複数クラスタリングによる効率的な塩基配列の検索手法

朱　成敏†　　遠山　元道‡

†慶應義塾大学 理工学研究科 開放環境科学専攻 〒223-8522 横浜市港北区日吉 3-14-1

‡慶應義塾大学 理工学部 情報工学科〒223-8522 横浜市港北区日吉 3-14-1

E-mail:　†joo@db.ics.keio.ac.jp,　‡toyama@ics.keio.ac.jp

あらまし　近年、生命科学分野の発展により、DNA の塩基配列の比較検索に関する研究が活発行われるようになった．しかし、塩基配列のデータは莫大な量の文字列で構成されていて、比較検索の実行のためには非常に時間がかかる．塩基配列などの遺伝情報の比較検索は早い速度も要求される同時に高い精密度が要求される．本研究は検索を行う前に塩基配列のデータをクラスタリングし、検索範囲を減らして実際に検索を行うときの検索時間の短縮を試みる．そして、クラスタリング検索から発生するエラーを減らすため、複数のクラスタリングを利用する手法を提案し、実験を通じて検証する．

キーワード　DNA, シーケンス、塩基配列、クラスタリング

An Efficient Retrieval of DNA Sequences Using Multiple Clustering

Sungmin JOO†　　Motomichi TOYAMA‡

†School of Science for OPEN and Environmental Systems, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi,Kouhoku-ku, Yokohama-shi, Japan 223-8522

‡Department of Information and Computer Science, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi,Kouhoku-ku, Yokohama-shi, Japan 223-8522

E-mail:　†joo@db.ics.keio.ac.jp,　‡toyama@ics.keio.ac.jp

Abstract A retrieval on large DNA sequence data is very important in bioinformatics. However, it takes much time to search and comparison on large DNA sequence data. In this paper, we propose multiple clustering for this problem and implement them. The result shows that multiple clustering is a more efficient approach for retrieval of DNA sequence.

Keyword　DNA, sequence, multiple clustering

1. はじめに

近年、生命科学分野の研究の発展により、遺伝情報に関する謎を解こうとする努力が持続されている．生命科学の研究が進むほど大量のデータが発生し、この大量の遺伝情報データの処理に関する研究も活発に行われている．その中に塩基配列やたんぱく質序列などの文字列データの相同性を判断する研究は生命科学分野において重要な一部である．

しかし、大量の遺伝情報データを分析して相同性を判断する処理は多くの計算量を要求するため莫大な時間がかかり、BLAST[4], FASTA[5]などの高速な検索法が用いられている．BLAST と FASTA などの手法は高速化のために対象データの一部を省略するヒューリスティックな処理を行い、精密度が高く要求される部分的な相同性の比較ではエラーが発生する可能性が高い．Needleman と Wunsch はダイナミックプログラミングを利用して生命科学データの類似度を計算する Needleman-Wunsch アルゴリズムを発表した[3]．その後、Needleman-Wunsch アルゴリズムを改良し現在まで最も頑健なアルゴリズムである Smith-Waterman アル

ゴリズムを発表した[6]. Smith-Waterman アルゴリズムは相同性の指標として重み付きの編集距離にギャップペナルティー値を加えたものを用い，その元で相同性をダイナミックプログラミングで計算する．この計算時間は比較する 2 つの配列のサイズを *N,M* とした場合，*O(NM)*となるため、その計算を行うことは時間的に現実的ではない．

そこで、遺伝情報の検索速度を向上のため,データをクラスタリングする手法も活発に研究されている．Krause ら[7]は BLAST や FASTA 検索用のクラスタを利用する手法を提案した．一貫性の条件に従ってたんぱく質情報をクラスタリングする手法であるが、単一なドメインで処理され、クラスタリング手法の一般的な測定誤差によるエラーが発生する可能性が高い．Yona らが発表した Protomap[8]はグラフに基づく手法である．Protomap は加重値として BLAST, FASTA, そして、Smith-Waterman アルゴリズムまで利用することができるが、設定したしきい値によって結果が違う場合が発生する．

そこで、本研究は精密度と検索速度の向上のため、Smith-Waterman アルゴリズムに基づいた複数クラスタリングを利用する手法を提案する．遺伝情報のデータの検索において一番重要に考えなければならないことが精密度である．精密度を高めるため複数のクラスタリングを利用し、その結果の組み合わせによって限定した検索範囲で検索を行い、検索時間を短縮する手法を提案する．

2. 背景

本節では背景となる概念と関連事項について述べる．

2.1. 塩基配列の文字列

塩基配列は DNA、RNA などの核酸において、それを構成しているヌクレオチドの結合順を記述したもののことである．DNA ではアデニン(A)、グアニン(G)、チミン(T)、シトシン(C)の 4 種類があり、RNA では、チミンのかわりにウラシル(U)になる．そして、この 4 つの文字で構成されている長い文字列で遺伝情報が表現される．この二つの文字列が類似だと判断されることは遺伝的に類似だと判断することと同じである．二つの塩基配列などの遺伝情報の文字列の類似度を判断する手法は現在、ヒューリスティックアルゴリズムを利用した BLAST と FASTA、そしてダイナミックプログラミングを利用した Smith-Waterman アルゴリズムがある．BLAST と FASTA は検索速度を、Smith-Waterman アルゴリズムは検索の精密さに注目したアルゴリズムである．

2.2. Smith-Waterman アルゴリズム

Smith-Waterman アルゴリズムは文字列の相同性を判断する最も基本となるアルゴリズムの一つである．Smith-Waterman アルゴリズムは二つの文字列の類似度を表すスコアと、このスコアの最大値を利用して文字列間の相同性を判断する．

長さを n と m とする二つの文字列 $S = s_1, s_2, ...s_n$ と $T = t_1, t_2, ..., t_m$ を比較するとき、スコアを計算するため S と T で $(m+1)\times(n+1)$ のマトリックスを D とする．

初期値 : $D(i,0) = D(0, j) = 0$,
where $0 \le i \le n, 0 \le j \le m$;

繰り返し : For $1 \le i \le n, 1 \le j \le m$,

$$D(i, j) = \max\begin{Bmatrix} 0 \\ D(i-1, j-1) + f(s_i, t_j) \\ D(i-1, j) - gap \\ D(i, j-1) - gap \end{Bmatrix}$$

$f(x, y)$ は一致する場合に与えるスコア関数で、そして gap は文字が違う場合に入れるペナルティ値である．

スコア関数 $f(x, y)$ は一致する場合にスコアを与える方法で、もし、$f(x, x) = +3,$ で設定し、不一致な探索に関しては $f(x, y) = f(x, -) = f(-, y) = -1$ を設定した場合、二つの文字が一致した場合 3 を、そして次にくる文字には 2 を与える．最終的なスコアはスコアの中に一番大きかったスコアに決める手法である．

$$score = \max_{1 \le i \le n, 1 \le j \le m} D(i, j)$$

Smith-Waterman アルゴリズムは文字列が似ているかの問題ではなく、二つの文字列の局所的な類似性を判断することができる．

2.3. K-means クラスタリング

K-means クラスタリングアルゴリズムは非階層的なクラスタリング手法の代表的な例である．あらかじめ固定された K 個のクラスタの各々にその代表であるプロトタイプを与え，それぞれの個体を最も近いプロト

タイプに割り当てることでクラスタリングを行う手法である. l 個のデータ集合を $S=\{x_1,x_2,...x_l\}$ とするとき、K-means クラスタリングアルゴリズムは次のようになる.

Input : The number of clusters K and a dataset containing l objects (X_i)
Output : A set of k clusters C_j that minimize the squared-error criterion

Begin
$m=1$;
initialize k prototypes $Z_j, j\in[1,K]$;
//初期のセンターとして任意に k 値を選ぶ.
Repeat
for $i=1$ to l do
begin
for $j=1$ to k do
compute $D(X_i,Z_j)=|X_i-Z_j|$;
if $D(X_i,Z_j)=\min\{D(X_i-Z_j)\}$
then $X_i\in C_j$;
end;
//各オブジェクトを平均に基づくクラスタに割り当てする.

if $m=1$ then $J_c(m)=\sum_{j=1}^{k}\sum_{X_i\in C_j}\left|X_i-Z_j\right|^2$;
$m=m+1$;
for $j=1$ to k do

$$Z_j=\frac{1}{l_j}\sum_{i=1}^{l_j}X_i^{(j)};$$

//各々にクラスタのオブジェクトの平均値を計算する.

$$J_c(m)=\sum_{j=1}^{k}\sum_{X_i\in C_j}\left|X_i-Z_j\right|^2;$$

Until $J_c(m)-J_c(m-1)<\xi$
End

K-means クラスタリングアルゴリズムは様々な研究で使われている手法であるが、与えられた初期 k 値を決める方法によって全く異なる結果が得られる. そのため、k 値を明確に決める方法を考えなければならない.

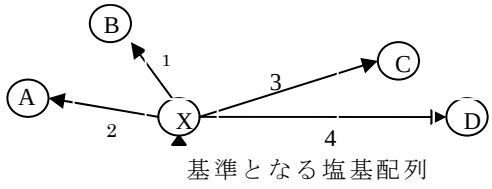
3. 問題の分析と提案する手法

本節では、前節で述べた関連事項と問題点を考えながら、本研究の中心となる複数クラスタリングによる効率的な塩基配列の検索手法に関して述べる.

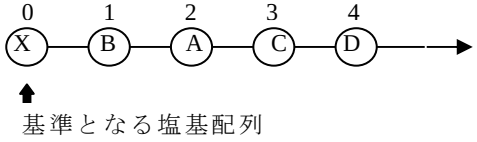
本研究で提案されるシステムは全処理と検索の二つの部分となる. 全処理は遺伝配列のデータに対してクラスタリングを行い、クラスタを生成し、その結果を索引ファイルで記録することである. そして、検索は使用者によって検索が行われて全処理で作られた索引ファイルを利用して検索することである.

大量の塩基配列データに K-means クラスタリングを適用するためには塩基配列間の相関関係による距離が必要である. そこで、本研究は塩基配列間の距離を Smith-Waterman アルゴリズムを利用して計算する方法を提案する. 全体塩基配列データの中で一つの塩基配列を任意で選び、他の塩基配列どの間の相同性を計算する. そうすると、基準の塩基配列が基準となって各々の塩基配列から相同性の高低によって並べることができる. このような方法を利用すると塩基配列間の相同性を距離に変換することができる. 基準の塩基配列から相同性が最も低い塩基配列までの 1 次元空間を生成し、その空間内の塩基配列を対象として K-means クラスタリングを行う.

K-means クラスタリングを行うため効果的に初期 k 値を選ぶことが重要である. 初期 k 値が大きいとクラスタの数が多くなって各クラスタが含むデータの数が少なくなり、最終的な検索範囲が狭すぎるようになるし、初期 k 値が小さいとクラスタが含むデータの数が増加して検索範囲が大きすぎるようになる.



a) Smith-Waterman アルゴリズムによる相同性計算.



b) 塩基配列間の相同性を距離に変換.

図 1　塩基配列間の相同性を利用する距離計算

初期 k 値は使用者が決める. まず、使用者が与えた初期 k 値で塩基配列が存在する空間を分ける. そして、

割り当てされる区間の中心点で探す．全体塩基配列 $S=\{\ x_1,x_2,...x_l\ \}$に対して初期 k 値の位置を決めるアルゴリズムは次のようになる．

Input：A dataset containing l objects (X_i) and arbitrarily chosen starting point x_0 .
Output : A set of centroid (k_i) .

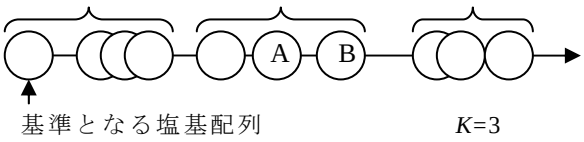
Begin
for $i=1$ to l to
compute $\max|x_i-x_0|,\min|x_i-x_0|$;
distance of D_0 = $\max|x_i-x_0|-\min|x_i-x_0|$;
// 基準 x_0 からの距離計算

$$k_j=\frac{j}{k}D_0;\qquad (0\le j\le k)$$

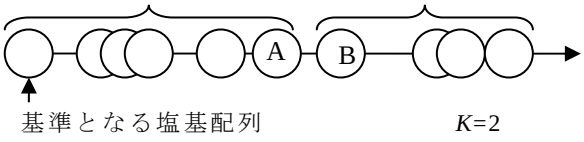
End

このように全体の塩基配列 $S=\{\ x_1,x_2,...x_l\ \}$に対して i 番目のクラスタ K の中心点 k_i を決めて K-means クラスタリングを行う．そうすると、クラスタリングの結果で k 個のクラスタ (C_i) が生成され、全体の塩基配列は $S=\{\ C_1,C_2,...C_k\ \}$になる．

しかし、K-means クラスタリングはクラスタによってオブジェクトが抜ける場合が発生する．このような問題点は使用者から不適切な初期 k 値が与えられてクラスタ内のオブジェクトの分布が悪くなった場合に発生する．



a) クラスタが適切に生成された場合．

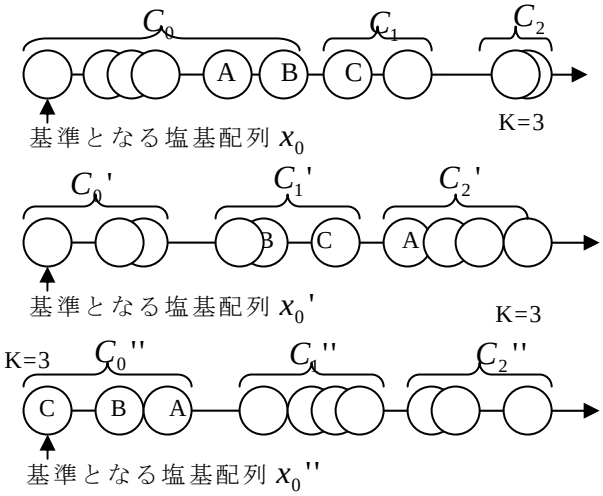


b) クラスタが不適切に生成された場合．

図 2　k 値によるクラスタの変化

図 2 において、塩基配列 A は塩基配列 B と近い距離に存在しているが、b)のように良くない初期 k 値が与えられると異なるクラスタに含まれる場合が発生する．精密度が高く要求される遺伝情報にクラスタリングアルゴリズムを適用するため、このような場合が発生しないように工夫しなければならない．本研究では図 2 のように使用者による精密度の低下を阻むために複数のクラスタリングを提案する．基準になる塩基配列を複数で選んで全体の塩基配列を複数でクラスタリングする．そうすると、図 2 のような場合が発生しても、ほかに行われたクラスタリングの結果を参考にして回避することが可能である．

図 3 は同じ塩基配列の集合で三つの塩基配列を基準として利用し、クラスタリングを 3 回行った場合である．相同性が高い三つの塩基配列 A,B,C の場合、x_0 を基準としたクラスタ C_0 では A と B だけを含んでいるが，x_0'を基準としたクラスタ C_1'では B と C を、C を基準として利用した C_0''の場合は A,B,C 全部を含んでいる．そうすると、クエリ B に対する検索範囲は C_0、C_1'、C_0''を含む集合となる．最終的には、この集合から出現頻度多い塩基配列を選んで、検索範囲として利用しする．この方法を利用して精密度を高めることができる．



検索範囲　$S=\{C_0,C_1',C_0''\}$
$=\{s_0,s_1,...,s_a,s_b,s_c,..,s_n\}$

図 3　複数クラスタリングが適用された場合

複数クラスタリングを適用するため基準となる塩基配列を決めるとき、K-means クラスタリングアルゴリズムの初期 k 値の場合のように相同性が低い塩基配列で選ぶことが効率的である．相同性が高い塩基配列で基点を決めると各々の基準からのクラスタが似ているようになって、複数クラスタリングによる効果が少な

くなる.

そのため、次のような方法を提案する. K-means クラスタリングアルゴリズムの初期 k 値の位置を決める方法と同じように、一番目のクラスタが完成した後、基準からの距離を n 個の区間で均一に分ける. そして、均一に分けた区間の中心点と一番近い塩基配列を n 番目のクラスタリングの基準として決める. 何回クラスタリングを行うかは使用者によって決められる. 複数のクラスタリングの基準となる塩基配列をを決めるアルゴリズムを整理すると次のようになる.

Input：A dataset containing l objects (X_i), a seed of K-means Clustering k, a number of Clustering n, and arbitrarily chosen starting point x_0.

Output: n datasets consist of k Clusters.

Begin

for $i=1$ to l to

compute $\max|x_i - x_0|, \min|x_i - x_0|$;

distance of $D_0 = \max|x_i - x_0| - \min|x_i - x_0|$;

// 基準 x_0 からの距離計算

begin

for $j=1$ to n to

$$x_j = \min\left| \frac{j}{n} D_0 - x_i \right|; \ (0 \le j \le n)$$

compute K-means Clustering

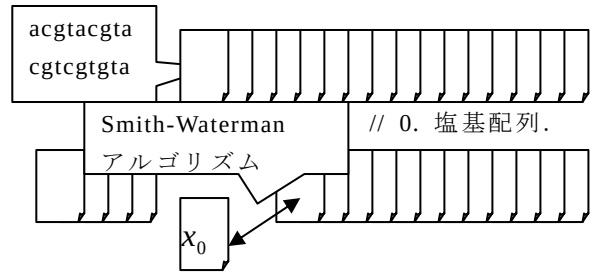
with starting point x_j and seed k;

End

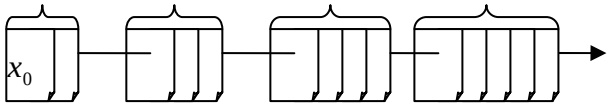
複数クラスタリングの結果として塩基配列の情報をファイルに格納する. ファイルには塩基配列のクエリが記録されたファイル名とクエリが所属されたクラスタの ID が格納されて、参照テーブルとして利用する. そして、クラスタの情報を格納するため各々基準からの空間の距離とクラスタの中心点を記録する.

4. システムの流れ

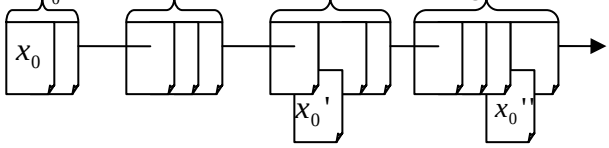
本節では前節で提案した手法による全体的なシステムの流れを説明する. システムの流れは事前処理と検索の二つに分けられる.



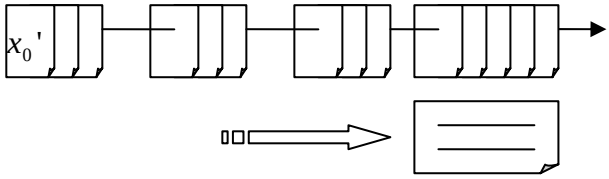
// 1. Smith-Waterman アルゴリズムによる相同性.



// 2. x_0 からの空間内で K-means Clustering を行う.

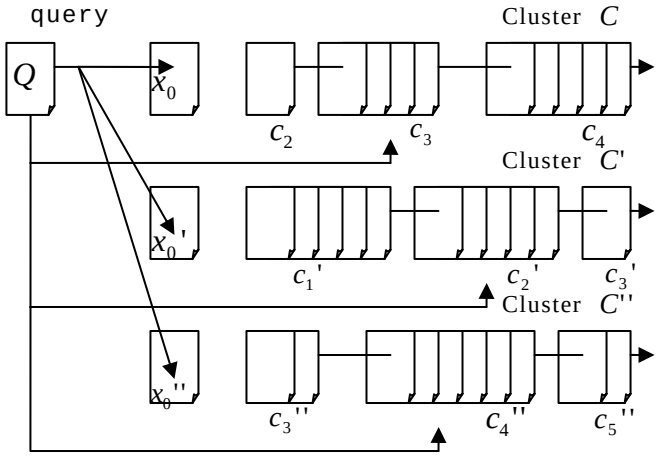


// 3. 複数クラスタリングのため原点となる配列を決める.



// 4. 複数クラスタリングを行い、その結果を記録.

図 4　事前処理の流れ



// 0．クエリの配列と各々のクラスタ原点となる配列を比較して相同性が近いクラスタを選ぶ.

$$Search(Q, \{c_3, c_2', c_4''\})$$

// 1．選ばれたクラスタだけを検索範囲として行う.

図 5　検索の流れ

本システムでは大量の塩基配列に対して複数クラスタリングを事前処理として行う．まず図 4 のように、基準となる塩基配列を決めて Smith-Waterman アルゴリズムを利用して塩基配列のデータ全体に対して相同性を計算し、相同性による距離で空間を生成する．この空間の距離を利用し、複数クラスタリングを行うための基準を決める．基準となる塩基配列が決められたら、使用者による初期 k 値で K-means クラスタリングを行い、結果を記録する．

そして、検索を行うときは図 5 のように、まず, クエリを各々基準になった塩基配列と Smith-Waterman アルゴリズムを利用して比較する．比較した結果と各々近い中心点のクラスタ全体を合わせて範囲として比較検索を行う．

5. 実験

本節では本研究で提案する複数クラスタリングによる効率的な塩基配列の検索手法について、その有効性を検証するための実験について述べる．

5.1. 環境

本研究の実験として利用したデータは約 2,000byte の塩基配列の文字列を 2,500 個である．そして、実験が行った環境は Pentium 4 3GHz の CPU, 1GByte のメモリ, OS は Fedora core 3(kernel 2.6.9) LINUX である．

5.2. 実験結果

実験での K-means クラスタリングには初期 k 値として全体データの数 2,500 の平方根である 50 を利用した．そして、クラスタリングを利用した検索範囲と塩基配列データ全体を Smith-Waterman アルゴリズムを利用した結果と比較した．格納された塩基配列データ全体を Smith-Waterman アルゴリズムを利用した結果中スコアが高い上位 50 個を正解の比較対照として利用した．そして、同じクエリに対するクラスタリングを利用した検索範囲との再現率(recall)と適合率(precision)を計算した．

$$\text{再現率} = \frac{(\text{クラスタリングで得られた中の正解数})}{(\text{正解数})}$$

$$\text{適合率} = \frac{(\text{クラスタリングで得られた中の正解数})}{(\text{クラスタリングで得られた結果数})}$$

複数クラスタリングは四つの基準を選んで行った．4回の複数クラスタリングを利用した結果は表 1 と表 2 に示している．同じクエリに対して塩基配列の全体検索と複数クラスタリングによる検索の結果から再現率と適合率を計算した．

表 1　検索結果の再現率と適合率

クエリ	再現率	適合率	検索範囲
[Q1]	1.00	0.56	89
[Q2]	1.00	0.45	110
[Q3]	1.00	0.46	107
[Q4]	1.00	0.40	124
[Q5]	1.00	0.55	99
[Q6]	1.00	0.54	91
[Q7]	1.00	0.43	116
[Q8]	1.00	0.46	109
[Q9]	1.00	0.48	105
[Q10]	1.00	0.45	111
平均	1.00	0.47	106.1

12 回検索を行って最も悪い結果といい結果を除いた．表 1 は全体検索と複数クラスタリングによう検索との検索範囲の比較と結果に対する再現率と適合率を示している．四つのクラスタリングを利用した場合は平均再現率が 1.00 であった．検索範囲を減らす実験なので、再現率が 1.00 になる時点を基準とした．平均適合率は 0.47 で平均検索範囲は 106.1 であった．全体 2,500 個の塩基配列を検索するより約 4.2%に当たる 106 個の塩基配列を検索するだけで同じ結果を得ることができた．

表 2　検索結果の時間比較

クエリ	全体検索	複数クラスタリングによる検索	比率
[Q1]	675,840	29,645	0.044
[Q2]	695,210	34,175	0.050
[Q3]	723,385	30,430	0.042
[Q4]	705,674	32,675	0.046
[Q5]	712,435	28,725	0.040
[Q6]	683,290	27,540	0.040
[Q7]	698,420	29,895	0.043
[Q8]	702,450	28,950	0.041
[Q9]	683,295	33,685	0.050
[Q10]	704,305	31,350	0.044
平均	698,430	30,707	0.044

(単位：$m$sec)

表1のクエリの実行時間の比較を表2に示している．同じクエリに対して塩基配列データの全体を検索するためにかかった平均時間は約700秒で、複数クラスタリングを利用した場合は平均約31秒であった．全体検索でかかった時間に約4.4%にあたる時間で同じ結果を得ることができた．

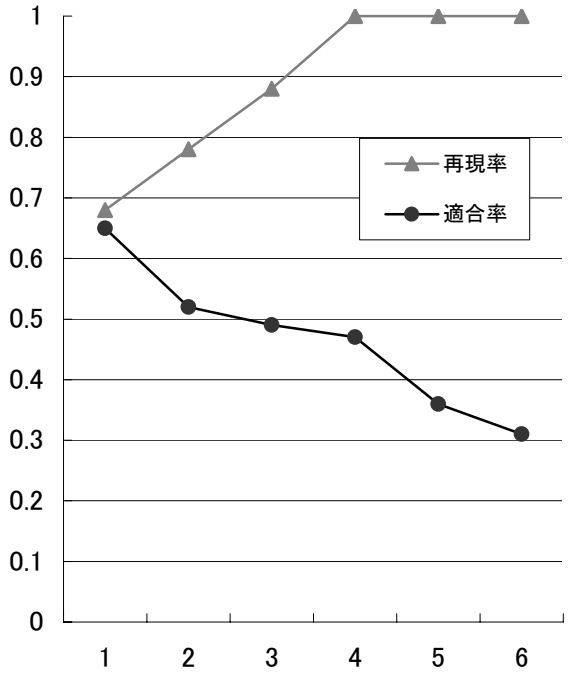


図 6　クラスタリング回数による再現率と適合率

最後に複数クラスタリングの効率を検討する．図6は同じクエリに対するクラスタリングを行った回数による再現率と適合率の測定結果である．図6はクラスタリングを1回から6回まで5つのクエリを利用した平均再現率と平均適合率を示している．4回クラスタリングを行った結果から再現率が1.00となった．そして、再現率が1.00である場合、4回クラスタリングを行ったときが適合率が最も高かったので、本実験の場合、最も効率的な回数として4回と決めて、実験を行った．

6. おわりに

本稿では検索速度と精密度が両方重要視される塩基配列検索に注目し、複数クラスタリング手法を提案した．提案した複数クラスタリング手法は塩基配列の検索において効率的であると実験を通じて証明した．そして、精密度が高く要求されるデータをクラスタリングする場合に複数クラスタリングを行うと効果的であることが分かった．

実際にシステムを遺伝情報の検索システムなど現場で利用するには、データモデルの汎用性が要求される．なぜなら本研究では一貫性があるフォーマットのデータを対象にしたからであり、現場で要求される様々な条件に対して柔軟に対処することが難しい．しかし、同じ属性を持つ大量のデータに適用すると効率的な結果を得ることができると言える．

今後の課題としては複数クラスタリングを行った結果を記録する方法として Smith-Waterman アルゴリズム以を利用したが他にも BLAST や FASTA などの適用し、比較する研究がある．また、塩基配列以外のデータにも複数クラスタリングアルゴリズムを適用し、その結果を測定することも重要な課題である．

文　献

[1] Lok-Lam Cheng, Cheung, D.W.; Siu-Ming Yiu , “Approximate string matching in DNA sequences, ” Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings. Eighth International Conference, pp. 303 – 310, 26-28 March 2003.

[2] Fang-Xiang Wu, Wen-Jun Zhang, Kusalik, A.J., “Determination of the minimum sample size in microarray experiments to cluster genes using k-means clustering , ” Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium, pp. 401 – 406, 2003.

[3] S.B.Needlme and C.D.Wunsch, “A general method applicable to the search for similartities in the amino acid sequence of two proteins,” Journal of Molecular Biology, 48, pp.443-453, 1970.

[4] S.Altscgul, W.M.W.Gish, E.W.Myers, and D. Lipman, “A basic local alignment search tool,” J.Mol.Biol.,, 215, pp.403-410, 1990.

[5] D.Lipman and W.R.Pearson., “Rapid and sensitive protein similarity searches,” Science, 227, pp.1435-1441,1985.

[6] T.F.Smith and M.S.Waterman, “Identification of common molecular subsequences,” Journal of Molecular Biology, 47(1), pp,195-197. 1981.

[7] Krause. A. and Vingron.M. ,”A set-theoretic approach to database searching and clustering,” Bioinformatics, 14, pp. 430-438, 1998.

[8] Yona. G., Linial.N. and Linial.M., “Protomap : automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space.” Proteins, 37, pp.360-378. 1999.

[9] Fa Zhang, Xiang-Zhen Qiao and Zhi-Yong Liu, “A parallel Smith-Waterman algorithm based on divide and conquer,” lgorithms and Architectures for Parallel Processing, 2002. Proceedings. Fifth International Conference on 23-25 Oct. 2002 , pp. 162 – 169, 2002.

[10] Nash. H., Blair D. and Grefenstette, J,” Comparing algorithms for large-scale sequence analysis,” Bioinformatics and Bioengineering Conference, 2001. Proceedings of the IEEE 2nd International Symposium on, pp.89-96, 2001.

[11] Yu-Fang Zhang, Jia-Li Mao, and Zhong-Yang Xiong, “An efficient clustering algorithm,” Machine Learning and Cybernetics, 2003 International Conference on, pp.261- 265, 2003.