岡山理科大学紀要 第41号 A pp. 87-95 (2005)

Multiple Sequence Alignmentの形式化手法であるMaximum Weight Trace法に対する解構築法

畝　哲広・片山　謙吾*・成久　洋之*
岡山理科大学大学院工学研究科情報工学専攻
*岡山理科大学工学部情報工学科

(2005年9月30日受付、2005年11月7日受理)

1. まえがき

現在，分子生物学分野の発展によりタンパク質のアミノ酸配列を読み取ることが可能となり解析が進められている．これらの解析により，治療法の存在しない病気の治療法の解明や未知のウイルスに対するワクチンの開発などの医学的利用，生物の進化の過程でどのような遺伝的変化が起こったかなどの生物学的利用が可能となる[2, 5, 11]．実際の解析は，タンパク質の構造を調べ特徴を導き出すことである．タンパク質の構造は，アミノ酸が連なった状態の一次構造，タンパク質の部分的な規則構造である二次構造，そしてタンパク質の空間的な折り畳みによる立体構造である三次構造と分類できる．本研究では，その中でも基本的な構造である一次構造を対象とするアミノ酸配列の Multiple Sequence Alignment (MSA) を対象とする．MSA とは，複数本のアミノ酸配列に対しギャップを配列中に挿入し，一致もしくは類似するアミノ酸が縦に揃うように並べる方法である．このような操作をアライメントと呼び，一致もしくは類似する部分が縦に揃うことで比較した配列間に存在する関連性が導き出せる．アライメントにより得られた関連性は，配列の構造や機能などの予測に利用される．もともと MSA は，生物学者がタンパク質の進化についての知識を用いて手作業で行っていたため，大規模な問題を扱うことはほぼ不可能であった．しかし，コンピュータの発展により手作業で行っていた MSA をコンピュータで行うことが可能となった．それに伴い，情報関連分野におけるアイディアを用いた様々な手法が提案されるようになった．

MSA における代表的な解法として，組合せ最適化問題で用いられる厳密解法のひとつである動的計画法を用いた MSA[10] や動的計画法を部分的に繰り返し適用する ClustalW[14]，メタヒューリスティクアルゴリズムの一つである遺伝的アルゴリズムを採用した SAGA[13] などがある．ClustalW や SAGA などの既存のアライメント手法は，任意の配列を横にずらすという操作によってアライメントを行っている．この方法は，配列の縦の関係という局所的な情報を頼りとするため大規模な問題に対しては情報が不足しがちになる．本研究では，この問題を回避する方法として Maximum Weight Trace (MWT)[6] 法を用いる．MWT 法は MSA をグラフ問題として扱うことにより，広範囲にわたって配列の縦の関係を考慮することが可能になることから，MSA のアプローチとして有望であると考えられる．我々は，この MWT 法に対して貪欲法[9] に基づく新たな解構築法を提案する．また，提案法の有効性を検証するため実験を行い実験結果から提案法の有効性を示す．

2. Multiple Sequence Alignment (MSA)

タンパク質は二十種類のアミノ酸からなり，それぞれのアミノ酸は以下のアルファベット一文字で表される．

A, C, D, E, F, G, H, I, K, L
M, N, P, Q, R, S, T, V, W, Y

また，アミノ酸配列はこのアルファベットを一列に並べた形で表される（図 1）．このアミノ酸配列を解析に利用する場合，一本のみを扱うのではなく複数のアミノ酸配列を対象とする．複数のアミノ酸配列を並べ，その配列群の特徴を明らかにするように整列させることを Multiple Sequence Alignment (MSA) と

呼ぶ．また，アミノ酸配列二本をアライメントすることをペアワイズアライメントと呼び，三本以上をアライメントすることをマルチプルアライメントと呼ぶ．一般に，アライメントを行う配列の本数が多いほどアライメントが困難であるといわれている．

アミノ酸配列の特徴を調べるために行われるのがアライメントである．アライメントとは，アミノ酸配列の一致もしくは類似する部分を縦に揃える操作である．アライメントによってアミノ酸が縦に揃えられた部分が配列の特徴といえる．アライメントによって特徴付けられる理由として，アミノ酸配列が進化（変化）する際その配列の一部にのみ変化が生じるということが挙げられる．つまり，進化前と進化後では一部分の変化しかないということになり，一部分の変化しかない配列同士は類似した構造や特徴を持っているのではないかと考えられる．主な変化としては，新たにアミノ酸が追加される挿入，アミノ酸が抜け落ちる欠失，アミノ酸が類似したアミノ酸に変化する置換がある．アライメントを行う際に欠失が起きたと思われる部分に挿入するのがギャップ“-”である．

アライメントの結果，類似する部分が縦に揃うことにより，配列間にある関連性が示される．しかし，得られたアライメントの結果が正しくなければ間違った結果を示してしまうこととなる．そこで，アライメントスコアを用いることでアライメントの正しさを調べることができる．アライメントスコアとは，アミノ酸置換行列と呼ばれるアミノ酸同士の類似性を示した行列で規定されるアミノ酸類似度を足し合わせたものである．また，スコアのつけ方によってはアミノ酸類似度の足し合わせに加えて，ギャップに対してギャップペナルティを科す方法もある．アミノ酸置換行列にはいくつかの種類があり，Dayhoff による PAM 行列[1]，S.Henikoff と J.G.Henikoff による BLOSUM 行列[4] などがある．このアミノ酸置換行列は，実際にアライメントを行う研究者がアライメントを行う配列に応じて任意に選択する．ペアワイズアライメントにおけるアライメントスコアは以下の式で求めることができる．

$$S(m) = G + \sum_{i} S(m_i)$$

ここで，m はアライメントを，m_i はアミノ酸配列の i 番目を，$S(m_i)$ は i 番目のアミノ酸類似度を表している．G はギャップペナルティである．本研究では，ギャップペナルティとしてアフィンギャップペナルティ[3] を用いる．アフィンギャップペナルティは，ギャップを開始する際に与えられるペナルティ d と，すでにあるギャップを伸ばす際に与えられるペナルティ e の二種類のペナルティを用いる．長さ g のギャップに対するギャップペナルティ G は以下の式で求められる．

$$G = -d - (g-1)e$$

同様にマルチプルアライメントのスコアは以下の式で求めることができる．

$$S(m_i) = G + \sum_{k<l} s(m_i^k, m_i^l)$$

m_i^k はマルチプルアライメントの k 本目のアミノ酸配列の i 番目を表している．このようなスコア方法を，SP スコア（sum of pairs スコア）という．得られたアライメントスコアが高いほど最適なアライメントに近いといえる．

SP スコアは，各配列におけるアミノ酸の縦の関係性をもとに得られる値であるため，このスコアを利用しアライメントを行う既存の方法は縦の関係性だけしか利用することができない．つまり，縦にどのように揃うかどうかという狭い範囲の情報しか扱えないのである．これは，大規模な問題例になるほどその影響は顕著に現れる．例えば，図 1 では下の二本の配列にギャップをそれぞれ二つずつ挿入するアライメントを行っているが，このギャップの挿入位置はこれ以外にも多数考えられる．また，下の二本の配列にギャップをそれぞれ挿入しているが，一番上の配列にもギャップは挿入可能でありその挿入位置の組合せも多数ある．実際のアライメント手法は，巧妙な方法を用いてすべての組合せを調べることはないが，大規模な問題例であればあるほどその組合せは膨大になる．この問題を解決する方法として提案されたのが Maximum Weight Trace (MWT) 法である．

```
NLFIYAL   NLFIYAL
NIYAL     N--IYAL
NYYAL     N-Y-YAL
```

図1　アミノ酸配列とアライメントの例

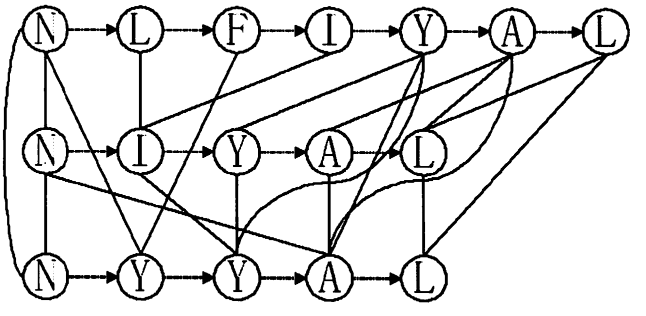

図2　アライメントグラフ

3. Maximum Weight Trace (MWT) 法

MWT 法は MSA に対する形式化手法で MSA を図2のようにグラフとして表現する方法である．このようなグラフをアライメントグラフと呼び，図1の左の配列が与えられた場合，図2のアライメントグラフを得ることができる．配列の文字一つ一つをグラフのノードに対応させ，アミノ酸同士の実現可能な縦の並びを枝で表現する．すべての枝は0より大きい値の重みを持つ．さらに配列の横の並びをアークで表す．また，アライメント全体を表現する枝の集合をトレースとし，アライメントの縦一列を表現する枝の集合を部分トレースと呼ぶ．なお，図2中において矢印付き直線がアーク，直線が枝である．また，図3下のアライメントグラフにおける直線が，図3上のアライメントを表現するトレースである．

最適なアライメントは，トレースを構成する枝の重みの総和を最大にすることで求めることができる．このような枝の重みの総和が最大となるトレースを求める問題を MWT 問題と呼ぶ．この MWT 問題に対する解法としては MWT 法の提案者である Kececioglu による分枝限定法[6]や多面的アプローチ[7]がある．これらの方法は小規模な問題例に対してのみ有効である．また，大規模な問題例に対しても利用可能な方法には貪欲法や進化的アルゴリズム[9]がある．

```
NLFIYAL
N--IYAL
N-Y-YAL
```

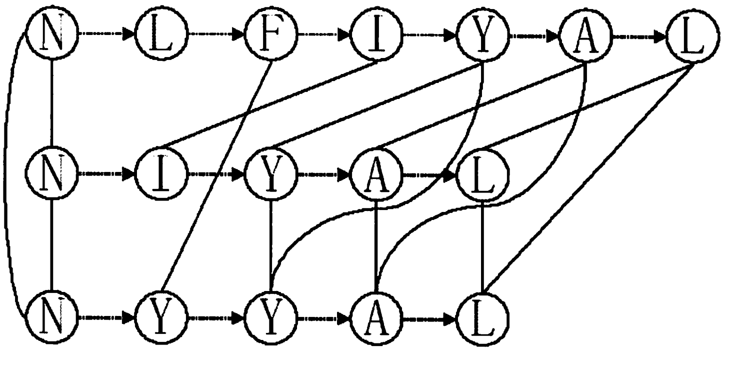

図3　アライメントとトレース

アライメントを構成する上で必要となるトレースの条件として以下のものがある．

条件1　部分トレースを構成する枝が接続しているノードは，すべて異なる配列のノードである（図4）．
図4のトレースは，二本目の配列の “Y”が一本目の配列の “Y”と “L”と同時に縦に揃うという状態を表している．このような状態は，配列の条件から存在し得ない．

条件 2　ある部分トレースを構成する枝が接続しているノードの他の枝を，別の部分トレースとすることはできない．
すでに部分トレースを構成している枝を別の部分トレースに追加するということは，あるアミノ酸が同時に複数の縦一列で整列するという実行不可能な状態を作り出すことになる．

条件 3　トレースとアークで閉路を成さない（図 5）．
図 5 のトレースは，“Y”と“Y”，“L”と“L”が縦に揃う状態になるが，これはアミノ酸の位置関係によりどちらか一方を縦に揃えると，もう一方は縦に揃うことが不可能になるため存在し得ない状態になってしまう．

条件 4　n 本の配列の部分トレースを構成する枝が接続しているノードの数は n 個であり，ノードと枝で完全グラフを成す（図 6）．
図 6 において，ノード番号 2 と 4，4 と 6 のノードに接続している枝がトレースであるが，2 と 6 のノード間には枝が存在しない．これは，アライメントの結果“L”と“Y”は縦に揃うことがないことを表している．もしノード番号 2 と 6 のノードの間に枝が存在し，トレースを構成するのであれば“L”と“I”，“Y”は縦に揃うが，図 6 の場合“L”と“I”，“I”と“Y”だけが縦に揃うことになる．しかし，“L”と“I”，“I”と“Y”を縦に揃えると必然的に“L”と“Y”も縦に揃ってしまうことになり，“L”と“Y”は縦に揃うことがないことに矛盾してしまう．

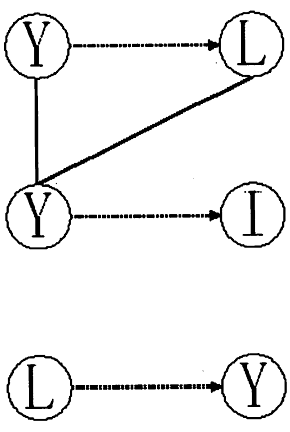

図 4　条件 1 の例

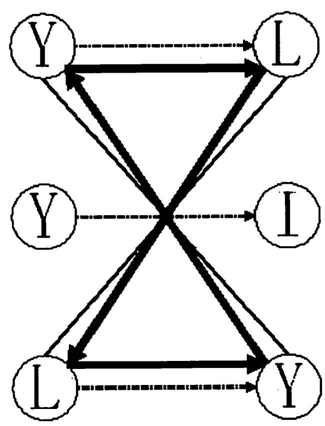

図 5　条件 3 の例

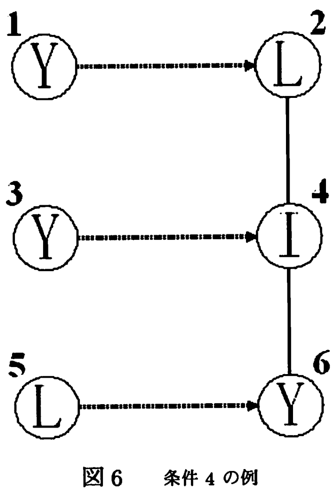

図 6　条件 4 の例

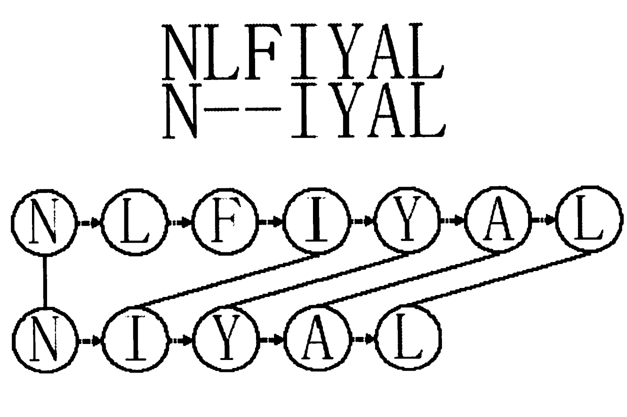

図 7　枝の張り方の例

3.1　枝の張り方

アライメントグラフの枝の張り方は，各配列ごとに二本ずつのアライメント（ペアワイズアライメント）を行いアミノ酸同士が縦に並んだ場合，それぞれのアミノ酸に該当するノード同士を枝で接続する（図

7)．この方法は，ペアワイズアライメントを行った結果から可能な縦の並びを求め，対応するノード間に枝を張る方法である．本研究では，ペアワイズアライメントに MSA における動的計画法の一つである Needleman-Wunsch 法[12] を用いる．

3.2 重みの付け方

枝の重み付けはペアワイズアライメントによる枝を張る操作と同時に行われる．本研究では，枝の重み付けの方法として Sequence Identity Score (SIS) を用いる．SIS は，ペアワイズアライメントを行った結果，縦に揃ったアミノ酸のうち一致するアミノ酸の割合をもとに得られる値である．SIS の値は以下の式で求められる．

$$SIS = \frac{m_1}{m_1 + m_0} * 100$$

ここで，m_0 はペアワイズアライメントの結果一致しなかったアミノ酸の数で，m_1 は一致したアミノ酸の数である．

図 7 の場合，アミノ酸同士が縦に並んでいる数とそのアミノ酸が一致する数が同じ，つまり縦に並んだアミノ酸が 100% 一致するので重みは 100 となる．

3.3 アライメントグラフの拡張

グラフに枝を張る方法として，ペアワイズアライメントを利用する方法を用いるが，この方法だけでは二本の配列の組合せのみを考慮するため枝の数が十分でない．そこで，グラフの推移性を用いアライメントグラフの拡張を行う．アライメントグラフの拡張は，すでに得られた枝を利用しグラフの推移性から新たな枝を追加する方法である．図 8 において，すでにノード番号 1 と 7，4 と 7 のノードの間に枝が張られており，重みはそれぞれ 50 と 70 である．このとき，グラフの推移性から 1 と 4 のノードの間に枝を張ることができる．このように二本の枝の推移性から追加された枝のことを拡張レベル 2 の枝と呼ぶ．

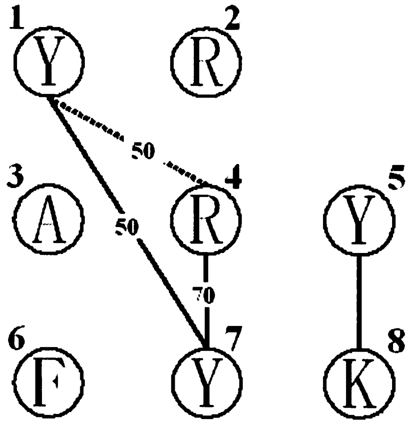

図 8　アライメントグラフの拡張の例

追加した枝の重みは以下の式で求められる．

$$w(e_t) = \frac{w_{min}(e_i)}{(k-1)}, \quad k \geq 2$$

ここで，k は拡張レベルで $w_{min}(e_i)$ は拡張する際に参照した枝のうち重みの一番少ない枝の重みである．図 8 の例では，重みが 50 と 70 の枝を対象とし拡張レベルが 2 であるから，新たに張られた枝の重みは上記式より 50 となる．

4. 提案法

本提案法は，貪欲法を基にした方法である．貪欲法は，重みの大きい枝から順にトレースに追加していく方法である．ただし，前述したトレースの条件を満たさない枝はトレースに追加されることはない．

貪欲法の手順に関して，図 9(a) を用いて説明する．貪欲法ではまず重みが最大である 75 の重みをもつ枝が選ばれ次に重みが 70 の枝が選ばれる．次に重い枝は 60 の枝であるが，トレースの条件 1 に反するのでこの枝はトレースの候補から除外される．次に重い枝は 50 の枝であるが，トレースの条件 3 に反するのでこの枝もトレースの候補から除外される．次に重い枝は 30 の枝で 2 つあるが，ノード番号 2 と 6 のノードに接続している枝はトレースの条件 1 に反するので除外され，5 と 6 のノードに接続している枝が選ばれる．最後に重みが 25 の枝が選ばれ操作は終了となる．この一連の操作によって得られるトレースは図 9(b) である．

貪欲法の弱点として，重みの大きいものを順に追加していくため本来であれば追加されるべき枝を，追加する前にトレースの条件によりトレースに追加する候補から除外してしまう可能性がある．この弱点を補う方法として，トレースに追加した枝の周辺の枝の重みを考慮した上で枝をトレースに追加する方法を考案した．

提案法の基本戦術は，トレースの重みをより大きくする有望な枝をすでに追加済みの枝の周辺の枝を考慮し選択することである．以下にその手順を示す．

1. トレースに追加可能な枝のうち，重みの最も大きい枝をトレースに追加する．
2. トレースに追加した枝の両端のノードに隣接するノードのうち，それらのノードに接続する枝の中で，追加可能な枝の重みの総和を最大にする枝を選び，トレースに追加する．
3. 2. の操作をトレースに追加する枝がなくなる，あるいはトレースを構成する枝に隣接するノードの数が配列数と同じになるまで行う．
4. グラフ全体でトレースに追加する枝がなくなるまで 1.〜3. の操作を行う．

提案法の手順に関して，貪欲法と同様に図 9(a) を用いて説明する．最初は貪欲法と同様に重みの一番大きい枝である 75 の重みをもつ枝が選ばれる．このとき，トレースに追加した枝の両端のノードに隣接するノードに注目する．隣接するノードのうち，それらのノードに接続する枝の中で，追加可能な枝の重みの総和を求めてみると，重みが 70 と 30 で 100 の枝と，60 と 50 で 110 の枝の 2 種類がある．この場合，枝の重みの総和が 110 となる枝を選択することでよりトレースの重みを大きくすることができる．このような操作をグラフ全体でトレースに追加する枝がなくなるまで行うと，図 9(c) のトレースが得られる．

ここで，図 9(b)，(c) のトレースの枝の重みの総和を比較してみると，(b) の場合重みの総和は 200 となり (c) の場合は 240 となる．図 9(a) のアライメントグラフが与えられた場合，提案法の方が良い結果を示している．これは，トレースの重みをより大きくする有望な枝をすでに追加済みの枝の周辺の枝を考慮し選択する基本戦術が効果を発揮した例であるといえる．ただし，すべての場合においてこの方法がうまくいくとは限らないと考えられる．あくまでも提案法は貪欲法をベースとしているので，貪欲法と同様の弱点つまり重みの大きいものを順に追加していくと本来であれば追加されるべき枝を，追加する前にトレースの条件によりトレースに追加する候補から除外してしまう可能性が少なからずあるということである．

5. 実験結果

提案法を評価検討するために，MWT を使用する解法である貪欲法と，進化的アルゴリズム (EA)[9]，また MWT を使用しない解法である SAGA と ClustalW とで，MSA のベンチマークである BAliBASE[15] の問題のうち 10 題を用い比較実験を行った．その結果を表 1 に示す．表 1 の値は BAliBASE によるアライメントの評価値であるアライメントスコアである．なお，このアライメントスコアは 1 に近いほど最適なアライメントに近いことを表している．EA の実験結果は文献[9] の結果を参照した．EA 以外の実験環境は，CPU:XEON 3.0GHz，OS:Fedora Core 2，開発言語は C 言語を使用した．

実験の結果，提案法は貪欲法に比べすべての問題例において良い結果が得られた．この結果は，重みの大きい枝の周辺を考慮する提案法の基本戦術が有効であったことを示している．しかし，同じ MWT を用いる方法である EA には結果は及ばなかった．これは，EA が初期解生成法として貪欲法を使用し，さらに進化的アルゴリズムにより解を改善させているからであると考えられる．なお提案法は，解構築法であるために EA にも導入可能であり，今回比較を行った EA の更なる改善も可能ではないかと考えられる．ま

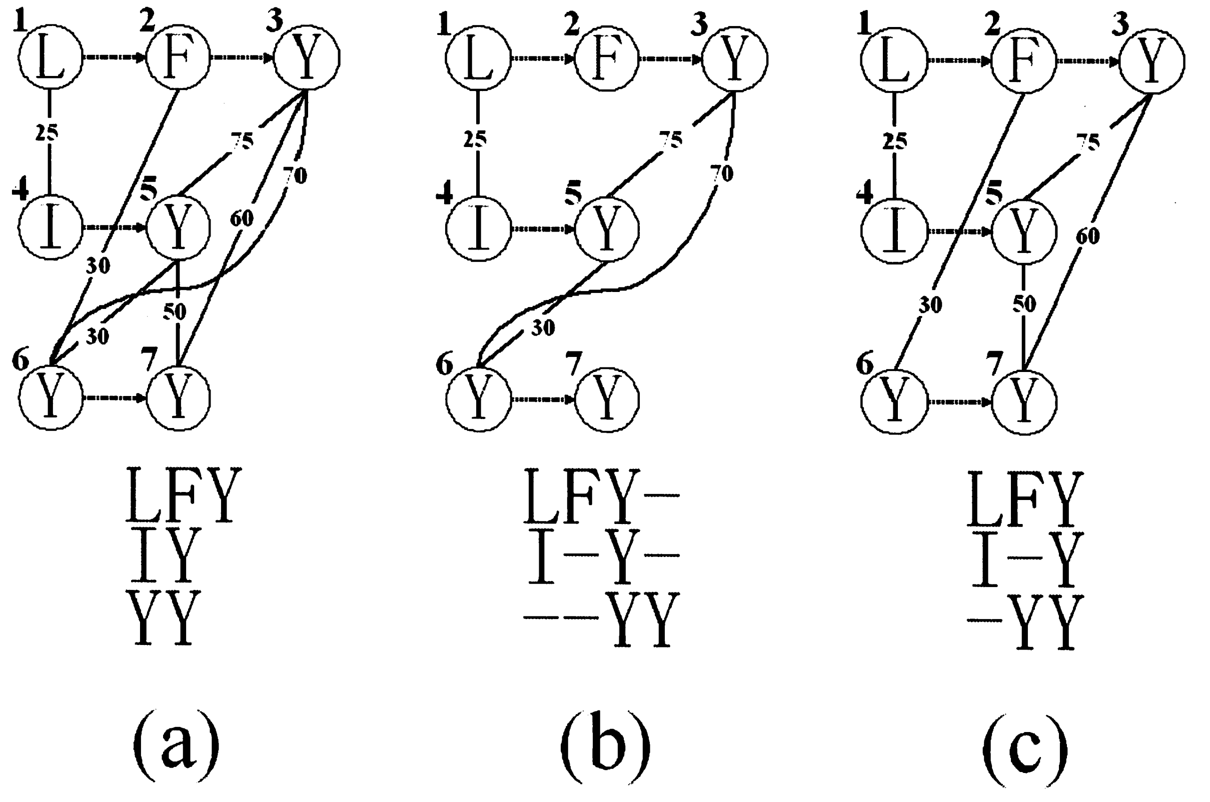

図 9　貪欲法と提案法

表 1　BAliBASE の問題例に対する実験結果

	MWT used			non-MWT used	
Instance	提案法	貪欲法	EA	SAGA	ClustalW
kinase	0.694	0.655	0.751	0.684	0.790
gal4	0.471	0.449	0.577	0.460	0.698
4enl	0.500	0.468	0.626	0.426	0.820
451c	0.714	0.662	0.820	0.668	0.649
3grs	0.458	0.412	0.469	0.485	0.410
3cyr	0.857	0.822	0.898	0.888	0.784
2trx	0.702	0.652	0.737	0.616	0.707
2pia	0.695	0.678	0.785	0.684	0.848
2myr	0.387	0.369	0.475	0.204	0.538
2hsdA	0.601	0.585	0.696	0.644	0.635

た，MWT を使用しない解法である SAGA には 10 問中 7 問，ClustalW には 10 問中 3 問で結果が上回った．このことから，MWT をベースにアルゴリズムを考える方が有効であると考えられる．

6. むすび

MSA の形式化手法である MWT に対して，貪欲法に基づく解構築法を提案した．実験結果から，提案法は貪欲法からの単純な改良にもかかわらず，比較的大きく解を改善することができた．また，提案法は貪欲法で生成した初期解を利用した EA には結果が及ばなかった．このことから，提案法を初期解としたヒューリスティックアルゴリズムを考える必要があるといえる．方法としては，部分トレースの枝を組み替えて新たな部分トレースを構築し組み替え前と比較して，重みが増大すればその組み替えを採用するといった局所探索法が考えられる．

また，MWT を用いる解法は多くの問題例において，MWT を用いない解法よりも性能が勝る傾向にあった．このことから，MWT を利用したアルゴリズムが有効であることがわかったが MWT にはいくつか考慮すべき点がある．一つ目は，グラフの枝を張る方法である．本研究では，枝を張る方法として動的計画法を用いた方法とグラフの推移性を利用した方法の二種類の方法を用いたが，このほかにも方法を検討する必要があると考えられる．なぜなら，枝を張る方法によって枝が張られる場所や全体の数が変化するため，得られる結果も変わってくるからである．二つ目は，枝の重みのつけ方である．本研究では，SIS を使用したがこのほかに両隣の枝の重みを考慮する方法[8] などが提案されており，今後 SIS との比較などを行いたいと考えている．

参考文献

1) M. O. Dayhoff, Ed, “Atlas of protein sequence and structure,” National Biomedical Reserch foundation, Georgetown University, Washington, D.C., Vol.5, pp.89–99, 1972.

2) R. Durbin, S. R. Eddy, A. Krogh, G. Mitchison, “バイオインフォマティクス–確率モデルによる遺伝子配列解析–,” 医学出版, 2001.

3) O. Gotoh, “An improved algorithm for matching biological sequences,” Journal of Molecular Evolution, Vol.162, pp.705–708, 1982.

4) S. Henikoff, J. G. Henikoff, “Amino acid substitution matrices from protein blocks,” Proceedings of the National Academiy of Sciences of the USA, Vol.89, pp.10915–10919, 1992.

5) 金久寛, “ポストゲノム情報への招待,” 共立出版, 2001.

6) J.D. Kececioglu, “The Maximum Weight Trace Problem in Multiple Sequence Alignment,” In Proceedings of the 4th Symposium on Combinatorial Pattern Matching, LNCS(684), 106-119, Springer, 1993.

7) J.D. Kececioglu, H.P. Lenhof, K. Mehlhorn, P. Mutzel, K. Reinert, M. Vingron, “A polyhedral approach to sequence alignment problems,” Discreate Applied Mathematics, Vol.104, 143-186, 2000.

8) G. Koller, G. R. Raidl, “An Evolutionary Algorithm for the Maximum Weight Trace Formulation of the Multiple Sequence Alignment Problem,” Parallel Problem Solving from Nature, 302-311, 2004.

9) S. Leopold, “An Alignment Graph based Evolutionary Algorithm for the Multiple Sequence Alignment Problem,” Master’s thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, February 2004.

10) D. J. Lipman, S. F. Altschul, J. D. Kececioglu, “A tool for multiple sequence alignment,” Proceedings of the National Academiy of Sciences of the USA, Vol.86, pp.4412–4415, 1989.

11) D. W. Mount, “バイオインフォマティクス–ゲノム配列から機能解析へ–,” メディカル・サイエンス・インターナショナル, 2002.

12) S. B. Needleman, C. D. Wunsch, “A General Method Applicable to the Search for Similatries in the Amino Acid Sequence of Two Proteins,” J. Mol. Biol., Vol.48, pp.443–453, 1970.

13) C. Notredame, D.G. Higgins, “SAGA: Sequence Alignment by Genetic Algorithm,” Nucleic Acid Research, Vol. 24, 1515-1524, 1996.

14) J. D. Thompson, D. G. Higgins, T. J. Gibson, “CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice,” Nucleic Acid Research, Vol. 22, 4673-4680, 1994.

15) J. D. Thompson, F. Plewniak, O. Poch, “BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs,” Bioinformatics, Vol.15, pp.87–88, 1999.

A Solution Construction Method for the Maximum Weight Trace Formulation of the Multiple Sequence Alignment Problem

Akihiro UNE, Kengo KATAYAMA* and Hiroyuki NARIHISA*

Graduate School of Engineering, Okayama University of Science.
**Department of Information and Computer Engineering, Faculty of Engineering, Okayama University of Science.*
1 - 1 Ridai-cho, Okayama, 700-0005, Japan.

(Received September 30, 2005; accepted November 7, 2005)

Finding a high-quality multiple sequence alignment (MSA) is one of the most important problems in computational molecular biology. It is well-known that the MSA problem can be reformulated as the problem of finding the maximum weight trace (MWT) in an alignment graph. In this paper, we propose a soluiton construction method based on greedy method for MWT. The main idea of proposed method is to select edges so that resulting trace weight is maximized. We compare the proposed method with other alignment techniques for benchmark problem instances. Experimental results show that the proposed method is effective in many instances.