ICOT Technical Memorandom: TM-1069

TM-1069

祖先配列を用いた マルチプルアライメント

廣澤 誠

July, 1991

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

1　はじめに

蛋白質は、アミノ酸が鎖状に連なった物質である。蛋白質の構成要素として使われているアミノ酸には２０種類あり、それぞれに特定のアルファベットが割り当てられている。蛋白質の機能・構造は、アミノ酸がどのような順序でつながれているかに依存している。つまり、蛋白質をアミノ酸配列で表現することができる。現在、アミノ酸配列が決定されている蛋白質は１万種類ある。しかし、この内、機能・構造がの両方が判明している蛋白質は、２００種類程度である。

マルチプル・アライメントとは、機能・構造が未知である蛋白質の機能・構造を、注目している蛋白質の配列と類似している配列を持つ蛋白質の機能・構造をもとに推測するための技術である。

配列が与えられた時、高速に、マルチプル・アライメントを求める方式が必要とされている。トーナメント方式によるマルチプル・アライメント（以下、トーナメント方式と呼ぶ）は、進化系統樹の考え方を用いてマルチプル・アライメントを高速に求める方式である。

トーナメント方式では、現存する生物種（以下、生物と呼ぶ）が持つ特定の蛋白質を意味するアミノ酸配列が複数本ある場合に、これらの祖先として存在した生物の対応するアミノ酸配列（以下、祖先配列）を求めていく。そして、この祖先配列を用いてマルチプル・アライメントを求める。この過程において同時に進化系統樹を構築することもできる。

トーナメント方式では配列間の類似度を求めるために $(n-1)^2$ 個（配列の本数をｎ本とする）の２次元アライメントを行う必要がある。ここでは、ＤＰ [Needleman 78] によるアライメントを採用している。これらのＤＰを行う計算量は、全体の計算量の大部分を占める。トーナメント方式では、マルチＰＳＩを用いてこれらのＤＰを並列に計算することにより高速にマルチプル・アライメントを求めることができる。

2　マルチプル・アライメントとは？

マルチプル・アライメントとは、複数の配列が与えられた時に、類似する文字を同じ列に並べる技術である。ここで各文字はアミノ酸に対応している。例えば、Ｄはアスパラギン酸、Ｙはチロシン、Ｆはフェニールアラニンを意味している。

類似度としては Dayhoff のマトリックス（[Dayhoff 78]）を用いている。{DIYA、DFA、DFAT} をアライメントした例を下に示す。類似した文字を並べるためにギャップ（ー で表されている）が用いられている。

```
DIYA-
D-FA-
D-FAT
```

Ｄの列と、Ａ（アラニン）の列の他にＹ（１行目）とＦ（２、３行目）からなる列があるのは、チロシン、フェニールアラニンが類似しているからである。また、Ｉ（１行目）とギャップ（２、３行目）

からなる列があるのは、I（イソロイシン）に対応するアミノ酸が2、3行目には存在しないからである。

3　トーナメント方式の基本概念

トーナメント方式は以下のことを目的するアライメント方式である。

- マルチプル・アライメント を求める。
- 進化木を求める。
- 祖先の蛋白質配列を求める。

トーナメント方式は、祖先の配列を進化を逆トレースすることによりマルチプル・アライメントを求めていく。図1、図2 を用いて基本概念を説明する。

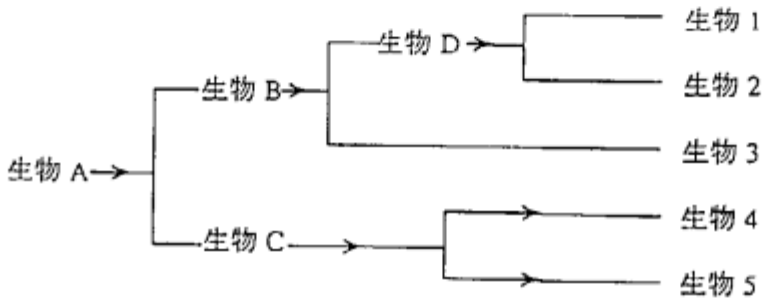

図1 生物の系統樹の例

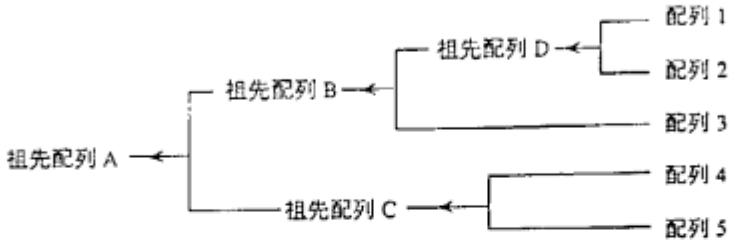

図2 祖先配列を求める

図1 は以下のことを示している。現存する生物 1 ～ 5 は、配列 A を持つ共通祖先 生物 A から、分岐した生物である。生物 A は過去に存在した生物であり現存しない。 まず、生物 A が 生物 B と生物 C に分岐した。それぞれ、祖先配列 B と 祖先配列 C を 持つ（祖先配列の正確な定義は次の節で行う）。生物 B も生物 C も現存しない。次に、生物 B が 生物 D と生物 3 に分岐した。それぞれ、祖先配列 D と 配列 3 を 持つ。生物 D も現存しない。その次に、生物 C が 生物 4 と生物 5 に分岐し

た。それぞれ、配列 4 と 配列 5 を持つ。最後に、生物 C が 生物 4 と生物 5 に分岐し、それぞれ、配列 4 と 配列 5 を持つ。

この時、現存しない生物の祖先配列を以下に示す順番で求めていく。この時、図1 に示されている進化系統樹の形を仮定しない。前にも述べたように、この進化系統樹は祖先配列を求めていく過程において順々に決まっていく。。祖先配列を求めた後に、祖先配列に含まれる情報を利用してマルチプル・アライメントを求める。これについては、次節において説明する。

祖先配列を求める順番について述べる。この時に、以下の仮定をする。この仮定は、進化速度が一定である場合などにおいて適切な仮定である（[石川統 85]）。

- 複数の生物からなるグループがある時に、一番最近に分岐した二つの生物はこのグループの中で一番類似している二つの生物である。

類似度の判定法については、次節において述べる。

上記の仮定のもとで、図1 に示した場合に祖先配列を求める手順は以下の様になる。

まず、生物 1 ～ 5 の中で、最も類似している二つの生物を求める。結果として、 生物 1 と 生物 2 となる。この直接の祖先が、図1 の生物 D に対応する。次節に述べる方法により祖先配列 D を求める。これは、図2 の祖先配列 D である。次に、生物 D 、生物 3 ～ 5 の中で最も類似している最も類似している生物を求める。結果は生物 4 と 生物 5 である。これは、生物 C に対応している。そして、祖先配列 C を求める。その次に、生物 C 、生物 3 、生物 D の中で最も類似している生物を求める。生物 C と 生物 3 が結果となる。これが、生物 B に対応する。そして祖先配列 B を求める。最後に、生物 B と 生物 C が残る。この二つの生物の祖先が生物 A に対応する。これは同時に生物 1 ～ 5 の共通祖先生物である。そして、祖先配列 A を求める。

4　トーナメント方式の詳細

第2章 ではトーナメント方式の基本概念を述べたが、ここではその詳細について述べる。以下、まず、祖先配列の定義と、二つの配列から祖先配列を求める方式を求める方式を説明してから、祖先配列を求める順番について述べる。その後に、祖先配列からマルチプル・アライメントを求める方法について説明する。

4.1　祖先配列の求め方

最も類似しているものとして選ばれた二つの配列から祖先配列を求める方法について説明する。

(1) 二つの配列のアライメント

まず、二つの配列のアライメントを求める。アライメントを求める方法としては、多くの方法が考えられるが、今回は、二つの配列の二次元ＤＰを採用した。

(2) 祖先配列の定義

二つの配列の祖先配列を、この二つの生物（この生物を子孫、そして各配列を子孫配列と呼ぶことにする）の直接の祖先の生物が持っていた配列であると定義した。具体的には、祖先配列は、子孫配列をアライメントした時、この各コラムにある二つの文字に対して求めた祖先文字の並びとなる（祖先文字の定義に付いては後述する）。例えば、〔DIYA、D-FA〕の祖先配列は D と D の祖先文字、I と - の祖先文字、Y と F の祖先文字、A と A の祖先文字を求めることにより求めることができる。以下、子孫配列を構成する文字を子孫文字と呼ぶことにする。

祖先文字は、二つの子孫文字が祖先配列においてどのような文字であったかを示すものである。祖先文字としては、、２０個のアミノ酸を表す文字の他に１９個の文字 $a \sim s$ を用いる（図3 参照）。この階層構造は Dayhoff のマトリックス（[Dayhoff 78]）に基づいて求めたものである。Dayhoff のマトリックスはアミノ酸の置換確率に基づいた類似度を表す。複数のアミノ酸配列からモチーフを求める方式ではアミノ酸の性質に基づいた階層構造（[Smith 86]）が用いられているが、これは祖先配列を求めるためには適切ではない。

$a \sim s$ は、複数のアミノ酸を要素とするクラスを表す文字である。類似しているアミノ酸は低い階層で１つのクラスにまとめられる。これとは反対に、類似していないアミノ酸は、高い階層において１つのアミノ酸にまとめられる。

クラスは、子孫の二つの文字が異なる場合に、祖先文字がこの二つのどちらかであるという任意性があることを示すために用いる文字である。例えば、二つの文字が F と Y のどちらかであるという任意性があるということを a を用いて表す。

4.2 祖先文字の求め方

前に述べたように祖先配列は祖先文字の並びであるので、二つの子孫配列の子孫文字に対応する祖先文字を求めることにより、祖先配列を求めることができる。

まず、二つの祖先文字が同じ場合は祖先文字もこれと同じである。その他の場合には、以下に説明する Gap Handling Rule, Specialization Rule, Generalization Rule を用いて祖先文字を求める。以下、一方の文字を LetterA, 他方の文字を LetterB と表す。

(1) Gap Handling Rule

LetterA がギャップある場合、LetterB を祖先文字とするルールである。これは、祖先配列のこの場所にあった LetterB が生物 B の配列 B には継承されたが、生物 A の配列 A では 欠損が起こったことを意味している。

(2) Specification Rule

これは、図3 において、LetterA が LetterB を包括している時に、LetterA の直接の子孫である LetterA1 または LettreA2 が LetterB に包括されているか、LetterB を包括している場合に、

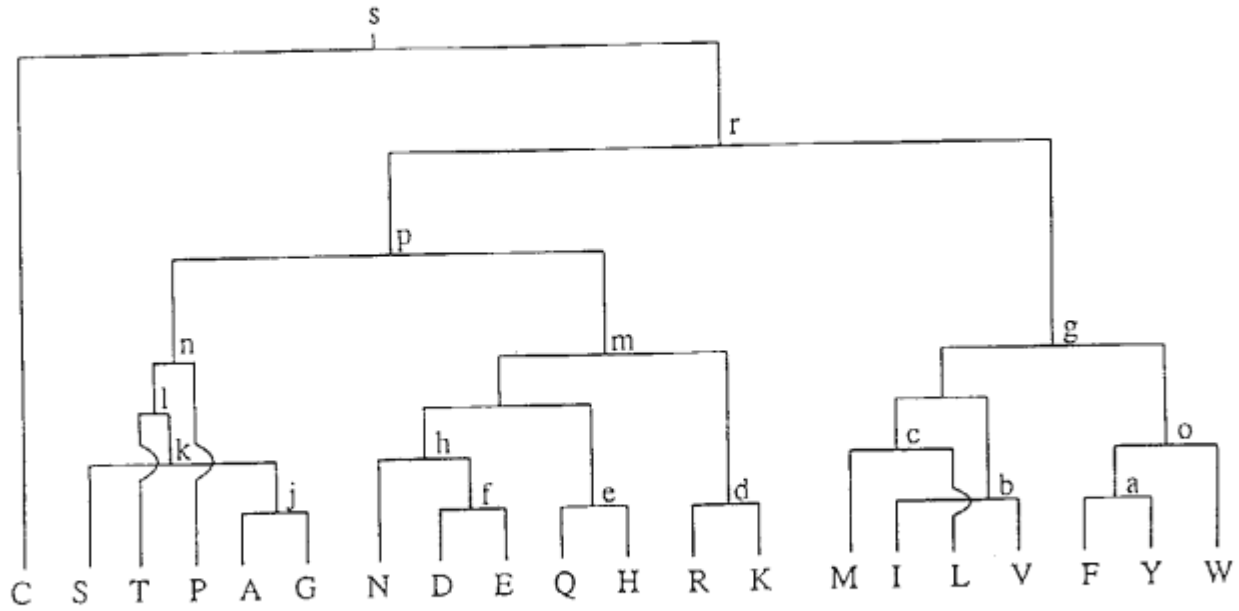

図3 祖先文字の構造

LetterB を祖先配列とするルールである。これは、とりえるアミノ酸の数が多い LetterA と、取り得るアミノ酸の数が絞られている LetterB がある時に、ある条件を満たすならば祖先文字として LetterB を選び とりえるアミノ酸の数を絞ることを意味する。

(3) Generalization Rule

これは、Specification Rule を適用できない時に、図3 において、LetterA と LetterB を包括する文字での中で、取り得るアミノ酸の数が最小の文字を祖先文字とするものである。

5 祖先配列を求める順序

祖先配列を求める順序を説明する。この順番は類似度に基づいて決める。類似度としてはＤＰのコストを採用する。祖先配列を求める順番を決めることは進化系統樹を枝の方から求めることに対応している（図2）。

ｎ個の配列の集合がある場合を考える。まず、ペアごとのＤＰを行う。このためには、ｎ（ｎ－1）／2 回のＤＰを行う必要がある。そして、類似度が最大である二つの配列の祖先配列を求める。

次に、選ばれた二つの配列を配列集合から除く、そして、求められた祖先配列を配列集合に加える。このｎ－1個の配列のＤＰのコストを比較する。この時、重複する計算は前回の計算を利用する。すなわち、実際に行うべきＤＰは祖先配列と他のｎ－2個の間のＤＰだけである。

この結果、類似度が最大である二つの配列の祖先配列を求める。祖先配列を求める時には、既に求めてある二つの配列間のアライメントを利用する。この時、

以降、次々に祖先配列を求めて行く。こうして、結果としてｎ－１個の祖先配列を求めることになる。祖先配列を求めるために必要なＤＰは、１個づつ減るので、ｉ番目の祖先配列を求める時に必要なＤＰの数は ｎ－ｉ個である。

6　祖先配列の求め方の例

祖先配列を次々に求めていく方法を DIYA, DFA, DIFT の祖先配列を求めることを例にして説明する。この中で、DIYA と DFA が最も類似しているとする。まず、DIYA と DFA のアライメントが必要であるが、これは類似度をＤＰにより求める時に計算されている。結果は {DIYA、D-FA} であるとする。Ｄ と Ｄ の祖先文字は D、Ｉ と - の祖先文字は Gap Handling Rule により I、Ｙ と Ｆ の祖先文字は Genaralization Rule により ａ 、Ａ と Ａ の祖先文字は Ａ であるので、祖先配列は、DIaA となる。その後に、この祖先配列と DIFT のアライメントを求める。結果は、{DIaA-、D-FAT} であるとする。この祖先配列は DIFAT となる。この時、Specification Rule を ａ と Ｆ から 祖先文字として Ｆ を求めるために用いた。

7　マルチプル・アライメントの求め方

マルチプル・アライメントは、祖先配列を求めた子孫配列のアライメントに置き換えていくことにより求めることができる。上記の例を用いて具体的に説明する。

DIYA, DFA, DIFT の祖先配列である、DIFAT を順次この配列の子孫配列に置き換えていく。まず、DIFAT を {DIaA-、D-FAT} に置き換える。そして、DIaA- に含まれる DIaA をこの子孫配列である {DIYA、D-FA} に置き換える。これにより、アライメントとして { {DIYA-、D-FA-} 、D-FAT} 、つまり {DIYA-、D-FA-、D-FAT} が求まる。

8　並列性

トーナメント方式では、ＤＰを並列に行うことにより台数効果をあげることができる。膨大なＤＰの計算量に比較してその他の処理量を無視することにする。そして、計算量を、１個のＤＰの計算量を単位として計ることにする。また、ｐ個のＰＥの内の１個をＤＰを割り振るＰＥとし、その他の（Ｐ－１）個のＰＥでＤＰを並列に行うとする。

まず、ｎ個の配列のマルチプル・アライメントの計算量は、最初に行うｎ個の配列間の $n(n-1)/2$ 個の ＤＰと、その後に行う $(n-1)(n-2)/2$ 個の ＤＰを合わせた $(n-1)^2$ 個のＤＰの計算量に対応する。そして、ｐ個のＰＥが全体として何サイクルのＤＰを行うかというと $\frac{(n-1)^2+\frac{n(p-1)}{2}}{p-1}$ 個である。したがって、台数効果の理論値は $\frac{p-1}{1+\frac{p-1}{2(n-1)}}$ となる 。

この式から分かるように配列数が少ない場合には、台数効果は配列数が少ない場合は低いが、配列数が多くなるとＰＥの数から１を引いた値に近づいていく。

9　結果

約６０個のアミノ酸からなる７本の蛋白質をマルチプル・アライメントした結果が図４に示す。約２５秒かかった。この配列を手でマルチプル・アライメントした九州大学の宮田研の結果（図５）（[宮田　86]）と比較すると、トーナメント方式の結果でも、宮田研の結果と同様に IKTDN というモチーフを中心とする保存部位を捕らえていることがわかる。これは、トーナメント方式が妥当なマルチプル・アライメントであることを示す証明の一つとなっている。

```
SMRV    -G-FILATRQTGEASKNVIS-HVI-HCL-A-TI-GKPHTIKTDNGPGYTGKNFQDFCQKL--QI--
MMTV    YSHFTFATARTGEATK-DVLQHLAQSF--AY--MGIPQKIKTDNAPAYVSRSIQEFLARW------
IAP     -G-VMFATTLTGE--K-A-S-YVIQHCLEAWSAWGKPR-IKTDNGPAYTSQKFRQFCRQM--DVT-
RSV     ---IV-VTQH-GRVTSVAVQHHWATAI--AV--LGRRKAIKTDNGSCFTSKSTREWLARWG--IAH
HTLV-1 ------SGAISATQKRKETSSEAISSLLQAIAHLGKPSYINTDNGPAYISQDFLNMC----TSLA-
HTLV-2 -D--TFSGAVSVSCKKKETSCETISAVLQAISLLGKPLHINYDNGPAFLSQEFQEFC----T----
BLV     -H------A-S-A-KRGLTTQTTIEGLLEAIVHLGRPKKLNTDQGANYTSKTFVRFCQQFGVSLS-
```

図 9.0.0-4 図4　トーナメント方式を用いたマルチプル・アラインメント

```
SMRV    ----GFILATPQTGE-ASKNVISHVIHCL-ATIGKPHTIKTDNGPGYTGKNFQDFCQKLQI---
MMTV    --YSHFTFATARTGE-ATKDVLQHLAQSF-AYMGIPQKIKTDNAPAYVSRSIQEFLARW-----
IAP     ----GVMFATTLTGEKASY-VIQHCLEAWSAW-GKPR-IKTDNGPAYTSQKFRQFCRQMDVT--
RSV     ------IVVTQH-GRVTSVAVQHHWATAI-AVLGRPKAIKTDNGSCFTSKSTREWLARWGIAH-
HTLV-1 ---SGAISATQK-RKETSSEAISSLLQAI-AHLGKPSYINTDNGRAYISQDFLNMCTSLA----
HTLV-2 DTFSGAVSVSCK-KKETSCETISAVLQAI-SLLGKPLHINTDNGPAFLSQEFQEFCT-------
BLV     -------HASAK-RGLTTQTTIEGLLEAI-VHLGRPKKLNTDQGANYTSKTFVRFCQQFGVSLS
```

図 9.0.0-5 図5　宮田研によるマルチプル・アラインメント

【参考文献】

[Needleman　78] Needleman,S.B. and Wunsch,C.D *"General Method Applicable to the Search fir Simularities in the Amino Acid Sequences of Two Proteins"* in *Journal fo MolecularBiology 48,* 1970, pp.443-453.

[Dayhoff　78] Dayhoff, Hunt and Hurst-Calderone *"Composition of Proteins"* in *Atlas of Protein Sequence and Structure 5:3,* Nat. Biomed. Res. Found., Washington, D. C., 1978, pp.363-373.

[石川統　85] 石川統 "分子進化", *Shouka-bou,* 1985.

[Smith　86] Smith,R and Smith,T *"Automatic generation of primary sequence paterns form sets of related protein sequences"* in *Biochemistory Vol.87* , 1990

[宮田　86] 宮田、藤、林田 "コンピューターによる逆転写酵素の探査", サイエンス １９８６年２月号.