情 報 学 基 礎 27－2
（1992. 9. 8）

並列反復改善法によるタンパク質配列のアライメント

星田 昌紀，石川 幹人，広沢 誠，戸谷 智之
（財）新世代コンピュータ技術開発機構

十時 泰
（株）情報数理研究所

タンパク質配列のマルチプルアライメントの問題は、組合せ最適化問題と捉えることができ、実用的規模の問題では、大量の計算量を必要とする。生物学者が手作業でアライメントをする場合も、大変な労力が必要であり、高品質で高速な自動アライメントシステムが望まれている。我々は、新世代コンピュータ技術開発機構（ICOT）で開発された並列推論マシンを利用し、実用規模のアライメントを、現実的な時間内に実行するシステムを構築した。本システムは（ツリーベース）並列反復改善法を用いており、典型的な従来法よりも実行時間は2倍程度になってしまうが、実用規模の問題においても、従来法より高品質なマルチプルアライメントを提供する。

Protein Sequence Alignment by Parallel Iterative Method

Masaki Hoshida, Masato Ishikawa, Makoto Hirosawa,
Tomoyuki Toya
Institute for New Generation Computer Technology (ICOT)

Yasushi Totoki
Information and Mathematical Science Laboratory, Inc.

We have studied application systems for parallel inference machine developed in ICOT. The systems analyze protein sequences and generate multiple sequence alignments. That analysis is very important in the field of genetic information processing. This paper reports two systems: a parallel iterative aligner and a tree-based parallel iterative aligner. Although the systems require twice as much execution time as a conventional method, they generate high-quality alignments for even practical-scale problems.

1 はじめに

本論文では、昨年の本研究会のゲノム特集での発表 [石川 91] にひきつづき、ICOT で開発した並列推論マシンを用いた、タンパク質のアミノ酸配列解析システムについて報告する。前回に報告した統合アライメントシステムは、その後 MASCOT[広沢 92] と命名したシステムまで整備を続けた。MASCOT は、大きな規模のマルチプルアライメントの問題にも、十分高品質の解を与えることができた。さらに今回は、MASCOT の開発で培ったノウハウを生かし、並列反復改善法という、より優れた方法を実装した、より実用レベルのアライメントシステムを構築したので、その報告を行う。

1.1 タンパク質とアミノ酸配列解析

遺伝情報は、DNA に格納されており、この DNA の情報がアミノ酸配列を指定し、アミノ酸配列がおれたたまって、タンパク質となる。タンパク質は、生体中で実際に機能する重要な物質であり、生命現象の中で中心的な役割を担っている。

タンパク質の構成要素であるアミノ酸には２０種類あり、それぞれアルファベット１文字で表される慣習になっている。アミノ酸には、大きさ、親水 - 疎水性、酸性 - 塩基性などの性質があり、どんな性質のアミノ酸がどんな順番に連なっているかで、タンパク質の構造や機能が決まってくる。タンパク質には大小さまざまなものがあり、短いものは数十個、長いものは数百個のアミノ酸が連なっている。

タンパク質の構造や機能は、実験を積み重ねて初めてわかるものとされている。事実、タンパク質の正確な構造は、まだ数百種類程度しか知られていない。一方、タンパク質のアミノ酸配列を調べる技術は、すでに確立されており、それを自動分析する機械も販売されている。現在では数万種類のタンパク質について、その配列が決定されている。この数字は近年、ますます増大している。

これらの状況から、アミノ酸配列からタンパク質に関する構造や機能などの知見を引き出す配列解析が、分子生物学での重要な技術のひとつになりつつある。

1.2 マルチプルアライメント

もっとも基本的な配列解析のひとつは、複数の配列の類似する部分を縦に揃えて並べ合わせる操作で、マルチプルアライメント (Multiple Alignment) と呼ばれる。たとえば次のようなタンパク質のアミノ酸配列が 4 本あったとする。

```
IEGLLEAIVHLGRPKKLNTDQGANYTSKTFVRFCQQFGVSL
YVIQHCLEAWSAWGKPRIKTDNGPAYTSQKFRQFCRQMDVT
VQHHWATAIAVLGRPKAIKTDNGSCFTSKSTREWLARWGIA
KNVISHVIHCLATIGKPHTIKTDNGPGYTGKNFQDFCQKLQ
```

ここで、それぞれの文字がひとつひとつのアミノ酸を表現する。アミノ酸は２０種類あり、２０種のアルファベットで識別される。これをアライメントすると、次のようになる。

```
---IEGL-LEAIVHLGRPKKLNTDQGANYTSKTFVRFCQQFGVSL
-YVIQHC-LEAWSAWGKPR-IKTDNGPAYTSQKFRQFCRQMDVT-
---VQHHWATAIAVLGRPKAIKTDNGSCFTSKSTREWLARWGIA-
KNVISHV-IHCLATIGKPHTIKTDNGPGYTGKNFQDFCQKLQ---
```

配列のところどころにギャップ "-" を入れることで、TDXG などの共通文字が同じ列に並んでいるのがわかる（X は不特定のアミノ酸１つを意味する）。TDXG のように複数の文字が、複数の配列で共通になっている文字の組を配列モチーフと呼び、タンパク質のうちの重要な部分を指し示していると判断される。この背景には、タンパク質の配列のうち重要である部分には遺伝的変異が起きにくいという進化論的考え方 [木村資生 86] がある。

一般のマルチプルアライメントでは、ひとつの列に同じ文字が揃うことは少なく、異なる文字でもそれらが表すアミノ酸の性質が似ていれば同じ列に置くことを許容して処理を行う。現在最も広く使われている類似性評価尺度は、図 1 の Dayhoff マトリックス [Dayhoff 78] である。この尺度は、当時知られていたアライメントをもれなく調べ、該当アミノ酸対の遺伝的変異が偶然に対して何如に大きいかを数値化したものである。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。本論文の、マルチプルアライメントシステムは、この評価尺度を使用している。この評価尺度は算出法が極めて妥当なものの、決められてからもう１０年以上も経過し、算出に使用したデータが古くなっている。最近、最新のデータで評価値を算出し直そうという動きがある [Jones 92]。

システイン	C	12	イオウ重合性																					
セリン	S	0	2	小型																				
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	酸性とそのアミド															
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	塩基性											
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	疎水性								
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	芳香族				
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 1: アミノ酸間の類似性評価尺度（Dayhoff マトリックス）

マルチプルアライメントされた結果は次のように利用できる。第一に、先に述べたように、重要な配列の部分であるモチーフを見出せる。モチーフはデータベースから新たな知見を引き出す手がかりになるなど、生物学的に有用である。第二に、マルチプルアライメントから進化系統樹を描くことができ、類似配列の各々がどのような遺伝的過程を経てきたかを推測することができる。第三に、類似した構造・機能を持つ配列をアライメントした結果から、共通の構造・機能に対応するアミノ酸の性質の並びがわかり、その構造・機能の形態を予測する助けとできる。このようにマルチプルアライメントは、遺伝子情報処理の中で重要な役割を担っている。

Dayhoff マトリックスのような評価尺度が与えられていれば、それにおいて最適なマルチプルアライメントを求めるダイナミックプログラミング法（以下 DP と呼ぶ）が知られている [Needleman 70][Waterman 81]。しかしそれは必要とする計算量が多く、配列が 3 本以上ではあまり実用的ではない。問題によって不要な部分の計算を削減しても、配列 6，7 本までが処理できる上限である [Carrillo 88]。そのため通常は、2 本ずつの比較を組み合わせてマルチプルアライメントを行う方法が試みられている [Feng 87]。最近出回っている配列解析ツールも、多くはこうした方法をとっている [Higgins 92]。なかでも最も典型的なのは、配列の類似性を前もって調べ、似ているものから順にツリー状に組み合わせるツリーベース法 [Barton 90] である。それでも精度が十分でなく、難しいところは、生物学者の勘に頼っている。すなわち、計算機で出力されるマルチプルアライメントは、経験を積んだ生物学者が行うマルチプルアライメントのレベルには、いまだに達してはいないということができる。

我々は、実用レベルの自動マルチプルアライメントを行う計算機システムの構築を目指し、研究を行っている。以下、我々の研究開発したシステムおよび今後の課題について議論する。

2 反復改善法

2.1 概要

昨年、反復改善法 [Berger 91] という、新しい視点からのアライメント手法が提案された。この方法もやはり、要素技術に DP を用いているが、それ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していくというものである。我々は反復改善法の潜在的な能力に注目し、この方法をベースに新しい方式によるアライメントシステムを構築した。この章では、まず DP について説明し、次に反復改善法についての解説を行う。さらに反復改善法の問題点について考察を行った後、次章では、我々の並列反復改善法について述べる。

2.2 ダイナミックプログラミングによるアライメント

ダイナミックプログラミング（DP）は段階的に決定を行う特徴を持つ最適化問題を解くためのアルゴリズムのひとつである。いくつもの段階で決定を行う必要があり、その各段階における決定が直前の段階の決定にのみ依存するという形式に最適化問題を定式化できるとき、DP を用いると非常に効率良く最適解（スコア最良の経路）を求めることができる。もしこの種の問題を、最初にすべての場合を尽くしてそのうちの最良のものを選ぶという解法で解いた場合、指数オーダーの計算量がかかる。しかし DP を用いると多項式オーダーで解を得ることができる。

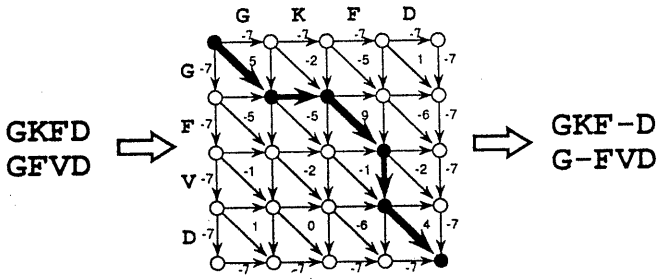


図 2: 2次元 DP による配列のアライメント

アミノ酸配列のアライメントは、この DP を用いて行うことができる。簡単のために配列2本のアライメントについて DP の概念的説明を、図2を用いて行う。たとえば、GKFD, GFVD という2つの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スコアとして割り振られる。この類似度には前述の Dayhoff マトリクスを用いている。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。ギャップコストは経験上、ギャップの長さ k に対して、a + b k のような一次式を与えるのが適当であり、DP にも通常そうしたギャップコストが実装される。ギャップコストの効率のよい実装法や、それに伴う計算量の分析も考察されている [後藤修 83]。

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スコア最良の経路を求めることに対応する。図の例では太いアークで表された経路がスコア最良となる。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。

スコア最良の経路は左上の端から右下の端に向かって（逆でも可能）各ノードに至る最適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。各段階は図2の右上がり斜め線上に存在するノード群に対応する。段階的に各ノードへ至る最適経路を求めていくと、いったん求まった部分的最適経路は、もはや変更されることがない。それゆえこの部分的最適経路を用いて次の段階の計算を行うことができる。具体的には、ある段階の各ノードの計算を行うためには、直前の段階の各ノード（2次元 DP では3つある）で求まった部分的最適経路のスコアを参照して、今求めたいノードに至るスコアをそれぞれ計算し、このうちの最良値を求めてそれをそのノードに至るスコアとすればよい。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。この操作を最後まで繰り返せば、ネットワーク全体の最適経路を求めることができる。

各段階で部分的最適経路を決定してしまうため、経路全体の組み合わせを考慮する必要はなく、ある段階から次の段階へ遷移するときの組み合わせだけを考慮すればよい。この性質があるために、全体の計算量が段階数の指数オー

ダーになることを回避できるわけである。これがDP手法の本質である。また、計算量を削減するために、最適解が得られる保証がなくなってしまうが、ネットワークのコーナー（図では左下と右上の部分）を問題に応じてカットすることもよく行われる。

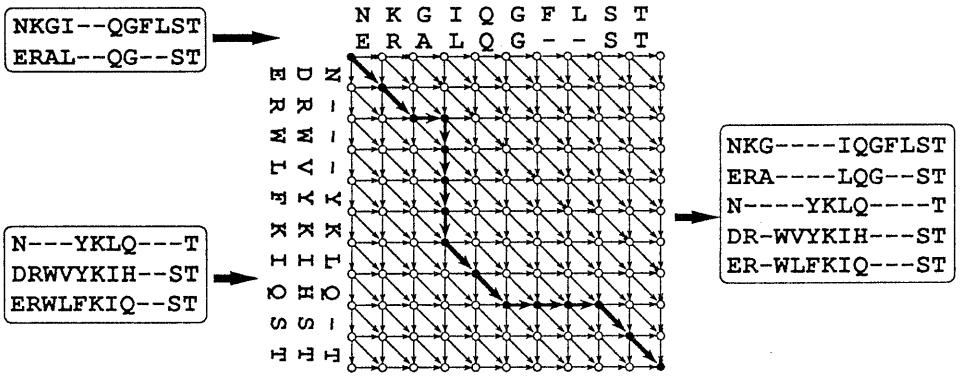


図 3: 2次元DPによる配列グループ間のアライメント

DPではまた、アライメントが済んだ配列グループ間同士のアライメントに拡張することができる。このグループ間の2次元DPを行うときは、それぞれの配列グループの中では、アミノ酸やギャップの縦方向の位置が変わらないように固定しておく。斜めのアークには、各列に存在するアミノ酸とギャップの総当たり組合わせの和の得点を与えるのが標準的である。図3では、配列グループ間の2次元DPの適用例を示している。

2.3 反復改善法とその問題点

反復改善法は、前述の配列グループ間の2次元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4）。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。例えば、一番簡単な方法としては、全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N本の配列を、ランダムに選んで2つのグループに分割する。この分割法は $2^{N-1}-1$ 通りある。分割された2つのグループ間に、2次元DPを適用する。その結果は、それぞれ、必ずひとつ前の状態の得点より改善しているか、悪くても同じ得点である。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2次元DPを適用する。この過程を1試行として、改善が行われる。この試行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。

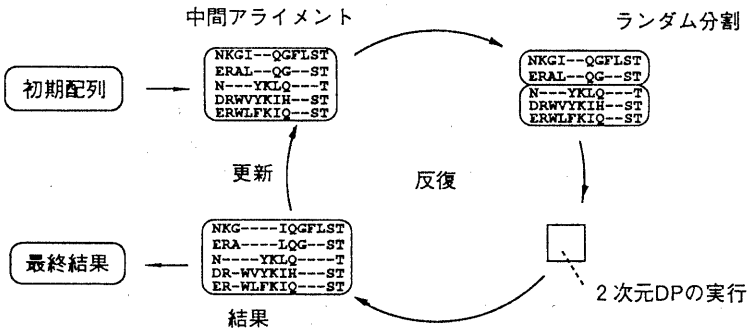


図 4: 反復改善法の手順

反復改善法は、強力な方法ではあるが、以下のような問題点も持っており、実用規模のマルチプルアライメントに適用するのが難しい。

- 問題点1：グループ間2次元DPの再試行による速度低下

配列を２つのグループに分割し２次元 DP を行う際、分割によっては、２次元 DP による改善が見られない試行も多い。試行の結果、改善が見られない場合、別の分割で再試行を行うことになる。とくに処理の後半に、この再試行が多くなる。実行時間のうちのかなりの部分が、この再試行にあてられ、全体として実行時間が長くなりがちである。

- 問題点２：配列の分割数の組合せ的爆発による速度低下

  配列を２つのグループに分割する際、ランダムに分割を行うと、K 本と N-K 本に分割される場合の数は ${}_KC_N$ である。すると、例えば２０本の配列を分割する場合、１０本と１０本とに分かれる場合が 184756 通りもあるのに対し、１本と１９本というような分割は、20 通りでほとんど行われないことになる。ところが、我々の行った実験によると、１本と１９本というような配列の本数に偏りのある分割こそがアライメントの改善には有効であり、１０本と１０本というような均等な分割がアライメントの改善に寄与することはまれである。反復改善法はこの分割の均等性（分割された２つのグループの中の配列の本数に偏りがないこと）が原因で、本数が増えた場合のアライメントの改善スピードが著しく遅くなっている。

- 問題点３：アライメント結果の初期状態依存性

  初期状態で、アミノ酸がたまたま比較的揃っているカラムが、改善の早期に大きく影響するので、得られるアライメント結果が初期状態に依存する傾向がある。すでに配列の重要な部分を同じ長さに切った問題を扱う場合には、切り揃えた初期状態において、すでに同様なアミノ酸がある程度並んでいるので、妥当な初期状態を与えることができる。しかし、一般に、アライメントしたい配列の長さはアンバランスであり、どの部分が類似しているかがわからない。すると、初期状態をどうするかという任意性が発生し、初期状態によってさまざまな解が得られるため、初期状態を変えて何回か繰り返すことが必要となってくる。

3 並列反復改善法

我々は、反復改善法に次の３つの点を拡張することにより反復改善法の問題点を克服し、実用規模の問題にも対処できるアライメントシステムを確立した。

- 試行錯誤過程の並列化
- 配列グループを限定して分割する方法
- ツリーベース状に配列を組み合わせる方法

それぞれの拡張点について、順に説明する。

3.1 試行錯誤過程の並列化

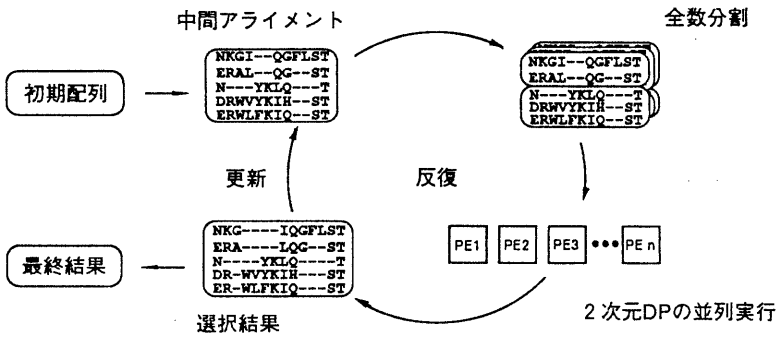


図 5: 並列反復改善法の手順

並列計算機を用いて、可能な分割を全て並列に実行することにより、再度試行を行う時間を節約し、反復改善法の全体の実行時間を短縮することができる [Ishikawa 92]。このように並列化を行うことにより、「問題点１」に対応できる。ICOT で開発された PIM という要素プロセッサ（PE）を 256 台もつ並列マシンでは、配列９本まで同時に試行可能である。我々は、この PIM を用いて９本の配列の全ての分割方法 $2^{9-1}-1=255$ 通りを同時に、グループ間の２次元 DP をし、その結果で一番評価値のよいものを、次のサイクルの初期状態として採用するという方法（図５）を行った。結果を集計して、一番評価値のよいものを選び出すマスタープロセスに１台プロセッサを割り当てているので、全部でプロセッサは 256 台（全部）使用される。この方法は探索問題における最急降下法に対応している。このような、並列化手法が拡張された反復改善法を「並列反復改善法」と呼ぶことにする。

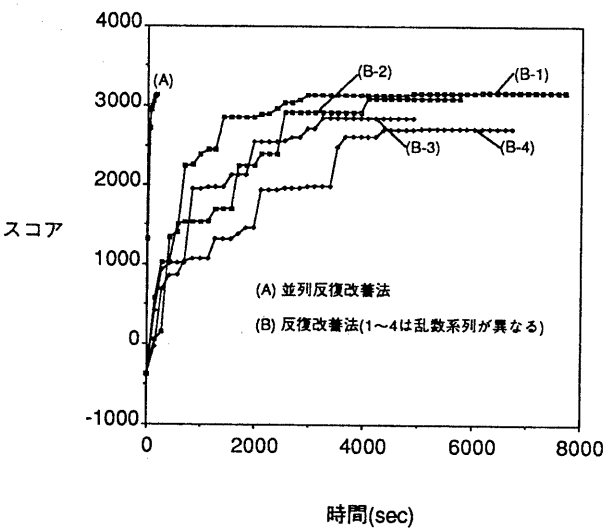


図 6: 並列反復改善法と反復改善法の比較 (配列９本)

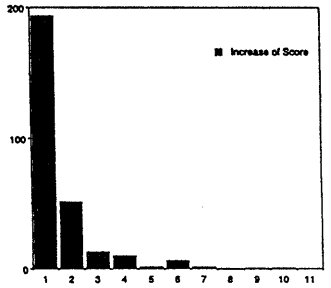
８０文字のアミノ酸配列９本のアライメント問題を対象に、並列反復改善法を評価したところ、図６に示されるような高速化が達成された。グラフを見て分かるように、並列反復改善法 (A) は非常に短時間の内に高得点に達しているのに対し、反復改善法 (B) は、改善が行われない試行がかなりあるので、グラフが水平に近い部分が多い。同じ程度の得点に達する時間で比較すると、約 60 倍近い差が存在している。また、反復改善法は乱数を使った逐次改善法のため、その系列によっては、比較的悪いローカルミニマに陥るものも多くあることが分かる。(B-3) (B-4) はこのような例である。並列反復改善法では、探索空間内を一番良い方向を見極めながら探索するので、ある程度良い解へ安定して至る。もちろん、並列反復改善法が必ずグローバルミニマに至るわけではないので、反復改善法の方が良い解を与えることも、まれに存在する。

3.2 限定分割法

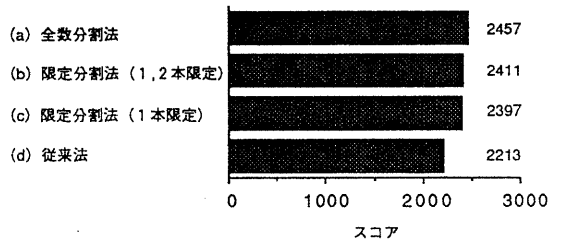
さて、我々の並列計算機で並列化を行っても、１０本以上の配列には一度には対応できないという問題は存在する。つまり１０本以上の配列を並列反復改善法で最急降下的に実行しようとすると、各グループ間 DP をもはや並列には実行できず、部分的に逐次的に実行せざるを得ない。本当に実用化を考えた場合は、２０本くらいまで並列に実行可能なのが望ましい。

我々は、「並列反復改善法」で実験を行った結果を分析するなかで、改善に寄与する配列の分割は、ほとんどの場合、本数が極端に偏っている場合だということを見い出した。その結果が図７の (1) である。図７の (1) は、２２本の配列について、１本と 21 本、および、２本と 20 本、３本と 19 本... というように分割を行ったとき、どの分割方法が、平均してどの程度スコアの改善に寄与したかを示している。ここで、グラフの横軸は、配列の少ない方のグループの配列の本数である。この実験では、普通の反復改善法とは異なり、配列の本数に関してある分割の仕方をまずランダムに選び、（例えば３本と 19 本の２グループに分けるとまず決めて）、その後でどの３本を選ぶかを決めてグ

ループ間の2次元DPを行い、そこで改善されたスコアを累計している。このグラフを見ると、N本の配列がある場合、1本とN-1本、および、2本とN-2本という分割が主に改善に寄与し、N/2本とN/2本というような分割は、非常にまれにしか改善に寄与しないことが分かる。



（1）配列の分割方法による改善度合の差



（2）全数分割 vs. 限定分割

図7: 限定分割法の妥当性

その理由は、おそらく以下のようなものであろう。N/2本とN/2本というような均等な分割を適当に行ってしまうと、それぞれのグループの各カラムには、多く（N/2個）のアミノ酸が存在し、それらのアミノ酸の性質がお互いに打ち消しあって、各カラムの個性（例えば疎水性の強いカラムであるとか、塩基性の強いカラムであるとかという特徴）をなくすような作用が働く。その結果、どちらのグループのどのカラムにも際だった個性がなくなってしまう。結果的に、ギャップを挿入するよりは、ギャップを挿入しないでそのままにしておいた方が全体としてのスコアが高いということになり、スコアが改善されないと推測できる。一方、1本とN-1本というような分割の場合は、1本の方の配列は、あるカラムが疎水性なら、その疎水性という性質がストレートにそのカラムの性質となるので、N-1本側のグループに微量だけ疎水性を示すカラムがあっても、そのカラムとマッチして得点を稼ぎやすくなるのだと考えられる。その結果、この2つのグループは相対的に動きやすくなり（つまりギャップが入りやすくなり）多くの場合、改善が見られることになると考えられる。我々が別途開発したアライメントエディタでは、各カラム毎のスコアを表示する機能があり、これを用いても、この理由がある程度正しいという確認を得ている。

このような配列の本数に関して非常に偏った分割を行う手法を「限定分割法」と呼ぶことにする。限定分割法を使用すれば、我々のプロセッサ256台構成の並列計算機で10本以上の配列を扱うことが可能になる。例えば、1本とN-1本および2本とN-2本という分割法だけを行うとすると、22本の配列を扱うことが可能である。（${}_{22}C_1 + {}_{22}C_2 = 22 + 231 = 253$）また、1本とN-1本というような分割法だけを行うとすると256本までの配列を扱える。もちろん、次の試行の初期状態には、並列に求めたうちの一番評価値のよいものを採用する。この限定分割法を用いることにより、前述の「問題点2」に対処できる。

限定分割法を検証するために、80文字のアミノ酸配列9本のアライメント30問について、(a)全ての分割を行うもの、(b)1本とN-1本、2本とN-2本という分割のみを行うもの（これを「1,2本限定」と呼ぶ）、(c)1本とN-1本という分割のみを行うもの（これを「1本限定」と呼ぶ）、それから、(d)典型的な従来法（ツリーベース法）の4通りについて比較を行ってみた。そのスコアの平均を求めた結果、図7の(2)のように、限定分割の方法は、全数分割の方法と、ほぼ同様なスコアを与えることが分かった。30問のうち、(a)が最大スコアを与えたのは12問、(b)が最大スコアを与えたのも同様に12問であった（残りは(c)が最大スコアを与えた）。限定分割を行えば、使用するプロセッサ台数を著しく削減でき、その分多くの配列を扱うことができる。また、「1,2本限定」と「1本限定」とを比較すると、「1,2本限定」の方がスコアが若干良いこと、反復改善法のスコアは、どれも、従来法よりもかなり良いこともわかった。

3.3 ツリーベース並列反復改善法

限定分割法によって実用的な規模まで配列の本数を増やすことができたが、限定分割を行うことによる弊害が発生する場合も存在する。それは、複数の配列の中に、際だって類似性の高い配列グループが存在し、その配列グループが3本以上の配列からなる場合である。我々は、MASCOTの開発段階において、類似性の高い配列同士はグループ

分けすることと、各グループ毎の初期のアライメントを注意深く行うことが、アライメント全体の品質に大きく影響を与えるという知見を得ていた[広沢 92]。そこで、ツリーベースのアライメントの各段階において、並列反復改善を行ってアライメント状態を確実なものにしながら、配列を組み合わせていくという「ツリーベース並列反復改善法」の発想に至った。

次に並列反復改善法ではうまく行かないが、ツリーベース並列反復改善法ならばうまく行く例を挙げよう。

```
並列反復改善法（限定分割1,2本選び）
  --------NGTTRVA-IKTLKPGTM---
  --------KKYSLTVAVKTLKEDTM---
  ----RLPSQDCKTVA-IKTLKDTSPGG-
  --------KDKTSVA-VKTCKEDLPQE-
  --RDIIKGEAETRVA-VKTVNESASLR-
  -VDILGVGSGEIKVA-VKTLKKGSTDQ-
  --HNLLPEQDKMLVA-VKALKEASES--
  IGLDKDKPNRVTKVA-VKMLKSDATEK-
  --FHLKGRAGYTTVA-VKMLKENASPS-
  ---IPEGEKVKIPVA-IKELREATSPK-
  ----KDTQKVYALKA-IRKSYIVSKS--
  ----KDTQRIYAMKV-LSKKVIVKKN--
  -VTGANTGKIFAMKV-LKKAMIVRNAK-
  ----KGTEELYAIKI-LKKDVVIQDD--
  ----KNTDRLCAIKV-LKKDNIIQNH--
  ----KETGNHYAMKI-LDKQKVVKLK--
  ---KSEESKTFAMKI-LKKRHIVDTR--
  ----ADTGKMYAMKC-LDKKRIKMK---
  ----KPTCKEYAVKI-IDVTGGGSFSAE
  ----VLAGQEYAAKI-INTKKLSAR---
  ----KKTGKVWAGKF-FKAYSAK-----
  ----KDDSKVVAIKV-ISKRGRATK---
           |---|

ツリーベース並列反復改善法
  --------NGTTRVAIKTLKPG------TM
  -------KKYSLTVAVKTLKED------TM
  ----RLPSQDCKTVAIKTLKDTS----PGG
  --------KDKTSVAVKTCKEDL----PQE
  --RDIIKGEAETRVAVKTVNESA----SLR
  -VDILGVGSGEIKVAVKTLKKGS----TDQ
  --HNLLPEQDKMLVAVKALKEA-----SES
  IGLDKDKPNRVTKVAVKMLKSDA----TEK
  --FHLKGRAGYTTVAVKMLKENA----SPS
  ---IPEGEKVKIPVAIKELREAT----SPK
  -------KDTQKVYALKAIRKSYIV--SKS
  -------KDTQRIYAMKVLSKKVIV--KKN
  ----VTGANTGKIFAMKVLKKAMIVR-NAK
  -------KGTEELYAIKILKKDVVI--QDD
  -------KNTDRLCAIKVLKKDNII--QNH
  -------KETGNHYAMKILDKQKVV--KLK
  ------KSEESKTFAMKILKKRHIV--DTR
  -------ADTGKMYAMKCLDKKRI---KMK
  -------KPTCKEYAVKIIDVTGGGSFSAE
  -------VLAGQEYAAKIINTKKL---SAR
  -------KKTGKVWAGKFFKAY-----SAK
  -------KDDSKVVAIKVISKRGR---ATK
```

この例では|---|付近にあるA（アラニン）が全部縦に揃った方が良いアライメントなのだが、配列群が真ん中のあたりから上下2つのグループに分かれてしまい、このグループ内から限定分割で1本や2本取り出して実行を行っても、もはや改善は見られない。なぜなら、グループ内での配列の類似性が高いので、選ばれた配列はグループ内の配列に強く引っ張られて、もう片方のグループの配列群の影響が及ばないためである。つまり、一種の局所的最適値（ローカルミニマ）に陥っており、抜け出すことができない例である。このような例を防ぐには、ひとつには全数分割を行うことだが、前述のように本数が多い場合、大規模な並列マシンをもってしても全数分割を行うのは困難である。

もうひとつ考えられる対策は、アライメントの状態から、どの配列とどの配列が、どのグループに属するかを検出し、適当な分割法を見い出す方法である。しかし、これを行うには、まだグループの分割基準などの研究課題が残されている。そこで我々は、そもそも始めからこうした不都合な状態をもたらさないように、ツリーベース実行を導入した。ツリーベース並列反復改善法を用いると、前の図の右側に示したような、A（アラニン）の揃ったアライメントが得られる。

これまでの反復改善法、および並列反復改善法では、初期状態となるアライメントに関して特別な指定はしていない。つまり、何らかの初期状態があればよいということで、我々は通常、アライメントしたい配列群を左詰め、あるいは右詰めにしたものを用いている。ツリーベース並列反復改善法では、このような恣意的な初期状態を用いるのではなく、前もって配列群のうち、どの配列とどの配列が近い関係にあるかを調べてツリーを作り、そのツリーに従って、徐々に配列の本数を増やしながら反復改善を行っていく。つまり、反復改善法の「問題点3」に対処することができ、初期状態に依存しないアライメントを得ることできる。

この方法の手順を具体的に述べると、次のように実行される。まず、配列の全てのペアについてDPを行いそのスコアを求める。そして、そのスコアを配列間の類似性と考え、良く似た配列から順に結線したツリーを作成する。これが前処理で、例えば図8に示すようなツリーが得られる。図8の下にある数字は配列の番号を示しており、図8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2次元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、2次元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並列反復改善を行い、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を行ったほうが良いので、今回は使用していないが、我々はその並列プログラムも用意している[戸谷 91]）。図8

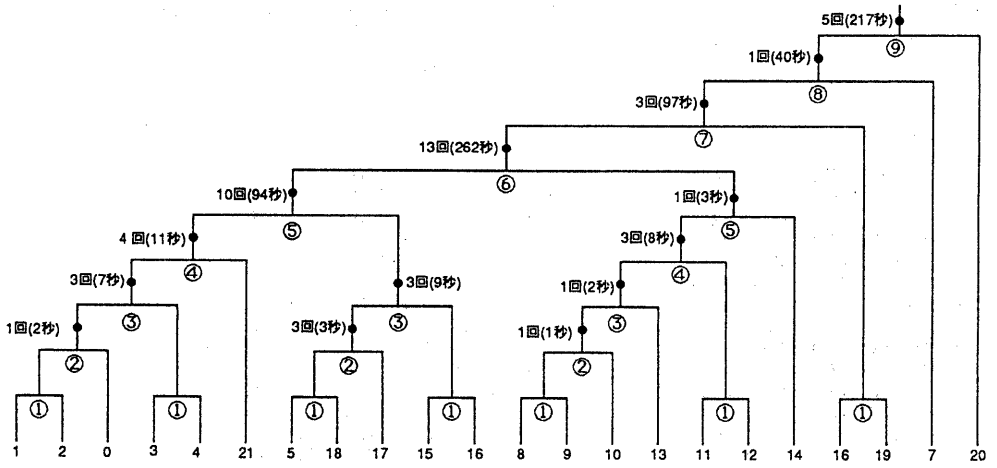


図 8: ツリーベース並列反復改善法の手順

の黒い丸で示される点は、並列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、何サイクル並列反復改善を行って、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2次元DPされた後の並列反復改善は、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。

また、ツリーベース並列反復改善法は、反復改善時の並列化に加えて、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8の中では①が、それぞれ個別にDP可能であるから、それらを並列に実行する。次に②がそれぞれ個別にグループ間のDPが可能であるから、それらを並列に実行する。といった具合である。⑥より上のレベルは、並列に実行する組合せはないが、その分、グループ間DPの後の並列反復改善法に、多くのプロセッサが費やされる。処理時間全体の比率からいうと、ツリー状に組み合わせるときの並列効果よりも、反復改善時の並列化の方が大きい。

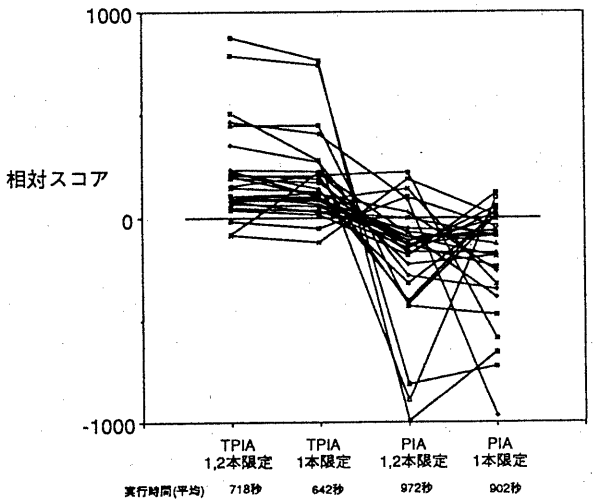


図 9: 並列反復改善法とツリーベース並列反復改善法の比較（限定分割あり、配列２２本）

ここで、並列反復改善法とツリーベース並列反復改善法との比較検討を行う。図9は、kinaseというグループのタンパク質（３０種）から、それぞれ類似性の比較的高い部分配列（８０文字分）を取り出した配列群（３０本）から、

ランダムに選び出した２２本に対して、それぞれの実験を３０回行った結果を示している。PIA(Parallel Iterative Alignment Method) とは並列反復改善法のこと、TPIA(Tree-based Parallel Iterative Alignment Method) とはツリーベース並列反復改善法のことである。どちらも限定分割を用いており、１本限定と、1,2本限定の２種を、PIA と TPIA のそれぞれに対して試した。相対スコアとは、４つの異なる方法を行ったときの、スコアの平均値からの相対的な差を表している。図９の実験から次のことが分かった。

- TPIA は、PIA よりも総じて良い結果をもたらしている。３０回の実験のうち、TPIA の方が悪い結果を与えたのは、ただ2回であり、その程度もわずかである。
- PIA における 1,2 本限定（PE253 台使用）のスコアと１本限定（PE22 台使用）のスコアは、平均して大きな差異はないが、同じ問題でも異なった値を与えている場合が多い。
- TPIA における 1,2 本限定のスコアと１本限定のスコアは、安定して同じような値を与えており、ほとんどの場合、1,2 本限定が若干良いスコアを与えている。事実、３０問中で、TPIA の 1,2 本限定が４手法のうちで最高のスコアを与えたことが、２２問あった。
- TPIA は、PIA より３割ほど実行時間が短い。また、1,2 本限定は、１本限定より、１割ほど実行時間が長い。

図９には示していないが、典型的な従来法（ツリーベース法）を用いて、同一の３０問を解く実験も行った。従来法は実行時間こそ平均 408 秒と、TPIA よりも４割ほど短いが、スコアは平均して、TPIA よりおよそ 3000、PIA よりもおよそ 2600 悪く、３０問のなかで TPIA、PIA のいずれかより良いスコアを与えたものはひとつもなかった。

以上より、我々の一連のアライメント手法は、実行時間は若干多く必要とするが、従来法よりも、高品質の結果を与えること、なかでも、TPIA の 1,2 本限定が、プロセッサは多く必要とするが、最も効果的な手法であることが、結論できる。

4　まとめと考察

我々は反復改善法に注目し、それをもとにして、実用規模の問題に高品質の解を与えるマルチプルアライメントシステムを構築した。そのために導入した要素技術は、並列化と、限定分割と、ツリーベース実行である。実験の結果、並列化と限定分割を導入した PIA（並列反復改善法）は、従来法より高品質のアライメントを与えることが判明した。さらにツリーベース実行も導入した TPIA（ツリーベース並列反復改善法）は、PIA よりも安定して高品質のアライメントを与えることもわかった。今回の実験は、比較的類似した配列部分を同じ長さに切って問題としたが、配列の長さが違う配列群をアライメントする場合には、類似した配列同士からアライメントしていく TPIA の方が、PIA より、ますます、きわだって良い結果を与えることが想像できる。今後は TPIA を、我々の標準システムとして評価改良を続けたい。

今後の改良検討の中心点は、限定分割のところである。現在 1,2 本限定を主体に限定分割を導入しているが、場合によっては、３本以上の分割も重要になってくる。我々は、余ったプロセッサで３本以上の分割のうち、いくつかを選んで並列に評価する構想を持っている。３本以上の分割のうち、どんな分割を試すかは、単にランダムに選ぶのではなく、配列の類似性を考慮したヒューリスティクスも有効である。また、並列評価をして、そのうちから最も良い解を選択する最急降下探索では、ローカルミニマに至る可能性がある。そこで、その可能性を少しでも抑えるため、最も良い解だけでなく、いくつかの解の候補を常に保持しておき、最も良い解が収束した後には、次の候補を反復改善する方法（いわゆるビームサーチ）の導入も考えている。この探索に、並列計算機を利用した遺伝アルゴリズムを導入することも、日本電気の小長谷氏のグループと検討している。

我々は、256 台という大規模な並列マシンで実行を行って大きな効果を得たが、もう少し小規模な並列マシン（十数台程度）を用いる場合は、限定分割に 1,2 本選びを行うのは、コストパフォーマンスが悪い。その場合は、限定分割を１本選びに限って実行すれば、許容できる実行時間内に、ある程度高品質の解が得られる。また、今回導入した要素技術の限定分割、およびツリーベース実行は、試行錯誤過程の並列化と同時に用いることによって、相乗効果を得たわけであるが、それらは並列化をしなくとも、それ自体に十分な意味があり、逐次マシンで、普通の反復改善法を行うときに限定分割やツリーベース実行を導入しても、解の質を向上させたり、実行時間を削減する効果がある。

本論文では、評価値の観点から高品質のマルチプルアライメントを得ることを目標にしてきたが、生物学的に高品質なアライメントには、単なる評価値には盛り込めないような視点が必要である [限啓— 92]。我々は、そのような

観点からも研究を行っており、TPIA をベースにして、知識処理を用いたマルチプルアライメント手法を検討している [Hirosawa 92]。

謝辞

ＩＣＯＴ遺伝子情報処理ワーキンググループ（主査：金久實）に参加されている、委員の方々、オブザーバーの方々、そして講師として参加された方々には、多くの助言や批判をいただき、深くお礼申し上げます。また，京都大学宮田研究室の隈啓一、岩部直之の両氏には、アライメントを利用する立場から、埼玉がんセンターの後藤修氏には、技術的立場から、とくに貴重な助言をいただきました。ここに感謝の意を表します。

最後に、本研究の機会を与えていただいた、ＩＣＯＴ第７研究室室長新田克己氏ならびに研究部長の内田俊一氏に感謝いたします。

参考文献

[石川 91] 石川、星田、広沢、戸谷、鬼塚、新田、金久: 並列推論マシンを用いたタンパク質の配列解析，情報処理学会情報学基礎研究会 23-2，1991.

[広沢 92] 広沢、星田、石川、戸谷: MASCOT— 3次元ダイナミックプログラミングに基づいた蛋白質のアライメントシステム，情報学シンポジウム講演論文集，日本学術会議，1992.

[木村資生 86] 木村資生：分子進化の中立説，紀伊国屋書店，1986.

[Dayhoff 78] M.O. Dayhoff, *et al.* *"A model of evolutionary change in protein"*, in *Atlas of Protein Sequence and Structure 5:3*, Nat. Biomed. Res. Found., Washington, D. C. 1978, pp.345–352.

[Jones 92] D.T. Jones, *et al.* *"The rapid generation of mutation data matrices from protein sequences"* in *CABIOS*, **8**, 1992, pp.275–282.

[Needleman 70] S.B. Needleman and C.D. Wunsch *"A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins"*, in *J. Mol. Biol.* **48**, 1970, pp.443–453.

[Waterman 81] T.F. Smith and M.F. Waterman *"Identification of common molecular subsequences"*, in *J. Mol. Biol.* **147**, 1981, pp.195–197.

[Carrillo 88] Himberto Carrillo and David Lipman *"The Multiple Sequence Alignment Problem in Biology"* in *J. Appl. Math.* **48**, 1988, pp.1073–1082.

[Feng 87] D. Feng and R.F. Doolittle *"Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees"* in *J. Mol. Evol.*, **25**, 1987, pp.351–360.

[Higgins 92] D.G. Higgins, *et al.* *"CLUSTAL V: improved software for multiple sequence alignment"* in *CABIOS*, **8**, 1992, pp.189–191.

[Barton 90] J.G. Barton *"Protein Multiple Sequence Alignment and Flexible Pattern Matching"* in *Methods in Enzymology Volume 183* Academic Press, 1990, pp.403–428.

[Berger 91] M.P. Berger and P.J. Munson *"A novel randomized iterative strategy for aligning multiple protein sequences"* in *CABIOS*, **7**, 1991, pp.479–484.

[後藤修 83] 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*, 1983, pp.477–480.

[Ishikawa 92] Ishikawa, Hoshida, Hirosawa, Toya, Onizuka, Nitta *"Protein sequece analysis by parallel inference machine"* in *Proceedings of Fifth Generation Computer Systems '92*, 1992, pp.294–299.

[戸谷 91] 戸谷、星田、石川、新田: 並列３次元ダイナミックプログラミング法によるタンパクの配列解析，情報処理学会第５回プログラミング研究会，1991.

[隈啓一 92] 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*, 1992, pp.62–64.

[Hirosawa 92] M. Hirosawa, M. Hoshida and M. Ishikawa *"Protein Sequence Analysis using Knowledge"* in 情報処理学会情報学基礎研究会 27-1，1992.