TR-0851

生物学データのアライメント法に関する解説

石川　幹人

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5

Institute for New Generation Computer Technology

生物学データのアライメント法に関する解説

第２研究部部長代理　石川 幹人

1　文字列のアライメント

本稿では、文字列として与えられる生物学データの類似性を、計算機によって解析する技術であるアライメント (alignment) を解説する。この解説は、読者がアライメントについての理解を深め、自らアライメントのプログラムを開発できるようになることを目標に書かれている。なお以下の解説では、アミノ酸の文字列を扱う場合に限って話を進める。塩基の文字列を扱う場合も原理的にはアミノ酸の場合とほとんど同様であるので、ここで解説するアライメントの技術を、ＤＮＡの解析へと応用するのも容易であろう。

1.1　アライメントとは何か？

まず、具体的にアライメントとはどんなものかを例で示そう。図１ (a) にはアミノ酸配列が６本示されている。これらはレトロウイルス (retro-virus) がもつエンドヌクレアーゼ (endonuclease) という酵素の一部分である。レトロウイルスは自分の遺伝情報を宿主細胞のＤＮＡに刷り込んで増殖する。このエンドヌクレアーゼは、そうしたレトロウイルスの増殖過程において、ＤＮＡを切る働きを担う酵素である。各配列の左側にある見出しは、それぞれの酵素断片が抽出されたレトロウイルスの名前を示している。たとえば、HTLV はヒトＴ細胞白血病ウイルスのことで、AIDS ウイルスに非常に近い仲間である。

```
(a)
copia  : ILDFHEKLLHPGIQKTTKLFGETYYFPNSQLLIQNIINECSICNLAKTEHRNTDMPTKTT
M-MuLV : LLDFLHQLTHLSFSKMKALLERSHSPYYMLNRDRTLKNITETCKACAQVNASKSAVKQGTR
HTLV   : LTDALLITPVLQLSPAELHSFTHCGQTALTLQGATTTEASNILRSCHACRGGNPQHQMPRGHI
RSV    : VADSQATFQAYPLREAKDLHTALHIGPRALSKACNISMQQAREVVQTCPHCNSAPALEAGVN
MMTV   : ISDPIHEATQAHTLHHLNAHTLRLLYKITREQARDIVKACKQCVVATPVPHLGVN
SMRV   : ILTALESAQESHALIHHQNAAALRFQFHITREQAREIVKLCPNCPDWGSAPQLGVN

(b)
copia  : ------ILD--F--------HEKLLHPGIQKTTK-LF--GET-YY-FPNSQLLIQNIINECSICNL-AKT-EHR--N-TDMPTKTT
M-MuLV : ------LLD-FL--------HQ-LTHLSFSKM-KALLERSHSPYYMLNRDRTL-KNITETCKACAQ-VNA-SKS--A-VKQGTR--
HTLV   : LTDALL-ITP-VLQLSPAELHS-FTHCGQTAL-T-LQ--------GATTTEA--SNILRSCHACRG-GNPQHQMPRGHI------
RSV    : VADSQATFQAYPLR-EAKDLHT-ALHIGPRAL-S-KA--------CNISMQQA--REVVQTCPHC---NSA-PALEAG-VN------
MMTV   : ------ISD-PIH-EATQAHT-LHHLNAHTL-R-LL--------YKITREQA--RDIVKACKQCVV-ATPVPHL--G-VN------
SMRV   : ------ILT-ALE-SAQESHA-LHHQNAAAL-R-FQ--------FHITREQA--REIVKLCPNCPDWGSA-PQL--G-VN------
         ....................H....H.....................................C..C.....................
```

図 1: アライメントの例：(a) アライメント前の配列群，(b) アライメント結果

図１ (b) には (a) をアライメントした結果が示してある。同じアミノ酸や性質の似たアミノ酸が縦に同じカラム位置になるように、ところどころハイフンが挿入されている。このハイフンの一連の横方向の連なりをギャップ (gap) という。各文字配列にギャップを入れた結果、図１ (b) では、Ｈで示されるヒスチジン (histidine) が２個と、Ｃで示されるシステイン (cysteine)

が２個とが縦に揃っている。縦方向に揃った代表的文字で構成される配列を、そのアライメントのコンセンサス配列 (consensus sequence) という。図１ (b) の最下行は、コンセンサス配列である。

また、コンセンサス配列のなかに見られるパターンが、アライメントされた配列群を特徴づけるものと判断できるとき、そのパターンを、配列モチーフ (sequence motif) とか、単にモチーフと呼ぶ。Ｃが２個と Ｈが２個で特徴づけられるパターンは、「亜鉛の指」、ジンクフィンガー (zinc finger) と呼ばれる有名なモチーフである 1)。図２は蛋白質のうちジンクフィンガー部分に相当する立体構造を示している。Ｃが２個とＨが２個で亜鉛イオンに結合することで、「指」に相当する構造が形成される。ジンクフィンガーをもつ蛋白質は一般に、何本ものそうした「指」をもち、ＤＮＡの２重らせんの溝に、それらの「指」が入り込むことで、特定のＤＮＡ配列を認識し、結合する機能をもつ。エンドヌクレアーゼはＤＮＡを切る働きをもつのであるから、その酵素に、ＤＮＡ結合機能をもつ部位が存在するのは当然予想されることである。すなわち、図１ (b) でアライメントにより抽出されたパターンは、ジンクフィンガーのモチーフである可能性が高い。

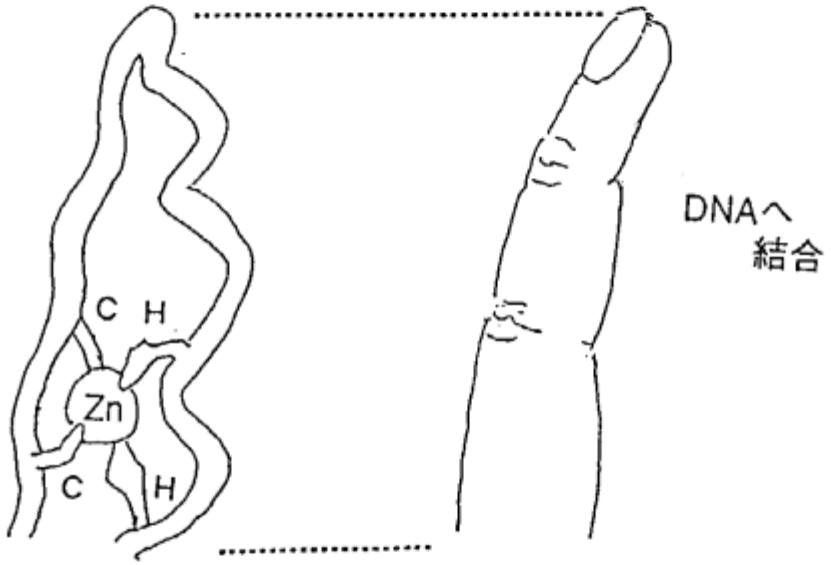


図 2: ジンクフィンガー（亜鉛の指）の構造： Zn は亜鉛イオン

アライメントのうち比較的類似したアミノ酸が縦に並んでおり、ギャップの入り方も少ない領域を、保存部位 (conservative site) という。図１ (b) では、２個のＨの周辺、２個のＣの周辺は保存性が高く、それに対し、それらの間の配列中央部は大きなギャップが入っており保存性が低い。保存性の高い部位は、配列のなかでも蛋白質の構造や機能の実現のうえで重要な部分であると推測できる。なぜなら、配列の多様性は、突然変異と自然淘汰からなる分子進化の結果で生まれたものだからである 2,3)。蛋白質の構造や機能の実現のうえで重要な部分に突然変異が起きると、多くは蛋白質の異常を起こし、その生物は淘汰されてしまう。すると、結果として、重要な配列部分は進化の過程のなかで保存されるので、生き残っている生物の同種の蛋白質をアライメントで調べると、自ずと保存部位が発見できるのである。

1.2 アライメントの利用目的

アライメントのうち、とくに配列の数が２本のものをペアワイズアライメント (pairwise alignment) と呼び、３本以上のものをマルチプルアライメント (multiple alignment) と呼んで区別する。本節では、これらのアライメントが主にどんなことに使用されているかを述べる。

ペアワイズアライメントは、基本的に配列と配列との類似性の度合を求めることに使われる。そこで、ペアワイズアライメントは第１に、マルチプルアライメントの始めの段階で、どの配列とどの配列が似かよっているかの判断に使用される。第２に、配列データベースから類似の配列を検索する手段に利用されている。たとえば、ペアワイズアライメントの手法を利用した高速配列検索ツールに FASTA 4) がある。

マルチプルアライメントは一度に複数本を比較するので、配列に存在するノイズの影響をあまり受けることなく、ペアワイズアライメントに比べて、より効果的に配列の共通性を見い出すことができる。

マルチプルアライメントの第１の利用目的は、モチーフ抽出である。すでに前節でアライメントからジンクフィンガーモチーフを抽出する例を紹介した。モチーフ抽出には、すでに判明しているモチーフを同定する場合と、新たなモチーフを発見する場合とがある。新たなモチーフの発見には、単にアライメントのコンセンサスに見られたパターンというだけでは不十分で、生物学の知見から生物学的な意義づけがされることが必要とする考え方が主流である。

第２の利用目的は、構造／機能の予測である。アライメントされた配列群のなかに配列の構造／機能がわかっているものがあれば、それとよくアライメントされる他の配列にも、同様な構造／機能があるに違いないと推測できる。図１ (b) の最初の配列 copia は、実はエンドヌレアーゼではなく、動物細胞のＤＮＡから見つかった配列をアミノ配列に翻訳したものである。copia は、他のレトロウイルスのエンドヌレアーゼとよくアライメントできるので、エンドヌレアーゼの活性をもつのではないかと推測できる 5)。

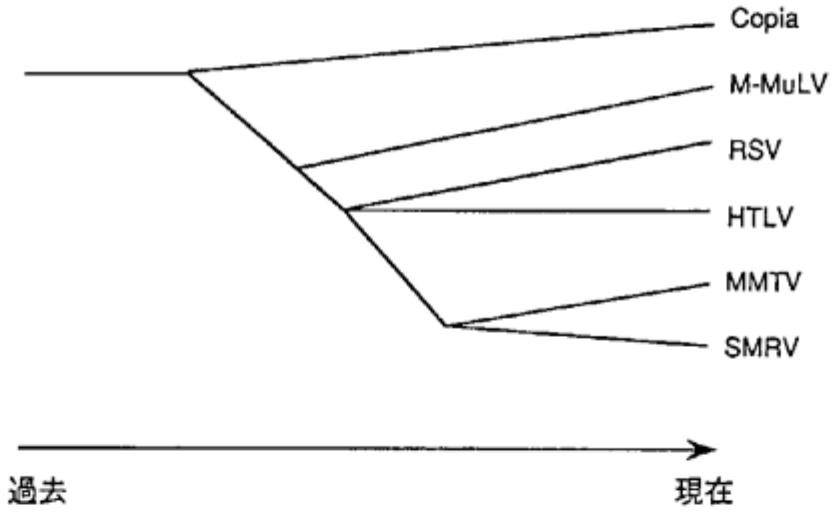


図 3: 進化系統樹の例

第３の利用目的は、進化系統樹の解析である。マルチプルアライメントの結果から、分子進化の過程でどのような突然変異（アミノ酸の置換、挿入、欠失）が起きたかを推定でき、たとえば図３のような系統樹が描ける。系統樹は過去から現在へ配列がどのような順で変化してきたか、生物種がどのように分化してきたかの歴史を示すものである。ただし、祖先の配列は現在残されていないので、いくつかの仮定から祖先の配列を推定しながら、系統関係を決めていく。マルチプルアライメントの結果から系統樹解析をする手法には、距離行列法、節約法、最尤法などがある。手法によって仮定とするところが少し異なるので、得られる系統樹の形も少しずつ異なってくる。本稿では系統樹解析手法の詳細は解説しないので、関連文献をあたって欲しい 6,7)。

1.3 アライメント手法の開発史

ここで、アライメント手法の開発の歴史をざっと振り返っておこう。機械的にアライメントを行うアルゴリズムは７０年代に開発された。そのアルゴリズムの原理は、オペレーションリサーチの分野で有名なダイナミックプログラミング (dynamic programming) であった。当時はまだ、計算機の技術が十分でなかったので、ペアワイズアライメントに応用されていたにすぎなかった。７０年代後半から８０年代にかけては、アライメントの評価値の検討や、ダイナミックプログラミングを用いたアルゴリズムの拡張の検討が加えられた。

８０年代後半になると、計算機の性能向上と、生物学分野への計算機の普及のため、ダイナミックプログラミングがマルチプルアライメントへと積極的に応用されてきた。それでも、実用的なマルチプルアライメントの問題を厳密に解くには膨大な計算時間を必要とし、主流はペアワイズアライメントを組み合わせるタイプの手法であった。この手法では、精度が十分でなく、難しいところは、もっぱら熟練した生物学者が手作業で、アライメントの課題に取り組んでいた。

近年、Genbank, PIR などの配列データベースが急速に膨らんでおり、解析すべき配列の数も増加してきたため、アライメント処理の機械化が不可避となってきた。それに呼応して、並列計算機などの高性能計算機の登場や、反復改善法の開発により、マルチプルアライメントの精度が飛躍的に向上してきており、機械化の要望に応えられる環境も整いつつある。しかし、その一方、従来の評価値体系の問題点が指摘されたり、アライメント結果から生物学的知見を得るには、アライメント過程にも生物学的知識を導入せねばならないと指摘されてきており、この分野の研究課題はいまだに山積している。

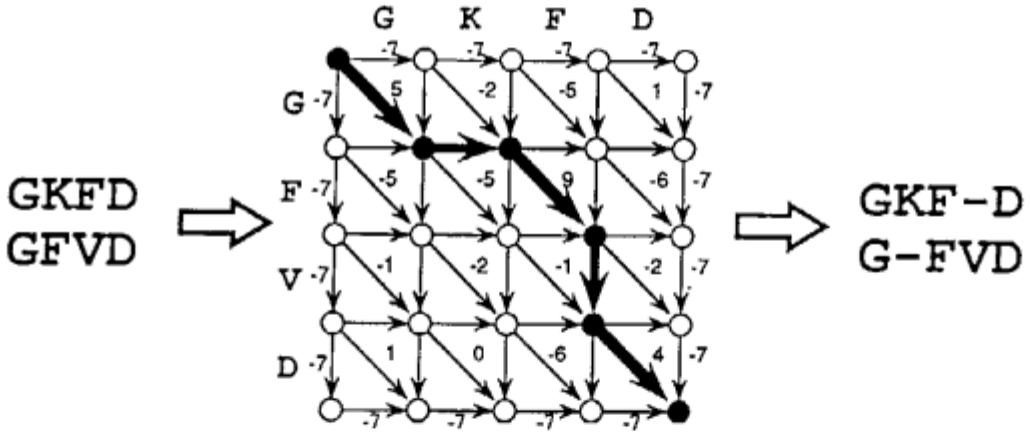


図 4: ダイナミックプログラミングによるペアワイズアライメントの解法

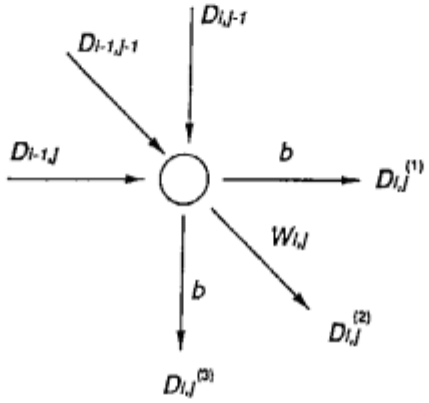
2 ペアワイズのアライメント法

配列２本（ペアワイズ）のアライメントについて、アライメント全体の評価値を最適化するようなマッチング (global matching) 問題は、ダイナミックプログラミング（以下 DP と略す）の技術で厳密に解決できる。それに対して、配列の局所的な類似性に注目してアライメントする、ローカルマッチング (local matching) 法も知られている。

2.1 グローバルマッチング

グローバルマッチングの問題は、与えられた評価基準に沿った最適解が、DPにより高速に求められる。DPは、最適化問題の解法として古くから知られていた技術であるが、生物の配列マッチングに応用されたのも古く、1970年のことである8)。

DPによるペアワイズアライメントの解法について概念的説明を、図4を用いて行う。たとえば、GKFD, GFVDという2つの短い配列をアライメントする場合、この2つの配列を図のような2次元のネットワークの辺に対応させる。斜め方向のアーク（矢）は、そのアークの位置に対応する2つのアミノ酸の類似度が割り振られる。また、縦および横方向のアークは、ギャップに対応し、ギャップを挿入するときのコストが割り振られる。このように問題を定式化すると、最適なアライメントを求めることは、このネットワーク上の最良の経路を求めることに対応する。たとえば、太いアークで表された経路が最良となったとする。この太いアークで表された経路を順に見ていくと、GとGが対応し、Kに対応するもう片方のアミノ酸はなく（つまりギャップが対応し）、Fには Fが対応し、という具合に解釈することができる。結果として図の右側にあるアライメントが得られる。



$$D_{i,j}^{(1)} = Max[(D_{i,j-1} + a), (D_{i-1,j-1} + a), D_{i-1,j}] + b$$

$$D_{i,j}^{(2)} = Max[D_{i,j-1}, D_{i-1,j-1}, D_{i-1,j}] + W_{i,j}$$

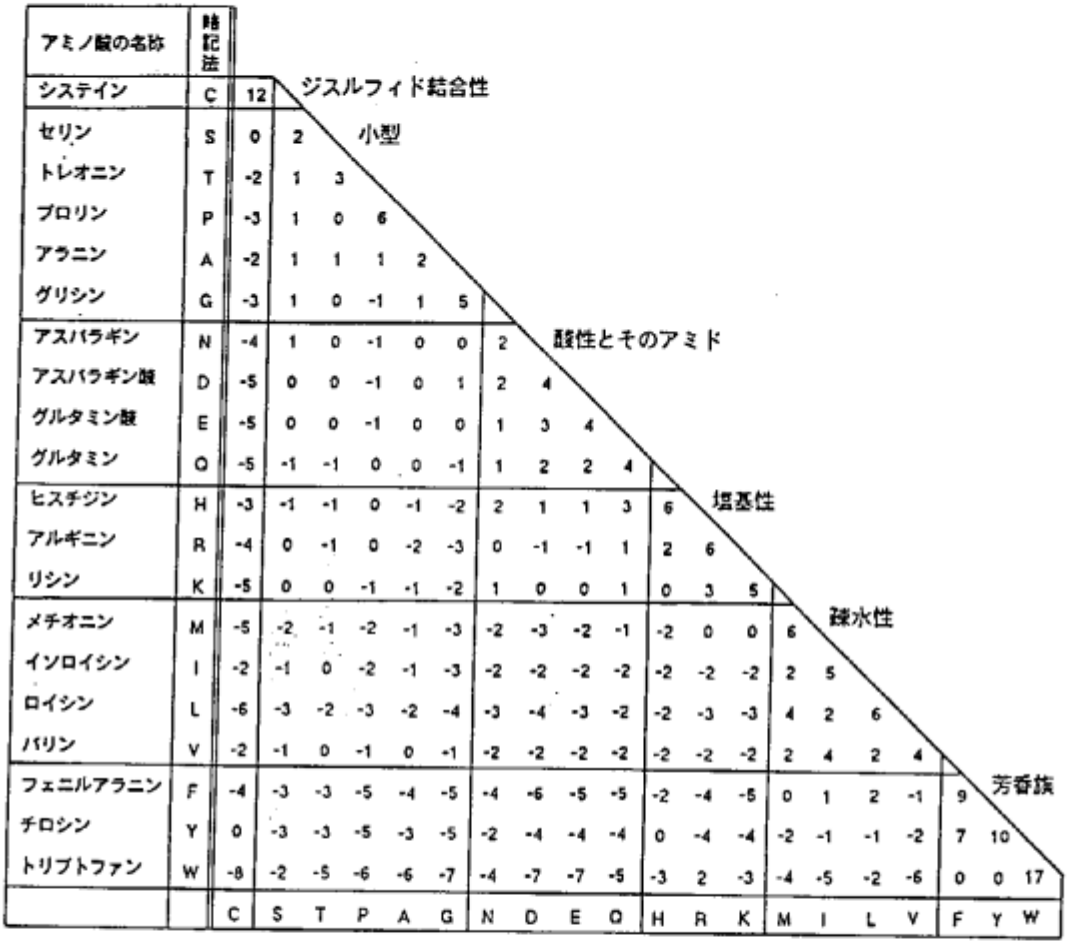
$$D_{i,j}^{(3)} = Max[D_{i,j-1}, (D_{i-1,j-1} + a), (D_{i-1,j} + a)] + b$$

図5: ダイナミックプログラミングにおける処理（ギャップコストａ＋ｂｋ）

最良の経路は、具体的には、左上の端から右下の端に向かって、各ノードに至る最良経路を段階的に決定していくことにより求めることができる。各ノードの計算を行うためには、直前の段階の各ノード（ペアワイズのときは3つある）で求まった、左上端からそこまでの最良経路の評価値を参照して、今求めたいノードに至る評価値をそれぞれ計算する。そして、それらの最良値を求めて、それをそのノードまでの評価値とすればよい。直前のどのノードを選択したかという情報も記憶しておく。この操作を右下のノードまで繰返し、最後に逆向きに、選択したノードをたどれば、ネットワーク全体の最良経路を求めることができる。

分子進化を考えると挿入や欠失が、ある程度のアミノ酸の長さでまとめて起こることから、ギャップコストはギャップの長さに依存した値にすることが望ましく、DPにも通常そうしたギャップコストが適用される9)。ギャップコストに伴う計算効率を考えると、ギャップの長

さkに対して、a＋kbのような一次式を与えるのが妥当である 10,11)。図5に、ギャップコストがa＋kbのときの、各ノードの処理を図示した。a＋kbの処理を実現するには、3方向に異なる累積評価値Dを送ると考えるとよい。斜め方向は、アミノ酸の類似度Wを最良の経路の評価値に加えるだけでよいが、縦や横方向は、注目ギャップが第一ギャップである場合、ギャップコストa (opening gap cost) を加味して最良経路を判定したうえで、ギャップコストb (extending gap cost) を加えねばならない。



アミノ酸の名称	略記法																			
システイン	C	12																		
セリン	S	0	2																	
トレオニン	T	-2	1	3																
プロリン	P	-3	1	0	6															
アラニン	A	-2	1	1	1	2														
グリシン	G	-3	1	0	-1	1	5													
アスパラギン	N	-4	1	0	-1	0	0	2												
アスパラギン酸	D	-5	0	0	-1	0	1	2	4											
グルタミン酸	E	-5	0	0	-1	0	0	1	3	4										
グルタミン	Q	-5	-1	-1	0	0	-1	1	2	2	4									
ヒスチジン	H	-3	-1	-1	0	-1	-2	2	1	1	3	6								
アルギニン	R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6							
リシン	K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5						
メチオニン	M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6					
イソロイシン	I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5				
ロイシン	L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6			
バリン	V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4		
フェニルアラニン	F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9	
チロシン	Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10
トリプトファン	W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
		C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

図 6: アミノ酸の類似度を与える Dayhoff マトリックス (PAM250)

アミノ酸の類似度には、Dayhoff マトリックス (PAM250) を用いるのが最も一般的である 12)。この尺度は、進化の過程で該当アミノ酸間での置換がどの程度起こりやすいかを推定し、数値化したものである（図6）。数値は確率の対数値になっているため、それらの足し算は複合事象の共起確率を算出したことに相当する。近頃、新しい配列データも増えてきたので、それに基づいたマトリックスの更新も試みられている 13,14)。Dayhoff マトリックスには、ギャップコストについての指針はないが、図6の値を使用するときは、a＝－7，b＝－1くらいが目安になる。もちろん、これらのパラメータを変えることによって、ギャップの入り方が変わるのであるから、問題に応じたパラメータ調整が必要となる 15)。アライメントの結果の両端に存在するギャップ（アウトギャップ）は、配列の内部に入るギャップとは異なったコストを与えられるようにするとなお良い。配列の類似部分が水平方向に大きくズレている配列対を効果的にマッチングするには、アウトギャップをゼロ（Dayhoff マトリックスにおける中立的な値）にしておくといよいだろう。

2.2 ローカルマッチング

ローカルマッチングは、配列全体としてはそれほど似てないが、局所的によく類似している部分を見い出すのに使われ、いくつかの手法が知られている。

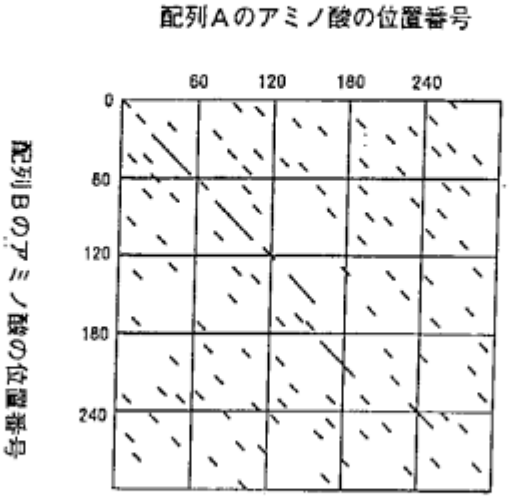


図 7: ホモロジーマトリックスの例

最も一般的なローカルマッチングは、k 個の文字の連なりであるセグメント（k-mer とか、k-tuple とも呼ばれる）を単位にして、類似セグメントを探すセグメント法である。比較される２本の配列のうちの部分セグメントを、順次照合していき、よく類似している配列位置の２次元平面に点を打っていくことで、図７のようなホモロジーマトリックス (homology matrix) を作成できる。対角線の方向に線が見える領域は、その２本の配列の類似性が局所的に高い部分に相当する。このように２本の配列の類似性が視覚的にわかりやすいため、ホモロジーマトリックスは人手で行うアライメントの補助としても、よく使用される表現である 16)。

$$D_{i,j}^{(1)} = Max[(D_{i,j-1} + a), (D_{i-1,j-1} + a), D_{i-1,j}, t] + b$$

$$D_{i,j}^{(2)} = Max[D_{i,j-1}, D_{i-1,j-1}, D_{i-1,j}, t] + W_{i,j}$$

$$D_{i,j}^{(3)} = Max[D_{i,j-1}, (D_{i-1,j-1} + a), (D_{i-1,j} + a), t] + b$$

図 8: 局所マッチングを行うときのノードの処理

ギャップを考慮した、精密なローカルマッチングの方法に Goad-Kanehisa 法がある 17)。この方法は DP を、局所マッチングが可能なように拡張したものである。DP では、図４に示すように、左上からの最良経路を各ノードの位置で決定していくものであるが、部分経路に注目して最良経路を決定していけば局所マッチングが可能となる。それを実現するため、図５の式を少し変形して、図８のようにする。つまり、最大和が、ある値 t（たとえばゼロ）に至らないときは、強制的に t にしてしまうのである。それと同時に、t が選ばれたときには、経路をリセットするようにしておくと、類似性の高いマッチングがなされたノードでは新たな経路が始まるので、とびとびの断片的な経路ができる。これを、左上から右下に行った後、右下から

左上に向かって逆に同じ処理を行い、双方の経路の共通部分をとると、局所的な類似部分が精度よく分離できる。

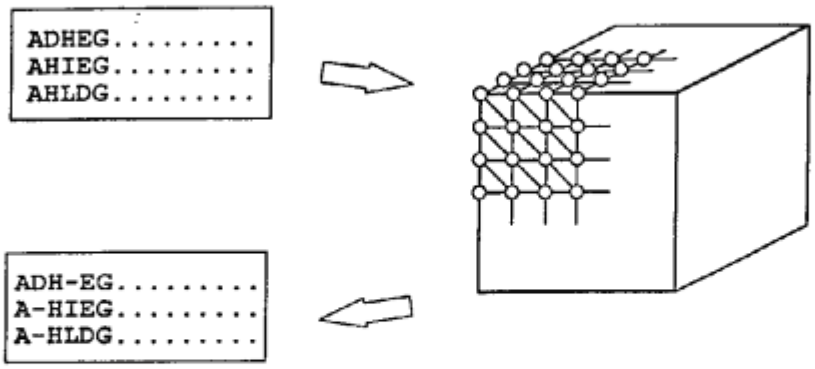


図 9: 配列３本のアライメントを行う３次元 DP の処理

2.3 マルチプルアライメントへの拡張

DP を多次元化することで、複数配列のマルチプルアライメントを行うことが、原理的には可能である。たとえば、配列３本のアライメントには、図９のように３次元 DP の処理を行えば良い 18,19)。DP は、原理的には一度に何本の配列でも同時に処理でき、与えられた評価値における最適なマルチプルアライメントが得られるはずである。ところがｎ本の配列を同時にアライメントするｎ次元の DP は、概して、配列のｎ乗の計算量とｎ乗のメモリー量が必要であり、現実的に可能なのは３次元までである。また多次元化すると、ギャップコストの扱い方も複雑になる。カラムに１個でもギャップのある配列があったら、そのカラムの評価値はゼロにするという簡便な方法 18) もなされているが、ギャップの多いアライメントの信頼性があまりよくない。ギャップコストの扱いは、処理時間とのトレードオフであり、議論の余地がある 20)。

アライメントに大幅なギャップが入ることはあまりないことから、ネットワークの対角部分（図９では、左上隅と右下隅を結ぶ部分）を適当な幅で残して、それ以外の部分をカットすると、処理をやや削減できる。Carrillo-Lipman 法 21) では、あらかじめ各組合せのペアワイズアライメントを行い、各２次元平面上の最適経路を求めておき、ｎ次元立体のうちのカットする部分の度合を決めている。ｎ次元 DP を行うときには、ｎ次元立体中の最適経路の各２次元平面に射影した経路が、あらかじめ求めておいた最適経路の幅を越えないような範囲で処理をする。この方法により、問題によっては４次元以上の DP が可能となる。しかし、それでも、配列１０本以上の実用的なマルチプルアライメントを行うと、膨大な計算時間を必要とし、DP の多次元化だけでは、一般のマルチプルアライメントの問題には対応できない。

一方、セグメント法を用いてマルチプルアライメントを行うことも可能である。セグメント法では、複数の配列にわたって共通した、あるいは類似した部分配列（セグメント）を見つけだし、その部分をピンで止め合わせるように縦に揃えることで、マルチプルアライメントを作る。この方法は、いわゆるローカルマッチングであり、配列群のなかに部分的に類似性の高い領域が存在するときには、その領域を貫くピン止めがすぐ発見でき、極めて有効である。しかし、その反面、類似性が低い領域を処理しようとすると、多数のピン止めが複雑に交差しあい、どれを優先すべきかという問題が発生する。これらの問題を解決するには、結局、配列全体を考慮せねばならず、グローバルマッチングとして対応せねばならない。実際、グローバルマッチングの前処理にセグメント法を使う試みもなされている 22)。

セグメント法のように、ローカルマッチングの観点からマルチプルアライメントを行う方法の開発もいくつかなされている 23,24) が、実用的なマルチプルアライメントの方法は、おもにグローバルマッチングの観点から開発されてきた。次章からは、そのグローバルマッチングの諸手法を解説していく。

図 10: 単純組合せ法によるマルチプルアライメント

3 組合せ法によるマルチプルアライメント

単純な組合せ法は図１０に示すように、配列１と配列２、配列２と配列３、という具合に、配列を２本ずつペアワイズにアライメントし、その結果を次々と組合せてマルチプルアライメントを作る方法である 25)。ペアワイズのアライメントには、通常ダイナミックプログラミングが用いられる。この方法は画一的で高速ではあるが、素朴に適用してしまうと、結果のアライメントの品質はあまり良くない。一番大きな問題は、一緒に比較していない配列同士の類似部分が、組合せたときにズレてしまうことである。図をみると、配列２と配列４とに同一な HIRA という部分があるが、配列２と配列４とは同時には比較されないので、マルチプルアライメントにおいて RA の部分が同一カラムに揃わないという現象が起きている。処理する配列群の類似性が低いときには、とくにこの現象が顕著に起きる。

これを防ぐには２つのアプローチがあるが、ひとつは、一度に比較する配列を増やすことであり、もうひとつは、より類似した配列から順に組合せることである。

3.1 逐次組合せ法

逐次組合せ法 26) は図１１(a) に示すような、４本の配列 (seq1, seq2, seq3, seq4) のアライメントを行うとき、次の手順で処理をする。まず、(b) seq1 と seq2 をペアワイズ DP にてアライメントし、(c) その結果と seq3 とを、グループ間の DP にてアライメントする。さらに、(d) その結果と seq4 とを再びグループ間 DP にてアライメントする。そうして、４本のマルチプルアライメントが得られる。５本以上のマルチプルアライメントも、この手順を繰り返

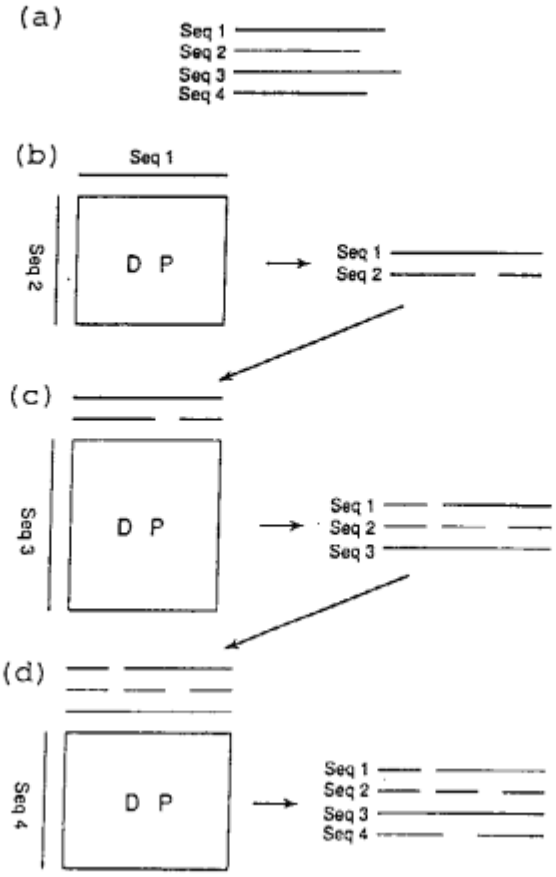


図 11: 逐次組合せ法によるマルチプルアライメント

すことで導くことができる。こうすると、新たにアライメントする配列は、すでにアライメントされているすべての配列と同時に評価されるので、信頼性が高い。

グループ間の DP とは、図１２に示すように、配列群Ａと配列群Ｂとを、配列群内の各々のアライメントはくずさずに、２次元の DP をするものである。グループ間 DP を行うときには、各アークに割当てられるアミノ酸の類似度が、そのアークの位置に対応する配列群Ａのアミノ酸と、配列群Ｂのアミノ酸との類似度の総和になる。配列群Ａが２本で、配列群Ｂが３本のときは、６通りのアミノ酸対に関する類似度の和をとることになる。類似度にはペアワイズ DP と同様に、Dayhoff マトリックスを使用するとよい。

ギャップコストの処理には注意を要する。ペアワイズ DP と違って、斜め方向のアークにもギャップが対応することがあるので、注目しているギャップがどの種類のギャップかが、ノードをさかのぼらないと判断できない。図１３に示すように、配列対でギャップ同士が対応しているカラムは、普通、ギャップとはみなさない（無視する）ので、注目しているギャップが第１ギャップ（ａ＋ｂを加える）か、第２ギャップ以降（ｂのみを加える）かは、ネットワークの経路を戻り、対応関係を調べてから判断する 27)。逐次組合せ法で使用するグループ間 DP は、必ず片側のグループの配列数が１本であるため、プロファイル化（６章参照）などの、DP 処理の効率化も可能である。

逐次組合せ法で、配列の組合せ順による影響を軽減する修正手法も、合わせて提案されている 26)。その修正手法は、いったん、マルチプルアライメントが形成されたならば、配列１本と、その１本を除いた残りのアライメントとを、再アライメントすることを次々と繰り返すものである。図１１の例で説明すると、(d) まで終えたのちに、最終アライメントから seq1 のみ取り出して、グループ間 DP で残りの配列群とアライメントする。そして、その結果から、

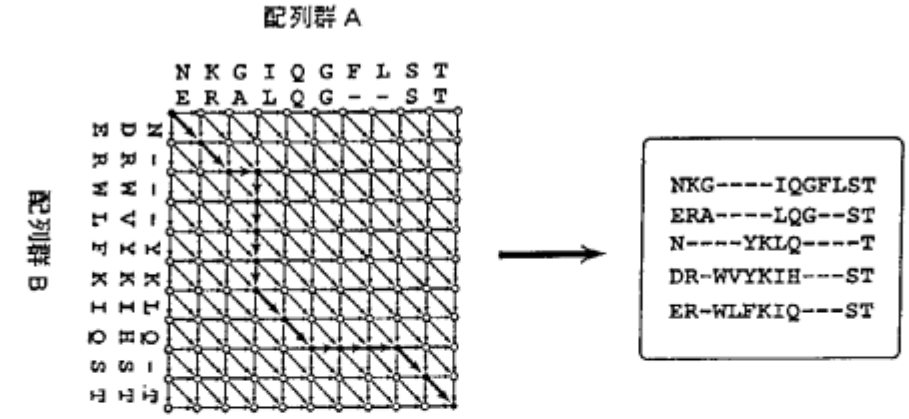


図 12: グループ間 DP によるアライメント

配列群Aのなかの１本　　LQERA^{0012}----IPG^{00}--TSF

配列群Bのなかの１本　　LQN$_{1200}$-----YKIP$_{1002}$-----SF

0：ギャップとはみなされない

1：第１ギャップとみなされる

2：第２ギャップとみなされる

図 13: グループ間 DP におけるギャップの判定

今度は seq2 を取り出して、グループ間 DP で残りの配列群とアライメントする。この操作を seq4 まで行ったら１サイクルとし、また seq1 から同じことを繰り返す。もはや、なんの変化も得られないサイクルまで至ったならば、修正を終了とする。こうした発想は、反復改善法（４章参照）へとつながっていく。

3.2　ツリーベース組合せ法

ツリーベース組合せ法は、配列間の類似性に従って描かれたツリー状の階層関係に基づいて、類似性の高い配列から順にマルチプルアライメントを形成していく手法である。類似した配列から順に組合せれば、組合せ法によって得られるアライメントの精度を大きく改善できる。なぜなら、類似した配列同士のアライメントは確実で，信頼性が高いからである。

最初に図１４を用いてツリーの作成法を説明しよう。すべての配列対についてペアワイズ DP を行い、得られたペアワイズアライメントの評価値をそれぞれの配列間の類似性とする。そうして得られた類似性評価の三角表から、(a) まず、最も類似性の高い配列対（この場合は seq1 と seq2）をツリーの枝にして結線する。その結合した配列対の、残りの配列に対する類似性

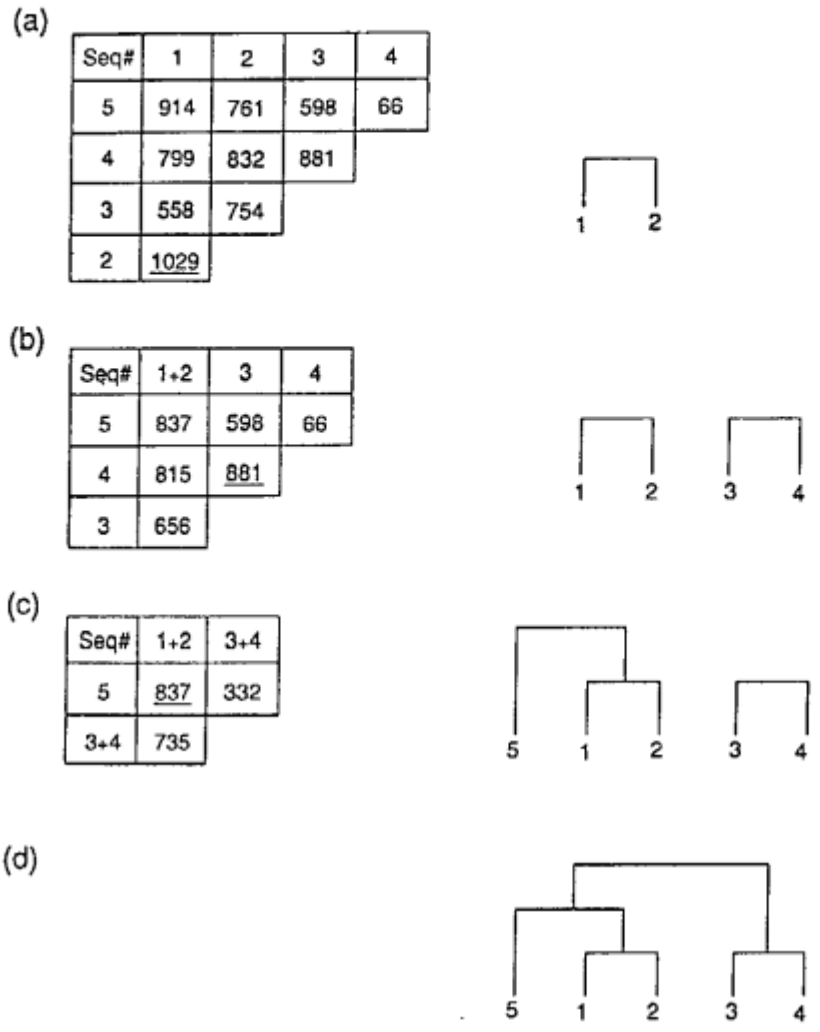


図 14: 組合せ順を決めるツリーの求め方

は、先の評価値の平均値として、三角表を更新する。そしてまた, (b) 最も類似性の高い配列対を三角表から探しだし、結線する。これを繰返す (c) と、ツリーが完成する (d)。この手順は UPGMA 28) (unweighted pair-group arithmetic average clustering) と呼ばれ、アライメントから系統樹を描くときの簡便な方法である平均距離法にも使われている 5)。

ツリーが完成したならば、ツリーの枝先から幹に向かって、枝が合わさる部分ごとに、ふたつの枝に相当するふたつの配列群を DP して、マルチプルアライメントを作っていく。そうすると最後には、全体のアライメントが得られる。その際に、ペアワイズ DP だけを行う方法と、グループ間 DP を使う方法とがある。

ペアワイズ DP だけを行う代表的な方法は、 Doolittle らの、 Progressive Alignment 29) である。配列群を組み合わせるときには、双方の配列群のうちから各 1 本をとった配列対のなかで、最も類似性の高い配列対のペアワイズアライメントに基づいて、配列群同士のアライメントを行う。たとえば、図 1 4 (d) で、 seq5,1,2 と seq3,4 とのアライメントをするときには、図 1 4 (a) の表から、 seq5,1,2 と seq3,4 との間でもっとも類似している組合せを選ぶ。するとこの場合、 seq2 と seq4 間であるから、 seq2 と seq4 との DP をとり、その結果を用いて、 seq5,1,2 のアライメントと seq3,4 のアライメントとを融合させる。

この方法は、DPを行うときに一度に比較する配列が2本であるため、単純組合せ法と同様に、初期の段階で起こったアライメントの誤りが、後々まで波及して、最後に得られるマルチプルアライメントの精度を落とす危険性がある。

その危険性を極力さけるために、彼らは事前のツリーを非常に注意深く描いている。ペアワイズアライメントの評価値は、厳密には比較する配列対の長さやアミノ酸の構成比に依存する部分があるので、統計的な規格化を行う必要がある。その規格化された評価値を求めるには、ペアワイズアライメントのだびに、ランダムにシャッフルした配列対の平均評価値を求めるため、ペアワイズアライメントを他に100回ほど行わねばならない。さらに彼らは、三角表で類似性の高い配列対が、評価値の上では均衡してふたつ以上ある場合には、該当部分のツリーを複数作っておいて、マルチプルアライメントをすべてのツリーについて行い、その結果、評価値の一番良いアライメントを採用する方法を提案している。たとえば図14(c)の三角表では、seq1+2に対して、seq5を結線するのが妥当であるが、seq3+4の評価値もそれほど悪くないので、seq 1+2とseq3+4を先に結線したツリーも念のため作成しておき、マルチプルアライメントを行った結果で良い方を選ぶようにする。

一方、グループ間DPを使う代表的なツールは、CLUSTAL 30)である。グループ間DPを使うと、事前のツリーの不正確さの問題はやや軽減される。さらに、反復改善法と組み合わせると、事前のツリーの不正確さの問題は、ほとんどなくなる。ただし、グループ間DPを行うときには、配列ごとに評価値に重みづけをしたい場合があることに注意されたい。DPされる配列群のなかには良く似た配列同士から、あまり似ていない配列同士が混在してくる。とくに、ツリーに従ってアライメントが完成に近づくと、その傾向が大きくなる。良く似た配列がたくさん存在するときには、それらの影響がグループ間DPの結果に強く出るので、それらの重みを下げる場合がある22)。また、あまり似ていない配列同士を、むりやりアライメントするのは問題があるので、そちらの重みを下げる場合もある。どちらにしろ、そうした重みづけは、配列の近縁関係を意識したもので、重みをどう付与するかは、最終的には系統樹解析のなかで考慮されるべき課題である31)。

また、計算量が多く並列計算機などの計算機資源が豊富にある環境向けではあるが、3次元DPを適用したアライメント結果を、ツリーに沿って組み合わせるマルチプルアライメント法も提案されている32)。

3.3 組合せ法と系統樹

マルチプルアライメントは、分子進化解析に必要な系統樹を作成する前段階として位置づけられることを前に述べた。にもかかわらず、マルチプルアライメントで、配列の組合せ順を求めるために、あらかじめツリーを描くのは、奇異に感じられる。実際、配列の近縁関係を正しく知るには、配列がマルチプルアライメントされてなければならないし、正確なマルチプルアライメントには、配列間の類似性の推定が不可欠である。つまり、マルチプルアライメントと系統樹は、鶏と卵のような関係になっている。

理想的には、マルチプルアライメントと系統樹が同時に得られる手法がよいのであろうが、それを厳密に実現するには、祖先にあたる配列の任意性が発生するので、n次元DPよりもさらに膨大な処理が必要である33)。祖先にあたる配列を近似的に推定しながら、マルチプルアライメントと系統樹が同時に得られる手法が、いくつか開発されている34,35)。系統樹解析を目的とする場合には、こうした方法を試みられたい。

また、マルチプルアライメントと系統樹とを交互に修正していく手法も提案されている36)。この手法では、ツリーをもとに作成したマルチプルアライメントについて、対応する系統樹を作成し、その系統樹に沿って組合せ法を行って得られたアライメント結果について、また系統樹を作成するというサイクルを、もはや系統樹が更新されなくなるまで続ける。富士通では、この手法を並列計算機上に実現したうえで、アミノ酸配列のマルチプルアライメントデータベー

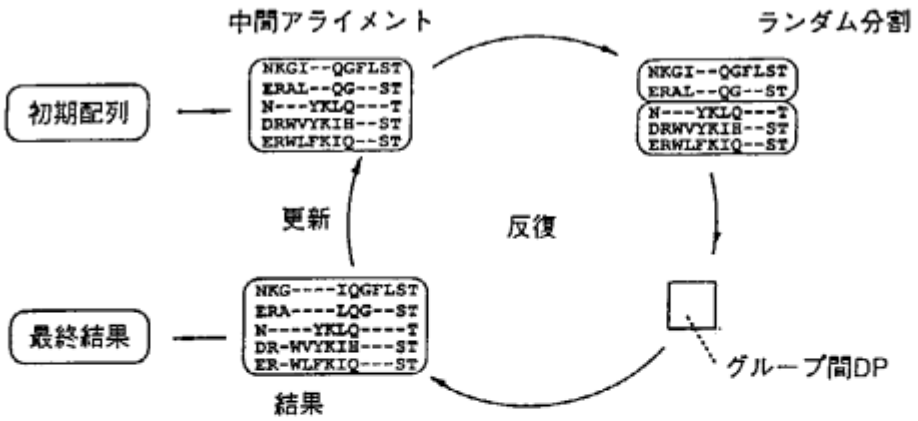


図 15: 反復改善法によるマルチプルアライメント

スを作成している 37)。

4　反復改善法によるマルチプルアライメント

反復改善法 38) は、グループ間 DP を反復的に適用することによりアライメントを徐々に改善する（図１５）。まず、何らかの方法で初期状態となるアライメントを作成する。たとえば、一番簡単な方法としては、すべての配列を左詰めにして、右の方の必要な場所にギャップを埋めるなどする。そして、これらの配列を、 ランダムに２つのグループに分割する。分割された２つのグループ間に、 DP を適用する。改善が見られた場合は、改善された状態を、新しい初期状態としてグループ間 DP を適用する。この過程を１サイクルとして、改善が行われる。このサイクルを繰り返すことにより、徐々にアライメントを改善していくことが可能である。評価値に収束がみられたら、その時点の状態を、最終アライメントとする。

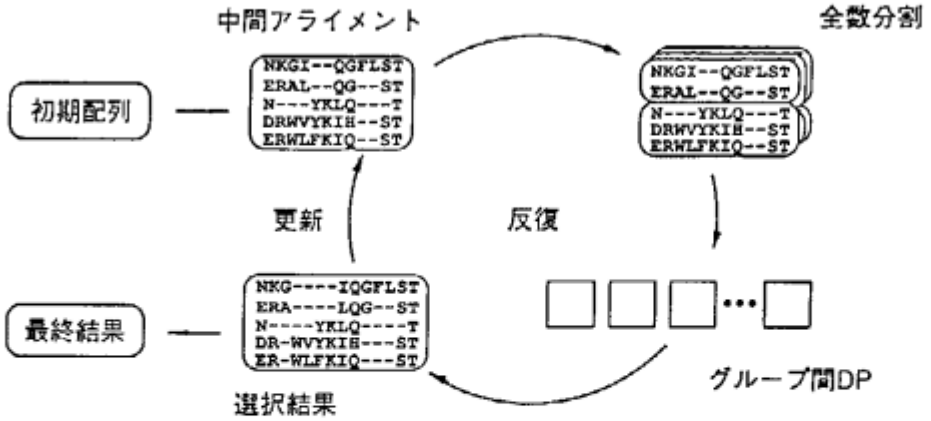


図 16: 最良探索反復改善法によるマルチプルアライメント

反復改善法は、強力な方法ではあるが、いくつかの問題点を持っており、このままでは実用規模のマルチプルアライメントに適用するのは難しい。第１は解の質の問題である。反復改善

法では評価値がだんだんと改善されるが、どの程度の解に至るかは、配列をランダムに分けるのに使用する乱数や、初期状態に大きく左右され、比較的悪い解にとどまってしまうことも多い。第２は実行時間の問題である。反復改善法で処理する配列の本数が多いと、グループ間 DP を行う時間が軽視できないうえに、ほとんどの分割が改善に寄与しないので、収束までのサイクル数も大きなものとなる。これらの問題を解決するために、以下のような、反復改善法の拡張が行われている。

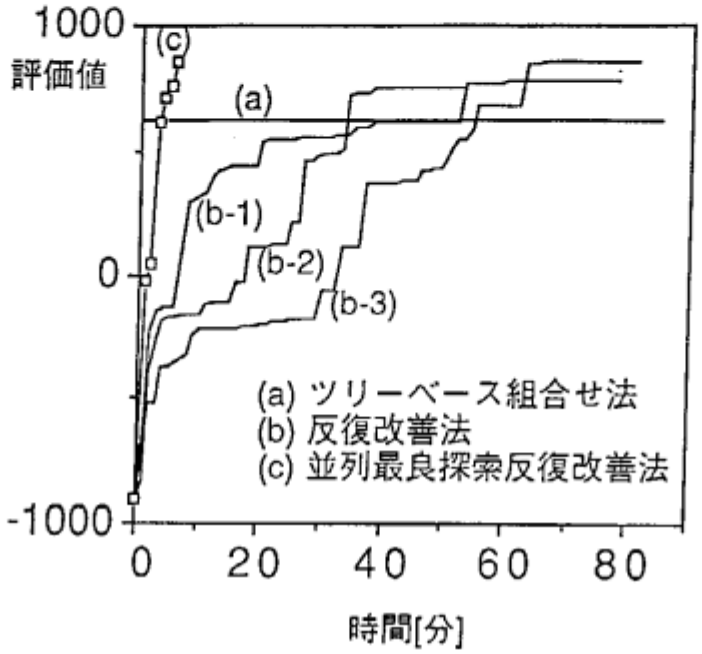


図 17: 反復改善法の性能比較グラフ

4.1 最良探索反復改善法

反復改善法の基本的アイデアは次のところにある。マルチプルアライメント全体の評価値が、それに含まれるすべての配列対の評価値の和（配列ごとに重みがついていてもよい）で与えられるという条件のもとでは、あるマルチプルアライメントを任意にふたつの配列群に分け、互いにグループ間 DP を行うと、得られるアライメントは、もとのマルチプルアライメントより、評価値がよい（少なくとも等しい）。つまり、グループ間 DP を繰返すと、マルチプルアライメントの評価値が初期状態から次第に良くなっていくのである。

しかし、次第に良くなっていくといえども、どこまでも単調に良くなるわけではない。乱数の与え方によりアライメントが不適当な方向へ進むと、比較的悪い局所最適値 (local optima) に陥ってしまい、それ以上もはや改善されない状態になる。そこで、最良探索 (best-first search) を用いると、安定して比較的良い解を求めることができる 39)。

最良探索とは、ランダムにひとつの分割を決めて DP するのではなく、すべての分割についてグループ間 DP を試し、そのうち最も良い評価値を与えるアライメントを、次のサイクルの初期値とするものである。さらに、並列計算機が使えれば、グループ間 DP を同時に実行でき、高速化が図れる（図１６）。

個々のグループ間 DP もカット（２章参照）を大幅に導入することで、高速化が可能である。反復改善法の場合、DP を繰り返し行うので、あるサイクルでカットによって最良経路が失わ

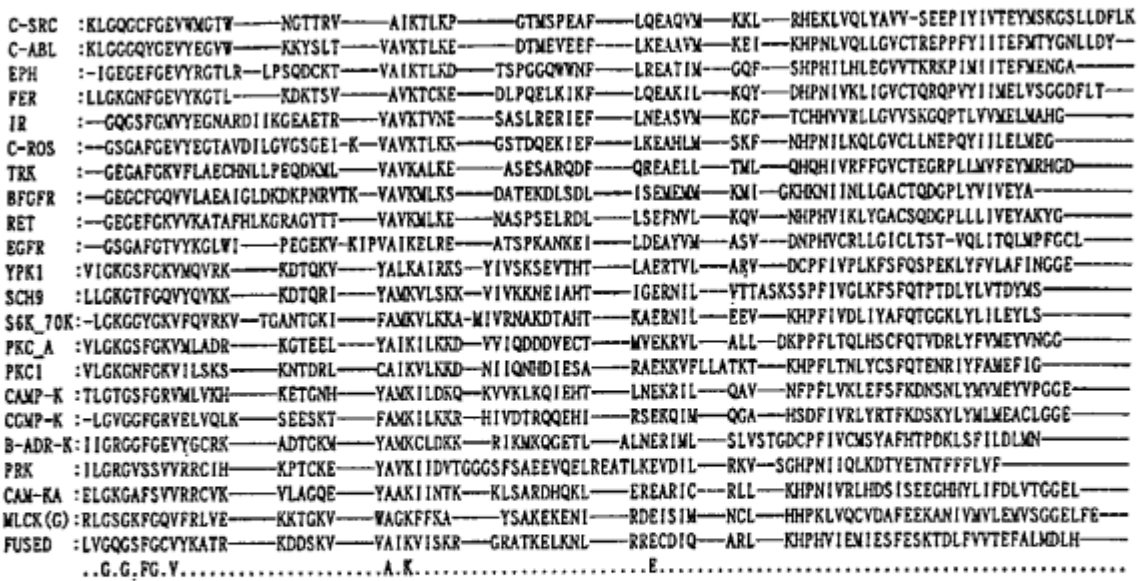


図 18: 最良探索反復改善法によるマルチプルアライメントの例

れても、その後のサイクルで補われることが多い。もし、大幅なカットを導入するならば、全配列にわたりギャップの入っているカラムの除去を、グループ間 DP を行ったあとにすべきである。カットを導入してないときは、毎回 DP を行う前に、分割された配列群の各カラムを調べ、全配列にわたりギャップが入っているカラムがあったならば、それを除去するのが処理量削減の観点から適当である。しかし、カラム除去をした後に大幅なカットを行うと、分割前の経路自体が失われて、評価値が下がってしまう事態が発生する。

図１７に示すグラフでは、80 文字７本の配列をアライメントしたときの、各手法の性能比較がなされている。水平線 (a) は、グループ間 DP を用いたツリーベース組合せ法で得られる評価値である。(b) は反復改善法による評価値の上昇線であり、３本はそれぞれ、違った乱数系列を使用した場合の時間経過を示している。どれも水平線 (a) を越えているが、越えるまでに長い時間を要している。それに対し、並列化を伴った最良探索反復改善法 (c) は、非常に早期に水平線 (a) を越えている。最終評価値も、(b) のうちの最も良い評価値と同一であった。良い乱数系列にあたると単なる反復改善法のほうが、最良探索を行うよりも、良い結果を与えることがまれにあるが、最良探索のほうが安定して良い結果を導く 27)。

上では７本の配列のアライメント行った結果を述べたが、このままでは１０本を越えるアライメントには適用できない。というのは、配列 n 本の異なる分割の数が、2の (n-1) 乗にも達するので、すべての分割をとても調べ切れない。これほどの数では、たとえ並列度の高い並列計算機をもってしても、容易には対応できない。そこで、調べる配列の分割法を限定する手法を用いる 40)。グループ間 DP による評価値の改善は、分割された配列群の配列数が不均等なほど大きい。たとえば、２０本の配列は、１本と１９本に分けて DP したほうが、１０本と１０本とに分けて DP するよりも、はるかに効率良く改善できる。その理由は、双方の配列群内の配列数がともに多いと、配列のカラムごとの特徴が平均化されてしまい、DP の最適化が十分に働かないからである。

図１８は、80 文字２２本の配列を、最良探索反復改善法で解いた結果である。分割は、1, 2 本に限定しているので、１本と２１本、２本と２０本の分割しか評価していない。256 台構成の並列計算機で、およそ１０分で解決された。これらの配列は、プロテインキナーゼ (protein kinase) という、相手の蛋白質のアミノ酸をリン酸化する働きを担う酵素の一部分である。先頭のあたりのコンセンサス配列は、ATP を把持する部位を示すモチーフである。

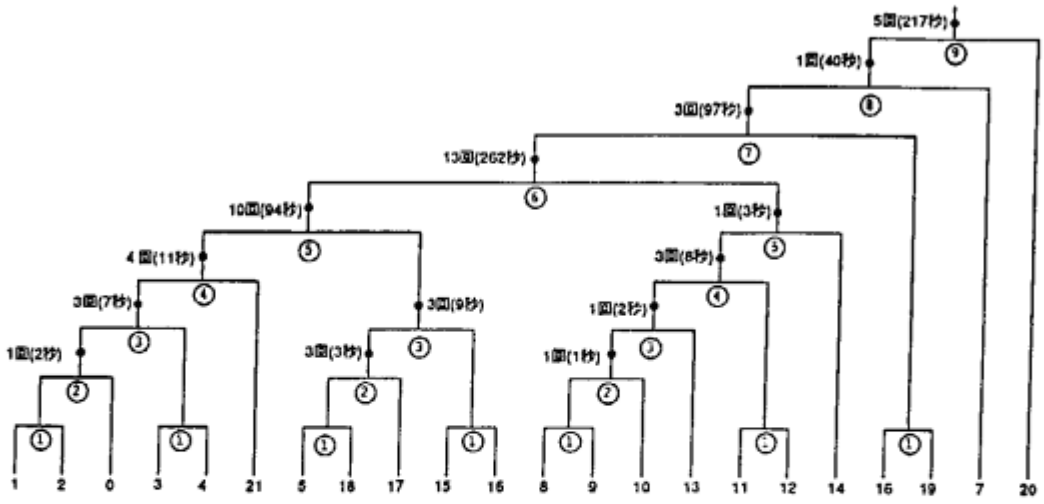


図 19: ツリーベース反復改善法の実行過程の例

このアライメントは効果的に解決できたが、限定分割手法にも問題点がある。たとえば、アライメントすべき２０本の配列が、ある点で似ている１０本の配列と、他の点で似ている１０本の配列とから成っていたらどうだろう。限定分割をいれた反復改善法では、どちらか先に揃い始めた１０本はうまくアライメントされるが、他の１０本は、始めの１０本のアライメントに影響されてうまく揃わない。この問題に対処するには、配列を分類したうえでアライメントするか、ときどき限定分割を解除してやるとよい。

4.2 ツリーベース反復改善法

ツリーベース反復改善法は、3.3.2 と同様に、始めにツリーを作成し、そのツリーに従って、グループ間 DP で、配列を次々とアライメントしていくが、配列群をアライメントしたのちには必ず、反復改善法を適用するものである 40)。

たとえば、図１９に示すようなツリーが前処理で得られたとする。図の下にある数字は配列の番号を示しており、図の例では、配列１と配列２は類似しており、それらに対して配列０が類似しているのがわかる。次に、このツリーに従って、類似している配列から順にグループ間 DP を使って、配列を組み合わせ、アライメントしていく。ただし、グループ間 DP を行った結果が配列３本以上のアライメントになるときは、そのあと必ず収束するまで反復改善を行い、その時点のアライメントを確実なものとする（配列３本のアライメントに限り、若干時間はかかるものの、直接３次元 DP を行うのも良い）。図の黒い丸で示される点は、反復改善を行った個所を表している。それぞれ、1,2 本限定で、何サイクル反復改善（最良探索をしている）を行って、何秒かかったかを、「回、秒」で示している。一般に、本数の多いアライメント同士がグループ間 DP された後の反復改善は、サイクル数が多くかかるうえに、配列の長さが（ギャップが挿入されて）増えているため、１サイクルの処理時間も大きくなる。

上の時間表示は並列計算機を使用した実績を記してある。ツリーベース反復改善法も、多分に並列計算機向きの手法である。この手法は、反復改善時の最良探索の並列化に加えて、ツリー状に組み合わせるときの、ツリーの各レベルが並列化できる。図の中では同じ番号のレベルは、それぞれ個別に DP 可能である。最上位のレベルは、並列に実行できるグループ間 DP はないが、その分、グループ間 DP の後の反復改善に、多くの並列要素がある。実行時間は、最良探索反復改善法に比べ収束が早いので、１割程度短く済む。（評価値の比較は図２４）

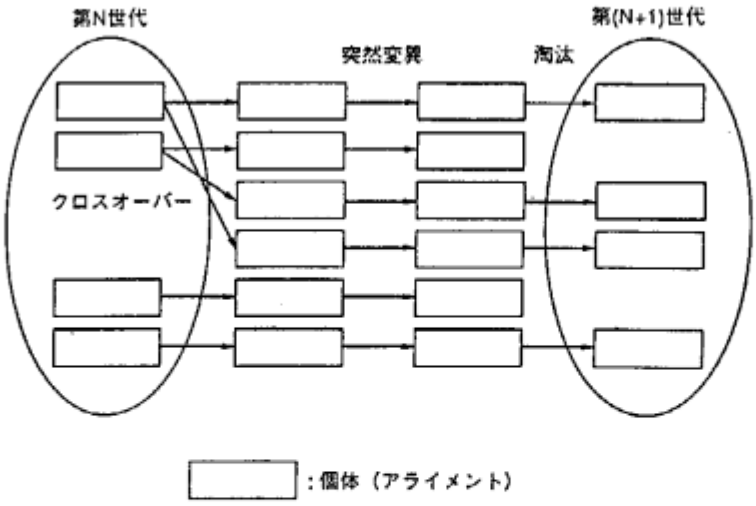


図 20: 遺伝的アルゴリズムの概念

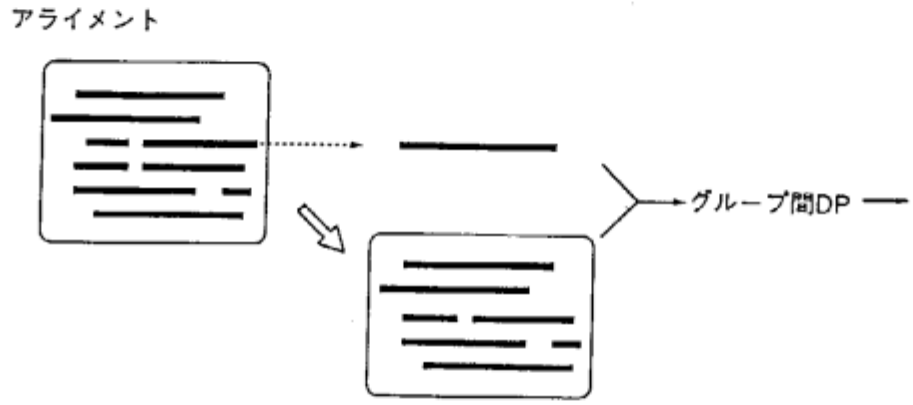


図 21: 突然変異の適用法

4.3 遺伝的アルゴリズムの導入

反復改善法で通常は限定分割を行いながらも、ときどき限定をせずに分割する手順が、遺伝的アルゴリズム (genetic algorithm) を導入することで、自然に実現できる。遺伝的アルゴリズムは図２０に示すように、多くの「個体」からなる集団が世代交代を繰り返すことで、「個体」の平均「適応率」をあげようとするものである 41)。「個体」を組合せ問題の解、「適応率」を解の評価値に対応させることで、組合せ最適化問題を解くことができる。遺伝的アルゴリズムは、マルチプルアライメントだけでなくモチーフ抽出 42) などにも応用されている。

マルチプルアライメントを解く場合には、「個体」に暫定的なマルチプルアライメントを、「適応率」にそのアライメントの評価値を対応させるとよい 43)。そして、突然変異、クロスオーバー、淘汰を次のように適用する。突然変異は、マルチプルアライメントからランダムに１本を抜きだし、残りとグループ間 DP する（図２１）。これは、１本限定分割の反復改善法に相当する。通常の遺伝的アルゴリズムで突然変異というと、評価値が良くなる場合も悪くなる場合もあるが、この適用の仕方であると、評価値は悪くなることはない。

クロスオーバーは、集団のなかから比較的評価値の良いアライメントふたつを選びだし、そ

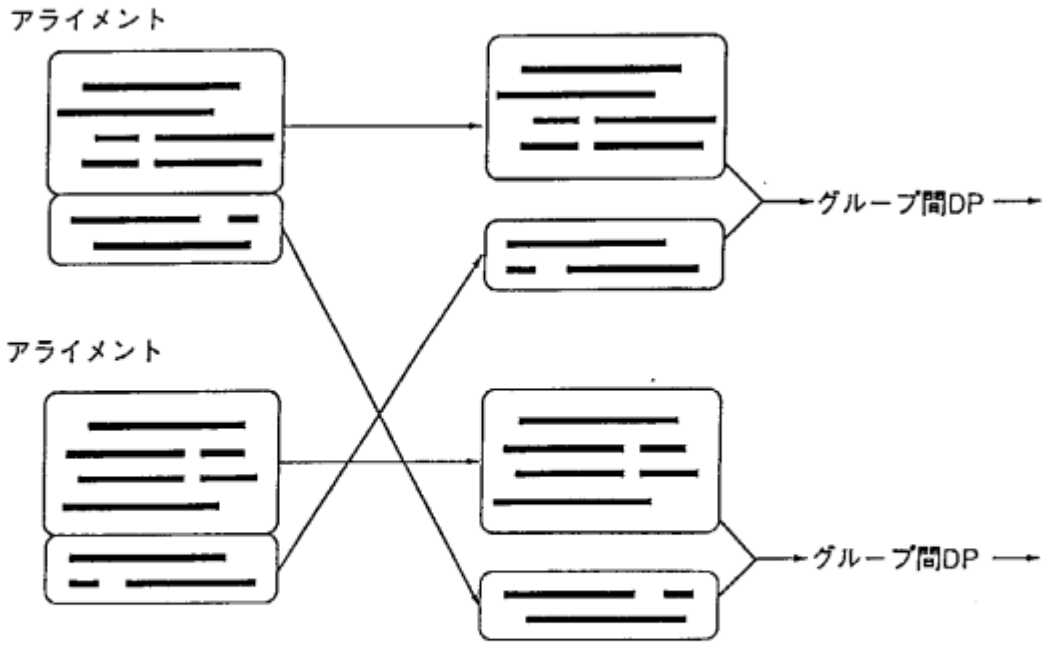


図 22: クロスオーバーの適用法

れらの配列群をランダムにふたつの配列グループに分ける。それらの片側の配列グループを交換し、グループ間 DP で融合する（図２２）。これにより、アライメントのなかに、比較的良く揃っている部分があれば、他のアライメントのなかにある別な良く揃っている部分と、互いに融合され、ひときわ高い評価のマルチプルアライメントができる可能性が生まれる。クロスオーバー１回で新たなアライメントが２つ生まれるが、増えた個体数は、淘汰によって調整される。淘汰は、集団のなかから比較的低い評価値のアライメントを捨ててしまう操作である。

図２３では、最良探索反復改善法と遺伝的アルゴリズムとの性能比較を示している。図１８と同様な問題を、256 台構成の並列計算機を用いて解いた結果である。最良探索反復改善法では、1,2 本限定分割を用いており、10 分程度で解が出ているが、これは比較的悪い局所最適解である。遺伝的アルゴリズムでは集団個体数 255、世代当たり淘汰率 10 %、1 世代時間 80 秒にセットしている。1 世代時間 80 秒というのは、この問題の場合、突然変異回数にしておよそ６回、クロスオーバーも行われる個体については、およそ４回に対応する時間である。収束条件は集団の最大評価値の個体が、100 世代にわたって変化しないこととした。

遺伝的アルゴリズムの最良個体の最大評価値上昇をみると、実行開始から 10 分後に、最良探索による解と同程度の値に至るものの、その後も徐々に評価値を上げ、２時間後に相対的にかなり良い値で収束している。集団の平均評価値は次第に最大評価値に接近していっており、最終的には集団のほとんどの個体が同一のアライメントを持つに至る。この問題の場合には、５時間後には、約８割の個体のアライメントが同一になった。

図２４では、図１９と同様な問題を、各手法で３０題解いたときに得られた平均評価値を比較している。遺伝的アルゴリズムで解いた結果が、平均して最も良かった。しかし、時間を比較すると、他の手法の実行時間が５分～１５分であるのに対して、遺伝的アルゴリズムは２～３時間を必要とする。

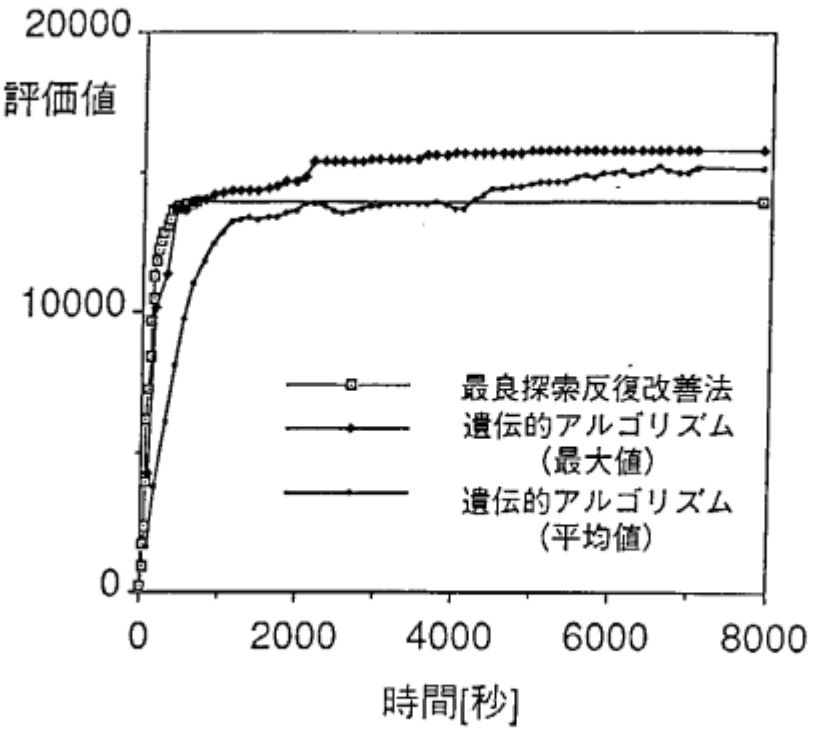


図 23: 遺伝的アルゴリズムの性能比較グラフ

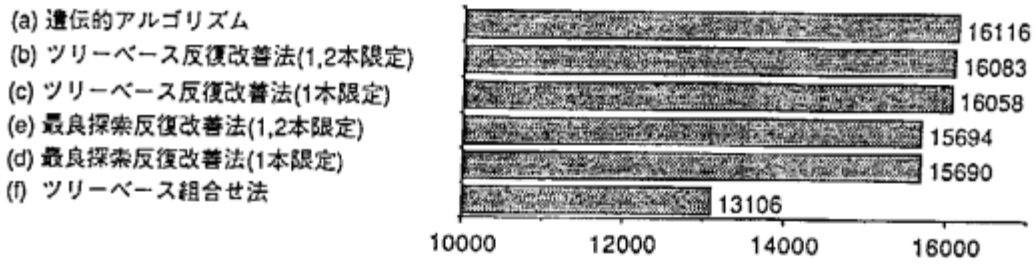


図 24: 各手法による解の平均評価値比較

5 マルチプルアライメントの精緻化

反復改善法では、マルチプルアライメントに与える評価値の体系は変えずに、とにかくその体系のもとで最も点数の高い結果を得ることを主眼としていた。しかし、前にも述べたように、系統樹解析の観点からすると、評価値体系は配列の系統関係の推定により変わり得るものである。また、立体構造解析の観点からみて良いアライメントは、系統樹解析の観点からみて良いアライメントとは必ずしも一致しないという議論もある 44)。こうしたことから、マルチプルアライメントの問題の解法には、計算機による完全自動化を求めるよりも、生物学者が所望のマルチプルアライメントを簡単に作成できる編集ツールを求めた方が近道だとする動きもある 45)。

ここでは、反復改善法などでなされたマルチプルアライメントの結果を、評価値には容易に帰することができない要素を考慮しながら計算機により修正、精緻化する手法を解説する。

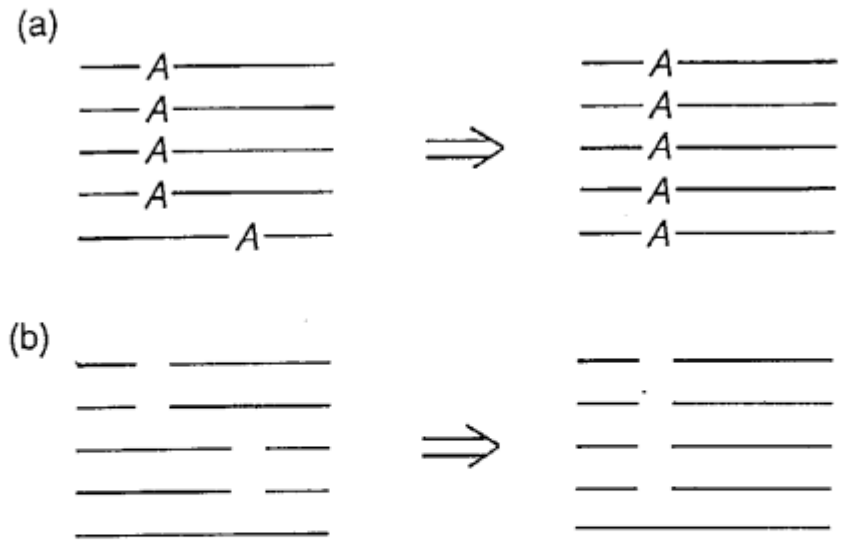


図 25: アライメントを向上させる操作の例

5.1 ルールベース法

熟練した専門家の知識を計算機に盛り込むのはエキスパートシステムの開発でよく行われる。マルチプルアライメントの精緻化にもエキスパートシステムの開発手法、知識情報処理の技術が利用できる 46)。類似性の低い配列部分にはギャップをむやみに入れるのをやめようとか、モチーフらしきものがみつかったら、すべての配列に同じものがないかを調べるとかを「もし～ならば～する」という形のルールに規則化し、計算機に入れるのである。計算機システムは、そうして表現されたルール群をルールベースとして管理し、現在のアライメントの状態から判断される最良のルールを検索して、適用するのである。

ルールの表現には、prolog や LISP などの高級言語、エキスパートシステム構築ツールを使用するとよいだろう。広沢ら 47) は prolog を使用して、コンセンサスパターンを抽出するルール（図２５(a)）、ギャップをグループ化するルール（図２５(b)）、配列を分類するルールなどを実装している。さらにモチーフ辞書も装備し、あるモチーフが検出されたら、関連のモチーフを探す手順もルール化している。

生物学者の知識を計算機に盛り込むアプローチは、端緒についたばかりで、研究課題も多い。しかし、これが実現されて始めてマルチプルアライメントの自動化が実現できると考えられる 48)。

ルールベース法の最大の研究課題は、ルールの優先づけである。通常、ひとつのアライメントの状態に適用できるルールは複数ある。ルールの数が増えるほど、この適用可能なルール数もどんどん増えてくる。ルールの優先度は、なかなか前もって設定できるものではない。それは、適用するアライメントの状態によって変わってくるうえ、適用してみるまで判断がつかないこともしばしばである。そうなると、やはり何か評価値のようなもので、アライメント状態を評価する基準を設けたりしたくなる。ルールベース法と評価値システムを統合した枠組が欲しくなるのである。こうした枠組として注目されるのが、次に解説するシミュレーテッドアニーリング法である。

5.2 シミュレーテッドアニーリング法

シミュレーテッドアニーリング法（以下 SA と略す）は、計算機科学の分野でしばしば使われる最適化アルゴリズム 49) であり、マルチプルアライメントにも応用できる。全体の評価値（とくにエネルギーと呼ばれる）を徐々に改善していくという点で反復改善法と似ているが、

```
begin
  Xo := 初期状態;
  {Tn}n=0,...,N-1 := 温度スケジュール;
  for n :=0 to N-1 do
  begin
      X'n := 変形操作を施したXn;
      ΔE := E(X'n) - E(Xn);
      if ΔE<0 then
         Xn+1 := X'n;
      else
         if exp(-ΔE/Tn) ≧ 区間[0,1]の乱数 then
            Xn+1 := X'n;
         else
            Xn+1 := Xn;
  end;
  解出力 Xn;
end;
```

図 26: シミュレーデッドアニーリングのアルゴリズム

DP を使わないので、DP では扱えないような複雑な評価値を入れることができる利点がある。そのうえ、解の探索にあたっては、変形操作（オペレーションと呼ばれる）にルール形式のものも装備できるので、SA は柔軟性のある枠組といえる。しかしその反面、反復改善法以上に、収束にかかる時間が大きいので、小さなマルチプルアライメントや、他の手法で得られた結果の精緻化に使われる。

図２６にSA のアルゴリズムを示したが、その基本的な考え方は次の通りである。SA では、最適化問題を解くことを、オペレーションを次々と適用しながら、エネルギー（評価値）が最良な状態を探し当てることとみなしている。探索にあたっては、最良探索のように画一的に探すのではなく、温度パラメータTに従いながら、探索の幅を調節している。つまり、Tの値が高いときには、エネルギーの悪化する（大きくなる）方向の探索をも、ある程度（確率的に）許し、Tの値が低いときには、エネルギーの向上する（小さくなる）方向の探索だけに制限していく。Tを高温から低温へと徐々に低下させれば、十分な時間ののちに最適解を求めることが理論的に可能である。現実には、処理時間が限られているので、その時間内に準最適な解を求めることとなる。一般的な温度低下法は、設定した最高温（解がランダム状態になる温度）から始めて、一定回数の変形を繰返したならば温度を９割程度に等比的に下げることを、エネルギー値が収束するまで行うものである。とくに、アライメントの精緻化などの、解の修正に使う場合には、中低温（現在の解の状態をそれほど崩さない温度）から始める。温度設定に柔軟に対応できる手法に温度並列 SA 50) がある。

SA を用いてマルチプルアライメントを解くには、いくつかの定式化が考えられる 51,52)。SA の定式化とは、エネルギーとオペレーションとを定義することと言いかえてもよい。エネルギーとしては、いろいろ複雑な評価値を使えるが、当然、複雑な評価値を入れるとそれだけ計算時間が大きくなる。3.4 と同様の Dayhoff マトリックスによる評価値（SA ではエネルギーは小さい程よいとする慣例から符号を反転して使う）を基盤にして、配列による重み付け、コ

ンセンサスへの報償、類似性の低い領域のギャップコストの低減など、いろいろ評価を工夫してみるとよい。

(a)

	-	-	-	-	-	-	H	E	K	L	L	H	P	G	I	Q	K	T
	-	-	-	-	-	-	H	Q	-	L	T	H	L	S	F	S	K	W
	L	S	P	A	E	L	H	S	-	F	T	H	C	G	Q	T	A	L
	-	E	A	K	D	L	H	T	-	A	L	H	I	G	P	R	A	L
	-	E	A	T	Q	A	H	T	-	L	H	H	L	N	A	H	T	L
	-	S	A	Q	E	S	H	A	-	L	H	H	Q	N	A	A	A	L
A	0	0	3	1	0	1	0	1	0	1	0	0	0	0	2	1	3	0
C	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
D	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	2	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
H	0	0	0	0	0	0	6	0	0	0	2	6	0	0	0	1	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
K	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	2	0
L	1	0	0	0	0	2	0	0	0	4	2	0	2	0	0	0	0	4
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
N	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
P	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
Q	0	0	0	1	1	0	0	1	0	0	0	0	1	0	1	1	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
S	0	2	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0
T	0	0	0	1	0	0	0	2	0	0	2	0	0	0	0	1	1	1
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	5	2	2	2	2	2	0	0	5	0	0	0	0	0	0	0	0	0

(b)

A	-6	-2	-1	-2	-2	-2	-1	0	-6	-1	0	-1	-1	0	0	0	-1
C	-6	-4	-3	-4	-5	-4	-3	-2	-6	-5	-3	-3	-1	-2	-3	-2	-3	-5
D	-6	-1	-2	-2	0	-3	1	0	-5	-3	-1	1	-2	1	-1	0	0	-3
E	-6	-1	-2	-2	0	-3	1	1	-5	-2	0	1	-2	0	-1	0	0	-2
F	-5	-5	-5	-5	-5	-2	-2	-3	-6	2	-1	-2	-1	-4	-1	-3	-4	0
G	-6	-2	-2	-2	-2	-3	-2	0	-6	-3	-2	-2	-2	2	-1	0	0	-3
H	-6	-2	-2	-2	-1	-3	6	0	-5	-1	1	6	-1	0	0	1	0	-1
I	-5	-3	-3	-3	-3	-2	-2	-1	-6	1	0	-2	0	-2	0	-1	-1	1
K	-6	-2	-3	-1	-2	-3	0	0	-5	-3	-1	0	-2	0	-1	0	1	-2
L	-4	-4	-3	-3	-4	-1	-2	-2	-6	4	0	-2	0	-3	0	-2	-2	4
M	-5	-3	-3	-2	-3	-1	-2	-1	-5	2	0	-2	0	-2	0	-1	0	3
N	-6	-1	-2	-2	-1	-3	2	0	-5	-2	0	2	-2	0	-1	0	0	-2
P	-6	-2	0	-2	-2	-3	0	0	-6	-2	-1	0	0	0	0	0	0	-2
Q	-6	-2	-2	-1	0	-3	3	0	-5	-2	0	3	-1	0	0	1	0	-1
R	-6	-2	-3	-2	-2	-3	2	0	-5	-3	0	2	-1	-1	-1	1	0	-2
S	-6	-1	-1	-2	-2	-2	-1	0	-5	-2	-1	-1	-1	1	0	0	0	-2
T	-6	-2	-1	-1	-2	-2	-1	1	-5	-1	0	-1	-1	0	0	0	1	-1
V	-5	-3	-2	-3	-3	-1	-2	0	-6	1	0	-2	0	-1	0	-1	0	1
W	-6	-5	-6	-5	-6	-4	-3	-5	-6	-2	-3	-3	-4	-5	-4	-3	-4	-2
Y	-6	-4	-4	-4	-5	-3	0	-3	-6	0	-1	0	-2	-3	-1	-2	-3	-1
-	-1	-4	-4	-4	-4	-4	-7	-7	-1	-7	-7	-7	-7	-7	-7	-7	-7	-7
JN	0	-2	1	-4	-4	2	-1	-2	-3	4	0	-1	2	0	1	-3	-4	3

図 27: プロファイル処理の例

オペレーションの定義では、まずギャップの扱い方を決める必要がある。ギャップの生成・消滅オペレーションを導入してもよいが、あらかじめ必要なギャップの個数が推測できるならば、配列の頭部や尾部に十分な数のギャップを付加して、そこから配列内部への移動オペレーションを定義するのも便利である。また、マルチプルアライメントにはギャップが固まりで入りやすいことから、ギャップを長方形の固まりで動かすようにオペレーションを拡張すると効果的である52)。さらに、揃い始めた文字を強制的に揃えてみるオペレーション（図２５(a)）、互い違いになっているギャップを縦に揃えてみるオペレーション（図２５(b)）など、前節で

ルールとして表現したものをそのまま利用できる。SA では、こうしたオペレーションの適用を確率的に行い、適用した結果のエネルギーの向上を評価し、温度パラメータに照らして採択か不採択かを判断する。その結果、定式化に従った特徴あるマルチプルアライメントが得られる。

6 プロファイル

マルチプルアライメントの結果を端的に表すのに、先に述べたコンセンサス配列があるが、類似性の低い配列のマルチプルアライメントには、なかなかコンセンサスが現れない。そうしたマルチプルアライメントの結果は、プロファイル 53) をとると、データベース検索などのその後の解析に有用である。最近では、マルチプルアライメントの結果を文法的に表現しようとする試みもなされている 54)。本章で示す隠れマルコフモデルもそのひとつと言えよう。今後、マルチプルアライメントの利用という観点に立った、アライメント結果の効率良い表現法が求められる。

6.1 プロファイル解析

プロファイルとは、マルチプルアライメントの結果において、各カラムの特徴を、何らかの基準で多次元の尺度に展開したものである。図２７には、図１のモチーフ周辺のプロファイルを示した。最も簡単には、図の (a) にあるように、各カラムにおいてアミノ酸の出現頻度をとる。これにより、各カラムのコンセンサスの度合がわかる。用途によっては、出現頻度を規格化したり、その対数をとったりする。さらに (b) のプロファイルでは、(a) に Dayhoff マトリックスを掛けている。ギャップコストには－７を与えており、配列の本数で平均化している。

Dayhoff マトリックスを掛けたプロファイルは、類似性検索に役立つ 53)。つまり、プロファイルをもとに、配列データベースなどの多数の配列のなかから、マルチプルアライメントと類似性の高い（部分）配列を見つけ出すのである。１本ずつ類似性検索したのでは見い出せない類似配列が、配列群のマルチプルアライメントを行い、それのプロファイルを用いた類似性検索により、見い出せることがよくある。検索アルゴリズムには、プロファイルと比較される配列間の DP を用いる。アミノ酸と各カラムの類似性度合は、プロファイルの該当アミノ酸の要素を参照すればわかるので、DP におけるマッチング演算は、グループ間 DP に比べ極めて容易である。

最近では、プロファイルに各カラムの内外性（疎水性）尺度、二次構造 55,56) の尺度とかを含めて、マルチプルアライメントからの蛋白質立体構造予測や、機能予測に利用する試みもなされている。図２７ (b) の最下行の JN の値は、内外性指標のひとつである Janin 57) の指標でプロファイルをとったものである。この指標では、疎水性で蛋白質の内側に入りやすいアミノ酸は大きな値、親水性で蛋白質の外側に出やすいアミノ酸は小さな値（負値）になっている。だから、この指標のプロファイルは、配列のその部分が蛋白質の内側か外側かの推測に使用でき立体構造の予測に役立つ 58)。

6.2 隠れマルコフモデル

プロファイルは、通常マルチプルアライメントした結果から求めるが、プロファイルを直接求めることが、隠れマルコフモデル (Hidden Markov Model) を適用することで可能である 59)。もちろん、そのプロファイルからマルチプルアライメントを逆に構成することもできる。隠れマルコフモデルは、文法を確率モデルとして表現するもので、連続音声認識の分野でよく知られた手法である 60)。最近、遺伝子情報処理の分野にも応用され始めてきた 61)。

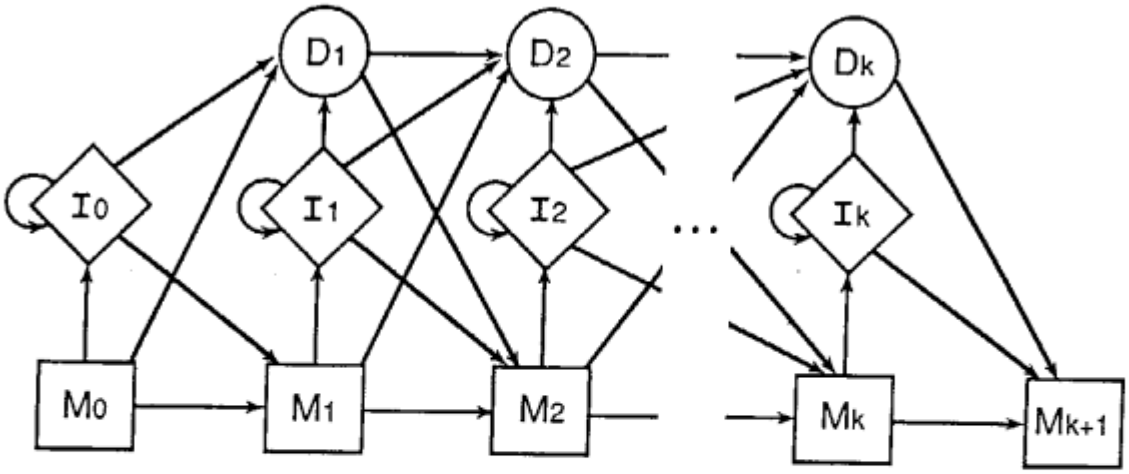


図 28: プロファイルを作る隠れマルコフモデル

隠れマルコフモデルで、プロファイル（あるいはマルチプルアライメント）を求めるには、図２８に示すようなネットワーク（マルコフ連鎖）を作成する。あらかじめ出来あがるプロファイルのカラム数（ｋ個）を推定し、それに見合うノード数を用意する。図２７(a) の形式のプロファイルを求めるのであれば、各Mノードには該当カラムにおけるアミノ酸の頻度分布を形成する（初期状態では一様分布）。そこに図の左からプロファイルを求めたい配列群を次々と投入する。投入された配列は、先頭のアミノ酸から、プロファイルのカラムとマッチする確率に従いMノードへ遷移し、アミノ酸をスキップする（配列に挿入 (insertion) があったと考える）確率に従いＩノードへ遷移し、プロファイルのカラムをスキップする（配列に欠失 (deletion) があったと考える）確率に従いＤノードへと遷移する。各配列について、最適遷移過程を DP と同様なアルゴリズム (Viterbi algorithm) で求めたならば、その情報から頻度分布、遷移確率などのパラメータを更新する。この操作をパラメータの更新がなくなるまで繰り返せば、各Mノードに各カラムのプロファイルが完成する。最後に、もう一度、配列をネットワークに通して、各配列の各アミノ酸がどのカラムに対応するかを調べれば、マルチプルアライメントを作成できる。

このモデル化で実行している内容は、１本限定の反復改善法を行っているのと大差ない62)。だから、アライメント手法として隠れマルコフモデルが、とくに秀でているわけではないだろう。しかし、プロファイルに基づくデータベース検索の際など、隠れマルコフモデルを使った表現が有効な場合があり、今後注目に値する手法と考えられる。

参考文献

1) Rhodes,D. and Klug,A.：ジンクフィンガーによる遺伝子の発現制御，日経サイエンス Vol.23 No.4, 96-106, 1993.

2) 木村資生：分子進化の中立説，紀伊国屋書店， 1986.

3) 星田昌紀，石川幹人：進化の謎を解く鍵ーアミノ酸配列解析， bit Vol.25 No.1, 51-60, 1993.

4) Pearson,W.G. and Lipman,D.J., Improved Tools for Biological Sequence Comparison, Proc. Natl. Acad. Sci. 85, 2444-2448, 1988.

5) 宮田隆，藤博幸，林田秀宜：コンピューターによる逆転写酵素遺伝子の探査，
日経サイエンス Vol.16 No.2，86-97，1986.

6) 根井正利：分子進化遺伝学，培風館，1990.

7) 五条堀孝ほか：分子進化実験法，新生化学実験講座第１６巻，日本生化学会編，
東京化学同人，1993.

8) Needleman,S.B. and Wunsch,C.D., A general method applicable to the search for similarities in the amino acid sequences of two proteins, J. Mol. Biol. 48, 443-453, 1970.

9) Smith,T.F. and Waterman,M.F., Identification of common molecular subsequences, J. Mol. Biol. 147, 195-197, 1981.

10) Gotoh,O., An improved algorithm for matching biological sequences, J. Mol. Biol. 162, 705-708, 1982.

11) 後藤修：核酸・蛋白質一次構造の計算機による解析，日本物理学会誌
Vol.38 No.6，477-480，1983.

12) Dayhoff,M.O., Schwartz,R.M. and Orcutt B.C., Atlas of Protein Sequence and Structure Vol.5 No.3, 345-352, Nat. Biomed. Res. Found., Washington DC, 1978.

13) Jones,D.T., Taylor,W.R. and Thornton,J.M., The rapid generation of mutation data matrices from protein sequences, Comput. Applic. Biosci. 8, 275-282, 1992.

14) Gonnet,G.H., Cohen,M.A. and Benner,S.A., Exhaustive Matching of the Entire Protein Sequence Database, Science 256, 1443-1445, 1992.

15) Fitch,W.M. and Smith,T.F., Optimal sequence alignments, Proc. Natl. Acad. Sci. 80, 1382-1386, 1983.

16) 宮田隆，林田秀宜，菊野玲子，安永照雄：コンピューターによる遺伝子
のホモロジー解析，続生化学実験講座第１巻遺伝子研究法Ｉ，p.381-423，
日本生化学会編，東京化学同人，1986.

17) Goad,W.B. and Kanehisa,M.I., Pattern recognition in nucleic acid sequences. I. A general method for finding local homologies and symmetries, Nucleic Acids Res. 10, 247-263, 1982.

18) Murata,M. Richardson,J.S. and Sussman,J.L., Simultenious comparison of three protein sequences, Proc. Natl. Acad. Sci. 82, 3073-3077, 1985.

19) 戸谷智之，星田昌紀，石川幹人，新田克己：並列３次元ダイナミック
プログラミング法による蛋白質の配列解析，情報処理学会第５回プログラミング
研究会，1991.

20) Altschul,S.F., Gap Costs for Multiple Sequence Alignment, J. Theor. Biol. 138, 297-309, 1989.

21) Carrillo,H. and Lipman,D., The multiple sequence alignment problem in biology, SIAM J. Appl. Math. 48, 1073-1082, 1988.

22) Vingron,M. and Argos,P., A fast and sensitive multiple sequence alignment algorithm, Comput. Applic. Biosci. 5, 115-121, 1989.

23) Bacon,D.J. and Anderson,W.F., Multiple Sequence Alignment, J. Mol. Biol. 191, 153-161, 1986.

24) Alexandrov,N.N., Local multiple alignment by consensus matrix, Comput. Applic. Biosci. 8, 339-345, 1992.

25) Taylor,W.R., Multiple sequence alignment by a pairwise algorithm, Comput. Applic. Biosci. 3, 81-87, 1987.

26) Barton,J.G. and Sternberg,M.J.E., A Strategy for Rapid Multiple Alignment of Protein Sequences, J. Mol. Biol. 198, 327-337, 1987.

27) Gotoh,O., Optimal Alignment between Groups of Sequences and its Application to Multiple Sequence Alignment, Comput. Applic. Biosci. 9, 1993.

28) Sneath,P.H.A and Sokal,R.R., Numerical Taxonomy, Freeman and Company, 1973.

29) Feng,D.-F. and Doolittle,R.F., Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees, J. Mol. Evol. 25, 351-360, 1987.

30) Higgins,D.G., Bleasby,A.J. and Fuchs,R., CLUSTAL V: improved software for multiple sequence alignment, Comput. Applic. Biosci. 8, 189-191, 1992.

31) Altschul,S.F. and Lipman,D.J., Trees, Stars, and Multiple Biological Sequence Alignment, SIAM J. Appl. Math. 49, 197-209, 1989.

32) Hirosawa,M., Hoshida,M., Ishikawa,M. and Toya,T., MASCOT: multiple alignment system for protein sequence based on three-way dynamic programming, Comput. Applic. Biosci. 9, 161-167, 1993.

33) Sankoff,D., Minimal mutation trees of sequences, SIAM J. Appl. Math. 28, 35-42, 1975.

34) Hogeweg,P. and Hesper,B., The Alignment of Sets of Sequences and the Construction of Phyletic Trees: An Integrated Method, J. Mol. Evol. 20, 175-186.

35) Hein,J., Unified Approach to Alignment and Phylogenies, Methods in Enzymology 183, 626-644, Academic Press, 1990.

36) Corpet,F., Multiple sequence alignment with hierarchical clustering, Nucleic Acids Res. 16, 10881-10890, 1988.

37) Kawai,M. Kishino,A. and Naito,K., Rapid Analysis Methodology for Gene Sequences Using a Parallel Processor, FUJITSU Scientific and Technical Journal 27, 270-277, 1991.

38) Berger,M.P. and Munson,P.J., A novel randomized iterative strategy for aligning multiple protein sequences, Comput. Applic. Biosci. 7, 479-484, 1991.

39) Ishikawa,M., Hoshida,M., Hirosawa,M., Toya,T., Onizuka,K. and Nitta,K., Protein Sequence Analysis by Parallel Inference Machine, Proc. Fifth Genaration Computer Systems '92, 294-299, 1992.

40) 星田昌紀, 石川幹人, 広沢誠, 戸谷智之, 十時泰：並列反復改善法による タンパク質配列のアライメント, 情報処理学会第２７回情報学基礎研究会, 13-24, 1992.

41) Goldberg,D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Co., 1989.

42) Konagaya,A. and Kondou,H., Stochastic Motif Extraction using a Genetic Algorithm with MDL Principle, Proc. 26th Hawaii Int'l. Conf. System Sciences Vol.1, 746-755, 1993.

43) Ishikawa,M., Toya,T., Totoki,Y. and Konagaya,A., Parallel Iterative Aligner with Genetic Algorithm, AI and Genome Workshop in IJCAI, 1993.

44) Heijne,G., Sequence Analysis in Molecular Biology, Academic Press, 1987.

45) Schuler,G.D., Altschul,S.F. and Lipman,D.J., A Workbench for Multiple Alignment Construction and Analysis, PROTEINS 9, 180-190, 1991.

46) 金久實, 新田克己, 小長谷明彦, 田中秀俊：知識情報処理技術とヒトゲノム 計画, 人工知能学会誌 6, 630-640, 1991.

47) Hirosawa,M., Hoshida,M. and Ishikawa,M., Protein Multiple Sequence Alignment using Knowledge, Proc. 26th Hawaii Int'l. Conf. System Sciences Vol.1, 803-812, 1993.

48) 隈啓一：アライメントの最適化と自動化への問題点, 日本生物物理学会誌 Vol.32 No.3, 62-64, 1992.

49) Kirkpatrick,S., Gelatt,C.D. and Vecci,M.P., Optimization by Simulated Annealing, Science 220, 4598, 1983.

50) 木村宏一, 瀧和男：時間的一様な並列アニーリングアルゴリズム, 電子情報 通信学会研究会 NC90-1, 1-8, 1990.

51) Ohya,M., Miyazaki,S. and Ogata,K., On Multiple Alignment of Genome Sequences, IEICE Trans. Commun. E75-B, 453-457, 1992.

52) Ishikawa,M., Toya,T., Hoshida,M., Nitta,K., Ogiwara,A. and Kanehisa,M., Multiple sequence alignment by parallel simulated annealing, Comput.

Applic. Biosci. 9, 1993.

53) Gribskov,M., McLachlan,A.D. and Eisenberg,D., Profile analysis: Detection of distantly related proteins, Proc. Natl. Acad. Sci. 84, 4355-4358, 1987.

54) Helgesen,C. and Sibbald,P.R., PALM: A Pattern Language for Molecular Biology, Proc. 1st Int'l. Conf. Intelligent Systems for Molecular Biol., 1993.

55) 西川建：タンパク質の二次構造予測，情報処理 31，887-896，1990.

56) 陶山明，江口至洋，上田裕三，和田昭允：コンピュータによる核酸塩基配列の情報解析，蛋白質 核酸 酵素 Vol.28 No.10，1165-1186，1983.

57) Janin,J., Surface and inside volumes in globular proteins. Nature 277, 491-492, 1979.

58) 谷村隆次，梅山秀明：エキスパートシステムを用いたタンパク質の立体構造の推定と病気治療への応用，情報処理 31，897-903，1990.

59) Haussler,D., Krogh,A., Mian,I.S. and Sjoelander,K., Protein Modeling using Hidden Markov Models: Analysis of Globins, Proc. 26th Hawaii Int'l. Conf. System Sciences Vol.1, 792-802, 1993.

60) 中川聖一：確率モデルによる音声認識，社団法人電子情報通信学会編，コロナ社，1988.

61) Asai,K., Hayamizu,S. and Onizuka,K., HMM with Protein Structure Grammar, Proc. 26th Hawaii Int'l. Conf. System Sciences Vol.1, 783-791, 1993.

62) Tanaka,H. Asai,K., Ishikawa,M. and Konagaya,A., Hidden Markov Models and Iterative Aligners: Study of their Equivalence and Possibilities, Proc. 1st Int'l. Conf. Intelligent Systems for Molecular Biol., 1993.