (3)を満たす任意の MGS' に対し，$|MGS'| \leqq |MGS|$ が成立する．

一般に，MGS が上記の(1) ~ (3)を満たすだけでは MGS を一意に定めることができないので，これらに上記(4)を追加し，最小汎化集合が一意に定まるようにしている．

4. 検索手法

本章では，検索手法について説明する．4.1 節では．全体の処理手順について説明を行う．4.2 節では，既存の手法のハミング距離を用いた類似性検索について述べ，4.3 節で編集距離による類似性検索について示す．

4.1 処理手順

図 1 を用いて，提案手法の処理手順を説明する．最初に，テキストデータベースに対して編集距離を用いた類似性検索を行い，ミスマッチクラスタ MIS を得る．得られたミスマッチクラスタ MIS に対して，累進法[1]を用いたアラインメント処理を行い，文字列長の等しいミスマッチクラスタ MIS'を得る．次に，ミスマッチクラスタ MIS'に対して$変換を行い，まとまりのある部分文字列を$(n)と表記する．また，変換元の文字列と変換後の$(n)記号の対応表 R を作成する．最後に，ミスマッチクラスタ MIS'に対して$変換したものに，汎化処理にかけ，最小汎化集合を抽出するという流れである．

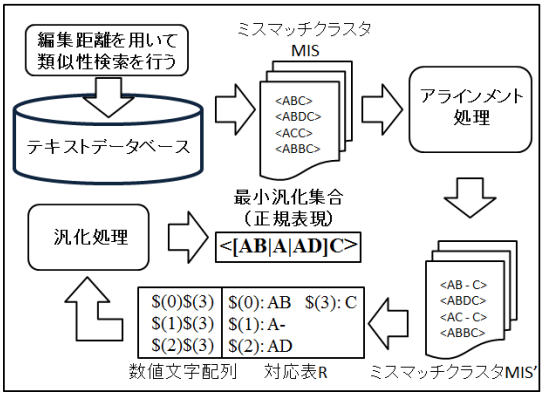

図 1：提案手法による処理手順

処理手順の詳細は，以下のとおりである．

(1) 配列データベース D に対して，キーワード K，誤差数 ε をそれぞれ入力し，編集距離を用いて類似性検索を行う．類似性検索で得られた結果をミスマッチクラスタ MIS とする．編集距離を用いた類似性検索についての詳細は，4. 3 節で説明する．
(2) ミスマッチクラスタ MIS に対して，累進法[5]を用いたアラインメント処理を行い，ギャップ記号”－“を入れることでミスマッチクラスタの各インスタンス$< inst_k >$ $(1 \leqq k \leqq n)$ の長さを同じにする．アラインメント処理を行った後のミスマッチクラスタを MIS' とする．
(3) ミスマッチクラスタ MIS' に対して$変換を行う．$変換関数を $Dollar(X)$(X はミスマッチクラスタ)とすると $MIS'' = Dollar(MIS')$ となる．$変換では，ミスマッチクラスタを構成するインスタンスを列ごとに比較し，列について，まとまりのある部分文字列を$(n)と表記する記号に変換する．また，変換元の文字列と変換後の$(n)記号の対応表 R を作成する．
(4) $変換によって各インスタンスが$(n)の記号列に変換されたミスマッチクラスタ MIS'' に対

して汎化処理を行い，最小汎化集合 *MGS* を抽出する．

(5) 対応表 *R* を使用して，$(*n*)の記号列を文字列に戻す．

4.2 ハミング距離による類似性検索

テキストデータベースにおいて既存の手法として，ハミング距離を用いた類似性検索がある．部分文字列 *K* と許容誤差数 ε (≧0) (本実験では，ε = 1とする) が, 問合せ*Q*として与えられているとき，この問合せ *Q* が，ハミング距離 *d* (*K, K'*) ≦ε を満たす部分文字列 *K'* のすべてを配列データベース *DB* から選択するとき, この問合せ *Q* による検索をハミング距離による類似性検索と呼ぶ.

4.3 編集距離による類似性検索

編集距離による類似性検索では編集グラフを用いて，類似性検索を行っていく．問合せ *Q* が，編集距離 *d* (*K, K'*) ≦ε (本実験では, *ε* = 1 とする)を満たす部分文字列 *K'* のすべてを配列データベース *DB* から選択するとき，この問合せ *Q* を，編集距離を用いた類似性検索と呼ぶ.

5. 評価実験

提案手法の評価を行うために，提案手法である編集距離を用いた類似性検索と，既存の手法であるハミング距離を用いた類似性検索の 2 種類の手法を *ZincFinfer* と *Kringle* の 2 種類のアミノ酸配列データベースに対してそれぞれ行い，得られたミスマッチクラスタから最小汎化集合を抽出することを行った．そして，抽出された最小汎化集合の汎化配列パターンの支持数が多い順にランキングを行い，それぞれの汎化配列パターンにどの程度の意味のあるパターンが含まれているか，どれくらいの精度で汎化配列パターンを抽出できているかを比較することによって，編集距離を用いた類似性検索とハミング距離を用いた類似性検索による影響を考察する．

5.1 データセット

表 1 に評価実験に用いたデータセットとその詳細を示す．用いたデータセットは，PROSITE[6]から取得した *ZincFinger*（登録番号:PS00028）と *Kringle*（登録番号:PS00021) の特徴を示す.

表 1 データセット詳細

データセット名	登録番号	データ件数	総長 (bytes)
ZincFinger	PS00028	1839	1146506
Kringle	PS00021	90	59123

実験では，最初に *ZincFinger* データセットに対して，編集距離による類似性検索を用いて曖昧な問合せを行う．*ZincFinger* のモチーフ配列パターンは，
<*C*-*x*(*2,4*)-*C*-*x*(*3*)-[*LIVNFYWC*]-*x*(*8*)-*H*-*x*(*3,5*)-*H*> として知られている．*ZincFinger* の検索文字は，<*C*-*x*(*2,4*)-*C*-*x*(*3*)-*L*-*x*(*8*)-*H*-*x*(*3,5*)-*H*>（可変長ワイルドカード）とし，許容誤差数は 1 で行う．次に，ハミング距離による類似性検索を用いて曖昧な問合せを行う．ハミング距離を用いる場合は，アラインメントを行った場合と行わなかった場合で行う．

次に，*Kringle* データセットに対する類似性検索である．*Kringle* のモチーフ配列パターン <[*FY*]-*C*-[*RH*]-[*NS*]-*x*(*7,8*)-[*WY*]-*C*>として知られているが，モチーフパターンの中で *Kringle* データセット中に出てきていないパターンがあるかを調べて，*Kringle* データセット中に出現しないモチーフパターンは，予め削除した．よって，モチーフ配列パターンは，
<[*FY*]-*C*-*R*-*N*-*x*(7,8)-*W*-*C*>,
<*F*-*C*-*R*-*S*-*x*(7)-*W*-*C*>,<*Y*-*C*-*R*-*N*-*x*(7)-*Y*-*C*>となった．また，検索文字列は，<*F*-*C*-*R*-*N*-*x*(*8*)-*W*-*C*>とし，許容誤差数 ε = 1 で行う．実験の手順は，*ZincFinger* と同様である．

また，表 3 ~ 表 8 に実行結果を示す.

本論文では，配列パターンの表記は，PROSITE 表記[6]に従って表記することとする．*x*(3)は，3 文字のワイルドカード領域(ワイルドカードとは，どんな文字でもよいことを示す)を示し，*x*(2,4)は，2 文字~4 文字のワイルドカード領域であることを示す.

5.2 実験結果

ZincFinger データセットと *Kringle* データセット対して，非類似度の尺度として，編集距離とハミング距離の 2 種類の類似性検索をそれぞれ行った．また，ハミング距離を用いた類似性検索では，累進法を用いたアラインメント処理を行う場合と行わない場合の 2 種類の場合で実験したが，編集距離を用いた類似性検索では，検索されたミスマッチクラスタは，長さが異なる要素が含まれるため，必ずアラインメント処理を行うこととした．この結果，返された合計 6 種類のミスマッチクラスタについて，ミスマッチクラスタごとに部分文字列の文字別ごとに検索を行った．汎化用ミスマッチクラスタとして，ミスマッチクラスタ *MIS* の構成条件と用いたデータセットによって次のように定義する．まず，データセットに *ZincFinger* を用い

た時，アラインメント処理を適用し，編集距離を用いて類似性検索を行った結果得られたミスマッチクラスタを *MIS*1 とし，アラインメント処理を適用し，ハミング距離を用いて類似性検索を行った結果得られたミスマッチクラスタを *MIS*2 とし，アラインメント処理を適用せず，ハミング距離を用いて類似性検索を行った結果得られたミスマッチクラスタを *MIS*3 とする．次に，データセットに *Kringle* を用いた時, アラインメント処理を適用し，編集距離を用いて類似性検索を行った結果得られたミスマッチクラスタを *MIS*4 とし，アラインメント処理を適用し，ハミング距離を用いて類似性検索を行った結果得られたミスマッチクラスタを *MIS*5 とし，アラインメント処理を適用せず，ハミング距離を用いて類似性検索を行った結果得られたミスマッチクラスタを *MIS*6 とする．これを表にまとめると以下のようになる．

表 2：汎化用ミスマッチクラスタの種類

MIS の構築条件 / データセット名	アラインメント処理有り		アラインメント処理無し
	編集距離	ハミング距離	ハミング距離
ZincFinger	*MIS*1	*MIS*2	*MIS*3
Kringle	*MIS*4	*MIS*5	*MIS*6

そして，抽出された最小汎化集合の汎化配列パターンを何個のインスタンスから構成されているかでランキングづけを行った．表 3 は，*MIS*1，表 4 は *MIS*2, 表 5 は *MIS*3, 表 6 は *MIS*4, 表 7 は *MIS*5, 表 8 は *MIS*6 のランキング結果をまとめたものである．

支持数とは，何本の配列データベースに汎化配列パターンが含まれるかを示す値である．また，*EVAL*(汎化配列パターン)のうち，モチーフに該当するインスタンスがどれだけの割合で含まれているかも示した．(以下，割合)すなわち，割合は，{モチーフに該当するインスタンス数}÷{*EVAL*(汎化配列パターン)のインスタンス数}で計算したものである．また，検索した結果が良かったかどうかは，出力した結果に，どれだけの必要な情報が含まれているか，ノイズと呼ばれる不要な情報がどれだけあったかで判断されるため，検索の評価の基準として，再現率(*Recall Ratio*)と適合率(*Precition Ratio*)を用いる．再現率とは，データベース全体の中にある全適合文献の内，何割が検索されたかをみるものであり，いわゆる検索漏れをはかるための指標である．また，適合率(精度)とは，検索で得られた文献の内，どれだけテーマに適合していたかを表すものであり，ノイズの割合をみるための指標である．また，再現率と適合率は以下のように定義する．

再現率の計算では，*A*+*B* はデータセットのデータ件数，*B* は汎化配列パターンの支持数とし，適合率の計算では，*B*+*C* は汎化配列パターン中のモチーフ配列パターンに含まれる文字列の個数と汎化配列パターン中のモチーフ配列に含まれない文字列(ノイズ)の個数の和，*B* は，汎化配列パターン中のモチーフ配列パターンに含まれる文字列の個数とすると，

$$\text{再現率}(R) = \frac{B}{A+B} \times 100(\%)$$

$$\text{適合率}(P) = \frac{B}{B+C} \times 100(\%)$$

と定義する．

また，再現率と適合率の調和平均として，トレードオフの指標である *F* 値を以下のように定義する．

$$F\ \text{値} = \frac{2 \times P \times R}{P+R}$$

この *F* 値が高いほど，性能が良いと言える．

よって，この割合と再現率と適合率と *F* 値を用いて，抽出された最小汎化集合の汎化配列パターンの検索精度を比較する．

また，汎化配列パターンにおいて，①汎化配列パターンの先頭に不要な文字がついている場合，②汎化配列パターンの語尾に不要な文字がついている場合，③ワイルドカードの前後に不要な文字がついている場合，①と②については，最小汎化集合とみなし③については，ノイズとみなすこととする．下線の文字列が該当箇所である．

例えば，①の場合の例として，

< (*Y*|*F*|*NF*|*AF*|*H*|*TF*)-*C*-*R*-*N*-*x*(*7*)-*W*-*C* >，②の場合の例として，

<*F*-*C*-*R*-*N*-*x*(*7*,*8*)-(*WC*|*WCF*|*WCY*|*WCH*|*W*|*C*|*PWC*|*VWC*|*AWC*) >,③の場合の例として，

<*F*-*C*-*R*-*N*-*x*(*7*,*8*)-(*WC*|*WCF*|*WCY*|*WCH*|*W*|*C*|*PWC*|*VWC*|*AWC*) >のような汎化配列パターンが実際に検索された．このとき，①と②の場合は，最小汎化集合とみなし，③の場合は，ノイズとみなすということである．これは，検索のときに，検索文字列の誤差を許容誤差数 1 として認めたが，検索される汎化配列パターンの文字列の長さは制限しなかったためであると考えられる．

5.3 考察

まず，*ZincFinger* データセットの場合であるが，編集距離を用いた類似性検索(表 3)とハミング距離を用いた類似性検索(表 4)，両者ともに高い精度で検索が行われている．編集距離を用いた場合とハミング距離を用いた場合の 3 位と 4 位をそれぞれ比較してみると，わずかであるがハミング距離を用いた類似性検索の方が，編集距離を用いた類似性検索よりわずかに適合率が高くなっている．つまり，ノイズの少ない検索が行えていると言える．一方，支持数に注目すると，編集距離を用いた場合は，5 位まで 1700 以上の支持数を維持しているのに対し，ハミング距離を用いた場合は，3 位までしか維持できていないことが分かる．

次に，*Kringle* データセットの場合であるが，データ件数が 90 件と少ないということもあり，かなり波のある結果となった．適合率に注目すると，編集距離を用いた類似検索(表 6)は，上位 4 位の平均は約 71%であるが，ハミング距離を用いた類似検索(表 7)は，約 98%という高い値をとっている．

よって，*ZincFinger* データセットと *Kringle* データセットで，編集距離を用いた類似性検索とハミング距離を用いた類似性検索を行った結果，ハミング距離を用いた類似性検索の方がノイズの少ない検索が行えていることが分かる．

また，アラインメント処理について着目すると，*ZincFinger* データセットでは，アラインメント処理を行う場合(表 4)の方が，アラインメント処理を行わない場合(表 5)に比べて，2 位以下に現れる支持数に大きく上回っていることがわかる．また，割合，再現率，適合率においても，ほとんどの場合に，アラインメント処理を行う場合の方が，高い値をとっていることから，やはり，アラインメント処理を行って，類似性検索を行うと精度のよい検索が行えることがわかる．

最後に *F* 値について着目すると，*ZincFinger* データセットでは，*MIS*1 (表 3)と *MIS*2(表 4)を比較すると，ほとんど同じくらいの値を示しているが，*Kringle* データセットでは，*MIS*4(表 6)と *MIS*5(表 7)を比較すると，*MIS*5(表 7)が上位で良い値をとっていることから，ハミング距離を用いた類似性検索の方が検索性能は，良いということが分かる．

表 3 : *MIS*1 のランキング

No	汎化配列パターン	支持数	割合	再現率	適合率	*F* 値
1	<*C*-[*x*(2)|*x*(4)]-*C*-*x*(3)-[*FLYCT*]-*x*(8)-*H*-*x*(3,5)-*H*>	1815	80.0	98.7	100	0.993
2	<*C*-[*x*(2)|*x*(4)]-*C*-*x*(3)-[*FLHYCTSGV*]-*x*(8)-*H*-x(3,4)-*H*>	1763	55.6	95.6	100	0.978
3	<*C*-*x*(2)-*C*-*x*(3)-[*FILHYCTSAGVM*]-*x*(8)-*H*-*x*(3,5)-*H*>	1731	58.3	90.8	96.5	0.936
4	<*C*-*x*(2,4)-*C*-*x*(3)-[*FL*]-*x*(8)-*H*-*x*(3,5)-*H*>	1725	100	91.6	97.8	0.946
5	<*C*-*x*(2,4)-*C*-*x*(3)-[*FLYTS*]-*x*(8)-*H*-*x*(3,4)-*H*>	1718	60.0	93.2	99.8	0.964
6	<*C*-*x*(2)-*C*-*x*(3)-[*FILHYNCTSDAGRVMW*]-*x*(8)-*H*-*x*(*3,4*)-*H*>	1667	50.0	90.6	99.8	0.950
7	<*C*-*x*(2,3)-*C*-*x*(3)-[*FLYSTA*]-*x*(8)-*H*-*x*(3,4)-*H*>	1631	50.0	88.3	99.6	0.936
8	<*C*-*x*(2,3)-*C*-*x*(3)-[*FLS*]-*x*(8)-*H*-*x*(3,5)-*H*>	1628	66.7	88.1	99.6	0.935
9	<*C*-[*x*(2)|*x*(4)]-*C*-*x*(3)-[*FILHYCTSGV*]-*x*(8)-*H*-*x*(3)-*H*>	1612	60.0	87.7	99.9	0.934
10	<*C*-*x*(2,4)-*C*-*x*(3)-[*FLYCTS*]-*x*(8)-*H*-*x*(3)-*H*>	1596	66.7	86.8	99.8	0.928

表 4 : *MIS*2 のランキング

No	汎化配列パターン	支持数	割合	再現率	適合率	*F* 値
1	<*C*-[*x*(2)|*x*(4)]-*C*-*x*(*3*)-[*CLYFT*]-*x*(8)-*H*-*x*(3,5)-*H*>	1815	80.0	98.7	100	0.993
2	<*C*-[*x*(2)|*x*(4)]-*C*-*x*(*3*)-[*CLYFHSTVG*]-*x*(8)-*H*-*x*(3,4)-*H*>	1763	55.6	95.9	100	0.979
3	<*C*-*x*(2)-*C*-*x*(3)-[*CLYIFHSTVMAG*]-*x*(8)-*H*-*x*(3,5)-*H*>	1731	58.3	94.1	99.7	0.968
4	<*C*-*x*(2,4)-*C*-*x*(*3*)-[*YFST*]-*x*(8)-*H*-*x*(3,4)-*H*>	1684	50.0	91.6	99.7	0.955
5	<*C*-*x*(2,4)-*C*-*x*(*3*)-*F*-*x*(*8*)-*H*-*x*(3,5)-*H*>	1678	100	91.2	100	0.954
6	<*C*-*x*(*2*)-*C*-*x*(3)-[*CLYIDRFHSTVMANWG*]-*x*(8)-*H*-*x*(3,4)-*H*>	1667	50.0	90.6	99.8	0.950
7	<*C*-[*x*(2)|*x*(4)]-*C*-*x*(3)-[*CLYIFHSTVG*]-*x*(8)-*H*-*x*(3)-*H*>	1612	60.0	87.7	99.9	0.934
8	<*C*-*x*(*2,3*)-*C*-*x*(3)-[*YFSTA*]-*x*(8)-*H*-*x*(3,4)-*H*>	1609	40.0	87.5	99.6	0.932
9	<*C*-*x*(2,3)-*C*-*x*(3)-[*FS*]-*x*(8)-*H*-*x*(3,5)-*H*>	1593	50.0	86.6	99.6	0.926
10	<*C*-*x*(2,4)-*C*-*x*(3)-[*CYFST*]-*x*(8)-*H*-*x*(3)-*H*>	1578	60.0	85.8	99.7	0.922

表 5 : *MIS*3 のランキング

No	汎化配列パターン	支持数	割合	再現率	適合率	*F* 値
1	<*C*-*x*(2)-*C*-*x*(3)-[*CDFGHILMNRSTVWYA*]-*x*(8)-*H*-*x*(3)-*H*>	1535	50	83.2	97.5	0.898
2	< *C*-*x*(2)-*C*-*x*(3)-[*CDEFGHIKLMNPQRSTVWYA*]-*x*(8)-*H*-*x*(4)-*H*>	573	40	29.3	79.2	0.428
3	<*C*-*x*(4)-*C*-*x*(3)-[*CFGHILSTVY*]-*x*(8)-*H*-*x*(3)-*H*>	442	60	22.3	94.8	0.361
4	<*C*-*x*(2)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(3)-[*CDEFHLMNPQRSTVA*]>	203	6.67	7.9	68.9	0.142
5	<*C*-*x*(2)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(4)-[*CFGHIKLMNPQRSTVWA*]>	203	5.88	2.7	22.7	0.048
6	<*C*-*x*(2)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(5)-[*DEFGHIKLMNPQRSTYA*]>	203	5.88	1.3	9.7	0.023
7	<*C*-*x*(2)-*C*-*x*(3)-[*CEFGHILMSTVYA*]-*x*(8)-*H*-*x*(5)-*H*>	202	53.8	9.9	74.5	0.175
8	<*C*-*x*(4)-*C*-*x*(3)-[*CFGHLQSTVYA*]-*x*(8)-*H*-*x*(4)-*H*>	173	45.5	8.6	92.2	0.157
9	<[*CDHIKNQRSTY*]-*x*(2)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(3)-*H*>	167	9.09	7.9	85.9	0.145
10	<[*CDEFGHKLNPQSTVWY*]-*x*(4)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(3)-*H*>	167	6.25	0.7	6.3	0.013

表 6 : *MIS*4 のランキング

No	汎化配列パターン	支持数	割合	再現率	適合率	*F* 値
1	< (*Y*|*F*|*NF*)-*C*-*R*-*N*-*x*(7,8)-*W*-*C* >	81	66.7	68.9	76.5	0.725
2	< (*Y*|*F*|*NF*|*AF*|*H*|*TF*)-*C*-*R*-*N*-*x*(7)-*W*-*C* >	60	33.3	50.0	75.0	0.600
3	< *F*-*C*-(*RN*|*R*)-*x*(8)-*W*-*C* >	31	50.0	12.2	35.5	0.182
4	< *F*-*C*-[*RN*|*RS*|*WN*]-*x*(7)-*W*-*C* >	24	66.7	25.6	95.8	0.404
5	<*F*-*C*-*R*-*N*-*x*(7,8)-(*WC*|*WCF*|*WCY*|*WCH*|*W*|*C*|*PWC*|*VWC*|*AWC*)->	22	38.5	20.0	81.8	0.321

表 7 : *MIS*5 のランキング

No	汎化配列パターン	支持数	割合	再現率	適合率	*F* 値
1	< [*YF*]-*C*-*R*-*N*-*x*(7,8)-*W*-*C* >	81	66.7	90.0	100	0.947
2	< [*YFH*]-*C*-*R*-*N*-*x*(7)-*W*-*C* >	60	33.3	65.5	98.3	0.786
3	< *F*-*C*-*R*-[*NS*]-*x*(7)-*W*-*C* >	23	50.0	25.6	100	0.408
4	< *F*-*C*-[*RW*]-*N*-*x*(7)-*W*-*C* >	15	50.0	15.6	93.3	0.267

表 8 : *MIS*6 のランキング

No	汎化配列パターン	支持数	割合	再現率	適合率	*F* 値
1	< [*FHY*]-*C*-[*RW*]-[*NS*]-*x*(8)-*C* >	69	33.3	76.7	99.1	0.865
2	< [*FHY*]-*C*-*R*-*N*-*x*(7)-*W*-*C* >	60	66.7	66.7	99.0	0.797
3	< [*FY*]-*C*-*R*-*N*-*x*(8)-*C* >	42	100	46.7	100	0.637
4	< *F*-*C*-*R*-[*NS*]-*x*(7)-*W*-*C* >	23	50.0	25.6	100	0.708

6. まとめ

本論文では，曖昧な問合せを行う際，編集距離を用いた類似性検索と，ハミング距離を用いた類似性検索の 2 種類の異なる尺度をもつ類似性検索により，最小汎化集合を抽出し，支持数に関して上位にランキングされた汎化配列パターンの精度等を評価し，2 種類の尺度の違いによる影響を比較・考察した．

適合率(精度)に着目すると，編集距離を用いた場合より，ハミング距離を用いた場合の方が高い検索精度であることが確認できた．同様に，*F* 値について着目すると，*ZincFinger* データセットでは，編集距離を用いた類似性検索とハミング距離を用いた類似性検索は，ほとんど同じ検索性能であった

が，*Kringle* データセットでは，ハミング距離を用いた類似性検索の方が，ランキングの上位で良い値をとっていることから，ハミング距離を用いた類似性検索の方が検索性能は，良いということが分かった．

また，ハミング距離を用いた類似性検索において，累進法を用いたアラインメント処理を行った方が精度の高い類似性検索が行えるということが確認できた．*F* 値についても同様に，*ZincFinger* データセットでは，すべての順位でアラインメント処理を行った方が高い値をとっており，*Kringle* データセットにおいても，1位と2位の上位の部分で，アラインメント処理を行った場合の方が高い値をとる結果になった．

課題として，今回は，*ZincFinger* データセットの検索文字は，<*C*-*x*(2,4)-*C*-*x*(3)-*L*-*x*(8)-*H*-*x*(3,5)-*H*>（可変長ワイルドカード）とし，許容誤差数は $\varepsilon = 1$ で行い，*Kringle* データセットの検索文字列は，<*F*-*C*-*R*-*N*-*x*(8)-*W*-*C*>とし，許容誤差数 $\varepsilon = 1$ として実験を行ったが，他の検索文字列の場合や，他のデータセットで類似性検索を行うとどのような影響が出るかを検証するということが今後の課題といえる．

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)，課題番号:20500137)の支援により行われた．

参考文献

[1] 宮原　和也，岡田 一志，田村　慶一，北上 始：アラインメント処理に基づく曖昧検索結果からの最小汎化集合の抽出，第 60 回電気・情報関連学会中国支部連合大会，pp.53-54，2009 年 10 月．

[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 ON SIGNAL-IMAGE TECHNOLOGY and INTERNET-BASED SYSTEMS, IEEE Computer Society Press, pp. 32-39 (2007).

[3] H. Kimura, H. Kitakami, K. Araki, and K. Tamura: A stepwise generalization method for extracting minimum generalized set from mismatch cluster, Proceedings of the 2008 International Conference on Bioinformatics and, Computational Biology (BIOCOMP'08), Vol. II, pp. 998-1004 (2008).

[4] Kazuya Miyahara, Hajime Kitakami, Yoshifumi Takahashi, Keiichi Tamura, Susumu Kuroki:
Mining Minimum Generalized Set Based on Multiple Alignments from Mismatch Cluster,
BIOCOMP 2010, pp.35-41, 2010.

[5] 阿久津達也：バイオインフォマティクスの数理とアルゴリズム，共立出版，2007 年.

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