遺伝的アルゴリズムによる分子系統樹の作成

川本芳久† 松田秀雄† 橋本昭洋†

生物の系統分類に用いられる系統樹を，種々の生物から得られたDNA塩基配列またはアミノ酸配列などの配列データどうしを比較することにより作成する新しい手法について述べる．系統樹の作成では，配列データを比較する方式の異なるいくつかの方法が提案されているが，我々の方法は最尤法に基づいている．最尤法では，与えられた配列データから構成可能な系統樹の候補（候補系統樹）を生成し，DNA塩基またはアミノ酸の統計的な置換頻度をもとに各候補系統樹が実現される確率を尤度として求め，候補系統樹の中から尤度最大のものを選択することにより系統樹を作成する．しかし，配列データ数が増えると候補系統樹の数は急激に増大するので，すべての候補系統樹を探索するのは困難であり，何らかの発見的探索が必要になる．そこで，我々は，遺伝的アルゴリズムに基づいて尤度最大の候補系統樹を探索する手法を開発した．我々の手法により実際にアミノ酸配列から分子系統樹を作成したところ，ほとんどの場合において他の分子系統樹作成法よりも良い結果が得られた．

Construction of Molecular Phylogenetic Trees Using a Genetic Algorithm

YOSHIHISA KAWAMOTO,† HIDEO MATSUDA† and AKIHIRO HASHIMOTO†

This paper presents a new method to construct phylogenetic trees based on the comparison of DNA or amino acid sequence data obtained from organisms. Several distinct methods have been proposed and they are distinguished by the difference of the ways to compare sequence data. Our method is based on the maximum likelihood method which constructs a phylogenetic tree as follows; generate possible alternative trees, compute the likelihood of the trees based on the statistical substitution frequencies of DNA bases or amino acids, and selects the optimal tree which has the maximum likelihood. The number of alternative trees is however increased combinatorially with the growth of the number of sequence data. Thus the exhaustive search is impractical and some type of heuristic search method is required. We developed a method to search for the maximum likelihood tree using a genetic algorithm. From the experimental results on phylogenetic trees of amino acid sequence, our method shows better performance than almost all the results of the other methods.

1. はじめに

DNAの解析技術のめざましい発展とともに，DNA塩基配列やアミノ酸配列などの分子レベルのデータから生物の系統関係を解析する手法が注目されてきている．このような分子レベルのデータに基づく系統分類では，対象生物の進化過程を表す系統樹（従来の表現形による系統樹と区別するため**分子系統樹**と呼ぶ）の作成が中心となる．特に細菌などの微生物の系統関係では，表現形では区別できない場合があるなどの理由から，分子系統樹による系統解析がさかんに行われている．

分子系統樹の作成は，基本的には，まず分類の対象となる生物の間で共通した機能を持つ分子データ（たとえば，同じ遺伝子の情報を持つDNA塩基配列やそれらが翻訳された結果生成されたアミノ酸配列）を選び出し，それらを解析して，対象生物間の系統関係を推定することにより行われる．

分子系統樹の作成法は，現在までに数多く提案されているが[1]，我々はそれらの中で最尤法[2]について研究してきた[3]．最尤法は，対象生物のDNA塩基配列（またはアミノ酸配列）を葉とする木として構成された候補系統樹をモデルとして与え，そのモデルのもとで，進化の過程で起こるDNA塩基（またはアミノ酸）の置換によって，対象生物の配列が実現される確率（これを尤度と呼ぶ）を求める方法である．尤度は候補系統樹ごとに異なるので，尤度最大の候補系統樹を真の

† 大阪大学基礎工学部情報工学科
Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University

```
C.albicans        GGTGEFEAGISKDGQTREHALLAYTLGVKQLIVAVNKMDS--VKWDKNRFEEIIKETSNF
D.discoideum      SPTGEFEAGIAKNGQTREHALLAYTLGVKQMIVAINKMDEKSTNYSQARYDEIVKEVSSF
E.gracilis        STTGGFEAGISKDGQTREHALLAYTLGVKQMIVATNKFDDKTVKYSQARYEEIKKEVSGY
E.histolytica     AGTGEFEAGISKNGQTREHILLSYTLGVKQMIVGVNKMDA--IQYKQERYEEIKKEISAF
P.falciparum      ADVGGFDGAFSKEGQTKEHVLLAFTLGVKQIVVGVNKMDT--VKYSEDRYEEIKKEVKDY
S.acidocaldarius  AKKGEYEAGMSAEGQTREHIILSKTMGINQVIVAINKMDLADTPYDEKRFKEIVDTVSKF
S.cerevisiae      GGVGEFEAGISKDGQTREHALLAFTLGVRQLIVAVNKMDS--VKWDESRFQEIVKETSNF
                     *  .        **.**  *. *.*. *..*  ** *         *. *.
```

図1 配列データの例（一部）
Fig. 1 Example of molecular sequence data (partial).

分子系統樹の最もよい候補として選ぶ．候補系統樹どうしを尤度により比較できることから，最尤法は現在までに提案されている分子系統樹作成法の中では最も定量的な解析法として知られている．

しかし，構成可能な候補系統樹の数は，対象生物の数が増えると急激に増大することが知られており，何らかの発見的探索法が必要になる．さらに，著者等の経験では，候補系統樹を一種の山登り法により探索した結果，多数の局所最適解が存在することが分かった[3]．

そこで，本研究では，組合せ最適化問題の代表的な近似探索アルゴリズムのひとつである遺伝的アルゴリズム[4]（以下，GA と略す）により尤度最大の候補系統樹を探索することを試みた．以下では，分子系統樹作成の原理と最尤法，およびGAによる分子系統樹作成の実現について述べる．

2. 分子系統樹の作成

分子系統樹を作成するための入力データとしては，対象生物のDNAまたはRNAの塩基配列やアミノ酸配列が用いられる．現状では，すべてのDNA塩基配列が解読された生物はわずかなので，対象生物の間でそれらのDNA全部を比較して系統関係を解析することはほとんど不可能であり，一部分の配列を切り出して解析することになる．ここで，各生物から切り出してくる部分配列は，各生物の進化の過程を反映するような部分配列，すなわちそれらの生物間で共通の祖先種における単一の部分配列から受け継がれた配列になっている必要がある．これは，実際には各生物の間で共通の機能を持つ遺伝子のDNA塩基配列（またはアミノ酸配列）を選び出すことが広く行われている．

このように共通な機能を持つDNA塩基配列やアミノ酸配列であっても，進化の過程での塩基の欠損/挿入によりそれらの長さが多少異なっているので，系統樹作成に先だって，多重アラインメント（multiple alignment）と呼ばれる処理を行い，DNA塩基配列またはアミノ酸配列中の相同な部分が同じ位置に来るようにギャップを入れて補正する．これにより，分子系統樹の作成の際には，配列間でのDNA塩基またはアミノ酸の置換だけを考えればよいことになる（欠損は塩基またはアミノ酸からのギャップへの置換，挿入はギャップから塩基またはアミノ酸への置換と見なせる）．

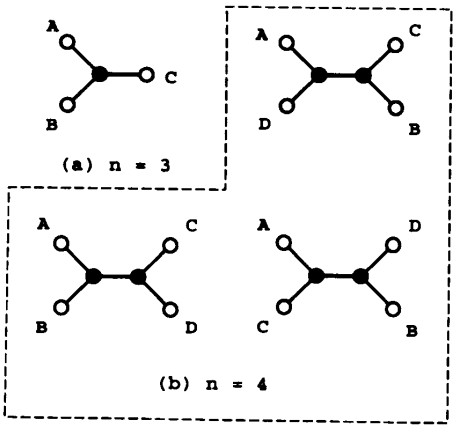


図2 無根木で表された系統樹
Fig. 2 Phylogenetic trees expressed as unrooted trees.

図1に多重アラインメントをかけた後のアミノ酸配列の例を示す．図1は，1種類の古細菌と6種類の真核生物のアミノ酸配列のうちの一部を取り出して横に並べたものであり，ギャップが"-"で表されている．なお，図1の配列の最初の列は，対象とした生物の学名の省略形を示しており，上から順に，カンジタ酵母，タマホコリカビ，ミドリムシの一種，赤痢アメーバ，マラリア病原虫，硫黄依存好熱菌の一種，パン酵母を示している．

図2(a)，(b)に，それぞれ，対象生物の数が3のときの系統樹と，対象生物の数が4のときの分子系統樹を示す．分子系統樹を木とみたときの葉節点を○，葉でない内部節点を●で表している．○は前述のアライメントされた配列に対応し，●は進化の過程で生物種の分岐が起こった位置，すなわち過去に存在したであろう生物（の配列）を表している．また，節点間の枝は，その両端の配列どうしを置換により関連付けている．枝の長さが置換回数を表しており，一方の端の配列からその長さで表された回数の置換が起こると，もう一方の端の配列に変化することを示す．

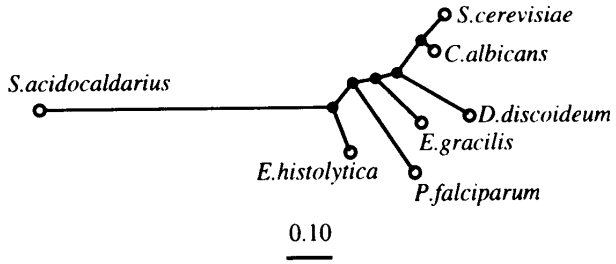


図3 分子系統樹の例
Fig. 3 Molecular phylogenetic tree.

表現形による進化系統樹と違って，分子系統樹は図2のように根（対象生物すべての共通祖先の配列を表す）を持たない無根木で構成されることが多い．これは，DNA塩基またはアミノ酸の置換だけの解析からでは，ある枝において進化の過程でどちらの方向に置換が起こったかの方向性が決定できないことによる．ただし，無根木の系統樹であっても，対象生物のいずれとも系統関係において大きくかけ離れている生物（これを群外種（outgroup）と呼ぶ）を選ぶことができれば，対象生物と群外種の共通の祖先の位置を対象生物だけからなる部分系統樹の根と見なすことができる．

図3に，図1に示したアミノ酸配列から作成した分子系統樹を示す．図3では，硫黄依存好熱菌の一種（*S. acidocaldarius*）は，他の生物とは系統的に大きくかけ離れていることが知られており，これを群外種として根の位置を決めている．なお，図3では下の0.10とラベルがついた線の長さが，配列の各位置あたり平均0.10回のアミノ酸置換に対応する長さを示している．

分子系統樹の作成法としては，現在までに様々な方法が提案されており，距離行列法，最大節約法，最尤法に大きく分けられる[1]．距離行列法とは，与えられた配列から，まずDNA塩基（またはアミノ酸）の置換回数に基づいて配列相互間の距離を計算し，それらの距離の情報から分子系統樹を作成するという2段階の手順をとる方法である．これに対して，最大節約法と最尤法は，配列を距離情報に変換せずにそのまま使って，構成可能な多数の候補系統樹の中から，ある尺度で最良と思われるものを選び出す方法である．その尺度は，最大節約法では候補系統樹での配列の置換回数（最小のものを選ぶ）であり，最尤法ではあらかじめ与えられたDNA塩基（またはアミノ酸）の置換確率に基づき計算された，候補系統樹の実現確率（これを尤度と呼び，尤度が最大のものを選ぶ）である．

これらの方法は，それぞれに一長一短があり，現時点ではどれが一番良いかは決められない．しかも，どの方法も，進化の過程で起こる置換に対してそれぞれ異なる仮定を導入しており，対象となる生物やそれらから切り出した配列によってその仮定が適当かどうかが変わってくる．

たとえば，最尤法については，仮定した置換確率が実際の進化における置換の起こり方と異なっている場合に問題が生じる．しかし，最尤法ではこのようなときでも，多くの場合において正しい候補系統樹を選ぶという頑健性（robustness）を持つことが，いくつかのシミュレーション実験から分かっている[5]．

3. 最尤法

3.1 対数尤度の計算

最尤法は，配列を構成するDNA塩基やアミノ酸の置換を1次のマルコフ過程に基づく確率モデルでモデル化している．たとえば，図2(a)にあるような対象生物3種の分子系統樹の場合は，尤度は以下のように求められる．なお，以下ではアミノ酸配列から尤度を求める方法に限定して述べているが，DNA塩基配列の場合も基本的には同じ方法で尤度が計算できる．

3種の分子系統樹は，3個の葉節点（現存する生物の既知のアミノ酸配列）と1個の内部節点（これらの生物の共通の祖先が持っていた未知のアミノ酸配列）からなる．この分子系統樹の3本ある枝の長さ，すなわち枝の両端の配列の各位置ごとの平均置換回数をそれぞれ v_1, v_2, v_3 とすると，配列上のある位置 i における尤度 $L(i)$ が式(1)で表される．

$$L(i) = \sum_{c^{(i)}=1}^{20} \left\{ \pi_{c^{(i)}} P_{c^{(i)} t_1^{(i)}}(v_1) \times P_{c^{(i)} t_2^{(i)}}(v_2) P_{c^{(i)} t_3^{(i)}}(v_3) \right\}, \quad (1)$$

ここで，$c^{(i)}$, $t_1^{(i)}$, $t_2^{(i)}$, $t_3^{(i)}$ は，それぞれ，内部節点の配列の位置 i におけるアミノ酸，および3つの葉節点に対応する配列の位置 i におけるアミノ酸を表し，20種類のアミノ酸に対応して1から20までの値を持つ．$c^{(i)}$ は未知なので，それのとりうる値それぞれについて計算したものを足し合わせる必要がある．また，π_x は中央節点の配列のアミノ酸が x を値として持つ確率（20種類のアミノ酸が一様に分布している場合は，$\pi_1 = \pi_2 = \cdots = \pi_{20} = 1/20$ となる）であり，$P_{yz}(v)$ は枝長 v だけ離れた2つの配列において，ある配列のアミノ酸 y が別の配列のアミノ酸 z に置換する遷移確率を表す．この遷移確率は，仮に各アミノ酸の間の置換が同じ頻度で起こると仮定すれば以下の式で表される．

$$P_{yz}(v) = \begin{cases} (1+19e^{-20\alpha v})/20, & (y=z) \\ (1-e^{-20\alpha v})/20, & (y \neq z) \end{cases} \quad (2)$$

ここで α は置換の頻度を表すパラメータである．この式は2つの配列間の距離 v が大きくなるにつれて（すなわち，共通祖先から分岐してからの時間が長いほど），あるアミノ酸から別のアミノ酸へ置換する確率（$y \neq z$ の式）が高くなり，置換が起こらない確率（$y = z$ の式，同じアミノ酸に置換する確率も含む）が低くなっていくことを表している．

実際には，アミノ酸配列は異なる置換頻度を持つので，Dayhoffらの PAM[6]やJonesらの JTT[7]などのアミノ酸どうしの置換頻度を統計的にまとめたデータが存在し，これらを使って遷移確率を求める手法が提案されている[8]．

最尤法では，アミノ酸置換は配列の各位置ごとに独立に起こることを仮定しているので，分子系統樹全体の尤度 L は式(3)のように求められる．

$$L = \prod_{i=1}^{m} L(i), \quad (3)$$

ここで，m はアミノ酸配列の長さである（多重アラインメントにより，各アミノ酸配列は同じ長さになるように調整されている）．なお，実際の計算では尤度そのものではなく，それの自然対数をとった対数尤度を求めることが多い．対数尤度 $\ln L$ は次の式で求められる．

$$\ln L = \sum_{i=1}^{m} \ln L(i). \quad (4)$$

分子系統樹の枝長は実際には，対数尤度 $\ln L$ を最大にするように，v_1，v_2，v_3 を未知数とする以下の連立微分方程式を解くことにより得られる．

$$\frac{\partial \ln L}{\partial v_1} = 0, \quad \frac{\partial \ln L}{\partial v_2} = 0, \quad \frac{\partial \ln L}{\partial v_3} = 0. \quad (5)$$

4種以上の分子系統樹は，3種の分子系統樹から段階的に種を付け加えていくことにより構成できる．たとえば，4種の分子系統樹は3種の分子系統樹の3本の枝のどれかに4種目の生物を付け加えることにより構成でき，5種の分子系統樹は4種の分子系統樹の5本の枝のどれかに5種目の生物を付け加えることにより構成できる．すなわち，一般に k 種の分子系統樹は，$k-1$ 種の分子系統樹が持つ $2k-5$ 本の枝に k 番目の生物を付け加えることにより構成できる．したがって，k 種の分子系統樹を構成するのに可能な候補系統樹の数は，

$$\prod_{k=3}^{n}(2k-5) = \frac{(2n-5)!}{2^{n-3}(n-3)!}, \quad (6)$$

という膨大な数になる．

式(6)から分かるように，種の数が増えると，構成可能な候補系統樹の数が増え，計算量が急激に増大する．したがって，すべての候補系統樹の尤度を調べるのは10種程度が限界で，これ以上の数の種の分子系統樹を作成するには何らかの発見的または近似的な探索手法が必要となる．

3.2 従来の探索手法

対数尤度最大の候補系統樹を発見的に探索するアルゴリズムとして，これまでにFelsensteinの開発した最尤法プログラム dnaml[9]で採用されている逐次付加法（stepwise addition），および足立らが開発した分子進化解析プログラム・パッケージ MOLPHY[10]で採用されている星状系統樹分割法（star decomposition）がある．

逐次付加法は，n 種の分子系統樹を構成するのに，まず**図4**(a)のように n 個の配列から任意の3個を選び3種の分子系統樹（候補系統樹はただ1つ）を作成し，4から n までの k について，図4(b)のように1個ずつ配列を選んで $k-1$ 種の系統樹から $2k-5$ 個の k 種の候補系統樹を作成していく方法である．各段階で対数尤度最大の候補系統樹1つを残して，残りの候補系統樹を捨てることにより，探索すべき候補系統樹の数を式(6)よりも大幅に減らしているが，この方法で得られるのは局所最適解であり，さらに結果が配列を選ぶ順番に依存するという特徴がある．

これに対して，星状系統樹分割法は距離行列法のひとつである近隣結合法に似た方式である．n 種の分子系統樹を構成するのに，まず**図5**(a)のように n 個の配列が1つの内部節点でのみ結合された星状系統樹を構成し，それらの配列の任意の2個を新しい内部節点でまとめた候補系統樹（$n(n-1)/2$ 個ある）の中から対数尤度最大のものを1つ選ぶ．このとき選ばれた

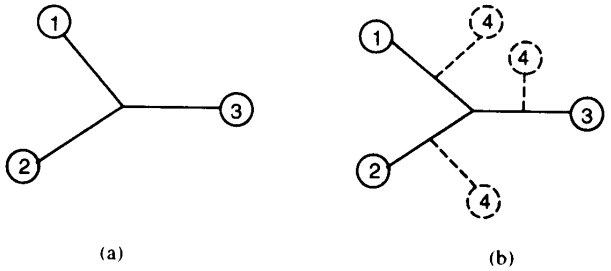


図4 逐次付加法
Fig. 4 Stepwise addition method.

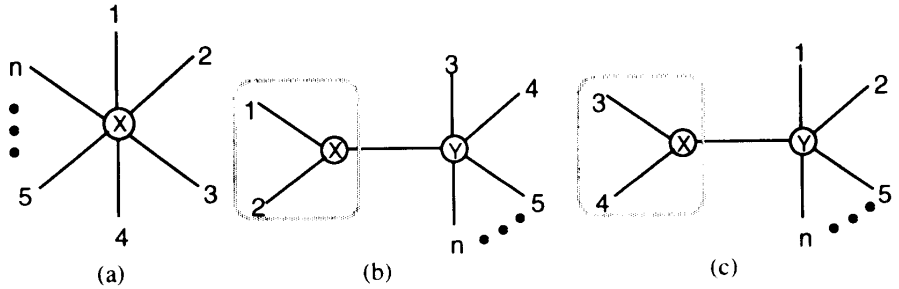


図5 星状系統樹分割法
Fig. 5 Star decomposition method.

配列の組を1つの内部節点で置き換え，この内部節点を加えた $n-1$ 個の配列の間でさらに2個ずつの組合せを行い，対数尤度最大のものを求める．この処理を繰り返すと，最終的には図2のような無根の2分木が得られる．この方法で得られる結果もやはり局所最適解ではあるが，逐次付加法のように配列の選択順に答が依存するということはない．

4. 分子系統樹作成への遺伝的アルゴリズムの適用

4.1 遺伝的アルゴリズム

GAは，生物進化の遺伝学的説明をもとにした，組合せ最適化問題の解法のひとつである．生物の進化過程においては，ある世代を構成している個体群から次の世代を生成する段階で，遺伝情報の突然変異や交叉によって元とは異なる個体が発生し，環境へより適応する個体が次の世代を構成していく．

GAでは，各個体は問題に対する解を，遺伝情報として1つ保持している．この遺伝情報は，突然変異や交叉といった操作が行える形でなければならず，また，適合度と呼ばれる関数が定義される．適合度は，問題に対する遺伝情報の優劣の判断に用いられる．このように表される個体の集団が世代交替を繰り返すことによって，最適解の探索が行われる．

各世代では，まず，個体の適合度が計算され，適合度に依存して個体の選択（増殖，淘汰）が行われる．次に，適当な2個体において，遺伝情報の交叉が行われる．さらに，各個体において，遺伝情報の突然変異が行われる．最後に，終了条件の判定が行われ，条件を満たした場合，その時点での適合度が最も高い個体が問題の準最適解となる．

4.2 最尤法への遺伝的アルゴリズムの適用

我々は最尤法に基づき対数尤度最大の候補系統樹を探索するアルゴリズムを，単純GA[4]をもとに開発した．GAを使った実現で問題になるのは，解くべき問題のデータ構造からGAで処理するデータ構造へのコード化，選択・交叉・突然変異などのオペレータや選択に必要な各個体の適合度の表現である．

まず，1個体で1つの候補系統樹を表すことにし，候補系統樹から個体の遺伝情報へのコード化については，単純GAのように2進数に変換するのではなく，候補系統樹をグラフ表現でそのまま表すことにした．具体的には，葉節点を正の整数，内部節点を負の整数でそれぞれ番号付けし，それらの間の接続関係を内部接点に隣接した節点番号の組を，内部節点の数だけ並べたリストで表現している．たとえば，図4(a)の系統樹は $\{(1,2,3)\}$ で表し，図4(b)で葉節点1が接続されている枝に葉節点4をつけたときの系統樹は葉節点1と葉節点4に隣接した内部節点を -1，葉節点2と葉節点3に隣接した内部節点を -2 と番号付けすることにより $\{(1,4,-2),(2,3,-1)\}$ で表している（節点番号の組の中での節点番号の順番と，リストの中での組の順番は任意）．

個体の持つ遺伝情報の表現が2進数ではないため，交叉，突然変異のオペレータには，単純GAで使われるような2進数列の部分的入れ換えやビットの反転といった手法は使えない．そこで，交叉と突然変異については，後述するような独自の方式を開発した．

なお，選択については単純GAと同じルーレット選択方式を採用した．ルーレット選択方式では，集団における全個体の適合度の和に対する各個体の適合度の割合によって，次の世代の個体として選択される確率を決定している．選択の基準となる各個体の適合度は，対数尤度最大の候補系統樹を探すわけであるから，基本的に対数尤度を適合度に選ぶだけでよいはずであるが，対数尤度は負の値をとるので，ルーレット選択方式では対数尤度の値をそのまま適合度とすることができない．そこで，各世代ごとに個体集団の中での対数尤度の最小値 $L_{min}^{(i)}$（i は世代）を求めておき，i 世代目での個体 j の適合度 $f_j^{(i)}$ をその個体の対数尤度 $L_j^{(i)}$ から以下のように計算することにした．

$$f_j^{(i)} = L_j^{(i)} - L_{min}^{(i)}. \quad (7)$$

これにより，適合度の値は負にはなることはなく，また，世代が進むにつれて個体集団における平均的な対数尤度の値が向上していっても，$L_{min}^{(i)}$ の値がそれに追従していくことから，世代ごとの適合度の値のばらつきをある程度抑えることができると考えられる．

以下，本アルゴリズムでは単純GAと同様，選択 → 交叉 → 突然変異という世代交替サイクルを，あらかじめ決められた世代数になるか，またはすべての個体がある特定のものに収束するまで繰り返す．なお，各世代において集団の中で最も適合度の高い個体は，後の交叉，突然変異で消えてしまうことがないようにしている（エリート保存）．

4.3 交叉・突然変異の実現

GA における交叉の考え方の基本には積木仮説（building block hypothesis）がある[4]．これは，個体の遺伝情報中に積木，すなわち適合度の向上に寄与する短いコード・パターンがいくつか存在し，これらを組み合わせることでより適合度の高い個体を得ることができるというものである．

分子系統樹の作成においても同様に，候補系統樹を構成する木構造の中に，対数尤度の向上に寄与する木構造パターンがいくつか存在すると考えられ，これらを積木として組み合わせることで，より対数尤度の大きい候補系統樹が得られると考えられる．

しかし，候補系統樹の中からこの積木を得るのは容易ではないので，我々は逆に積木ではない，すなわち対数尤度の向上に寄与しないだろうと思われる木構造パターン（これを反積木と呼ぶことにする）を求めることを考えた．反積木内では，遷移確率が低い，すなわち式(2)から分かるように節点間の距離が大きいはずである．これは，類似性の低い配列どうしが近接して置かれていることを意味する．

2つの個体が表す分子系統樹の反積木以外の部分を組み合わせた候補系統樹は，元の系統樹と比べて必ずしも対数尤度の向上したものになるとは限らないが，近接して類似性の低い配列が置かれている状況を改善することから，結果的に積木を組み合わせた対数尤度の大きいものとなることが期待できる．

具体的な交叉の手順を以下に示す．

Step 1. 集団の中から2つの個体 i と j を選び，それらの表している候補系統樹における葉節点間の距離をそれぞれ求める（**図6** の (a), (b)）．

Step 2. Step 1 で求めた距離から，次の式のように i と j それぞれの葉節点間の距離における相対的な差分 $r_{ij}(x,y)$ を計算する．

$$r_{ij}(x,y) = \frac{|d_i(x,y) - d_j(x,y)|}{d_i(x,y) + d_j(x,y)}, \quad (8)$$

ここで，x, y は葉節点，$d_i(x,y)$, $d_j(x,y)$ はそれぞれ i, j における x と y の間の距離を表す．

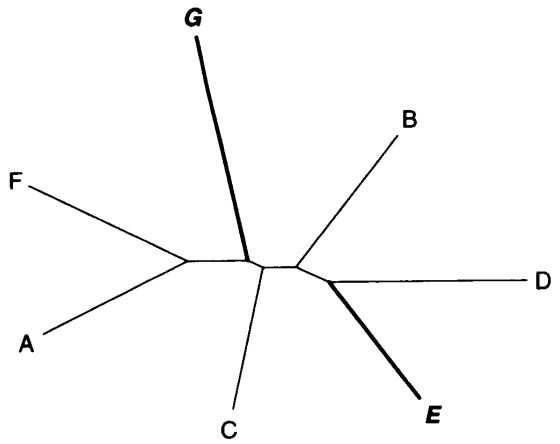
Step 3. Step 2 の結果から $r_{ij}(x,y)$ が最大値をとるときの x, y の値の組を探す（これを x', y' とする）．そして，$d_i(x',y')$ と $d_j(x',y')$ を比べる．今，仮に $d_i(x',y') > d_j(x',y')$ であれば，i は j との比較において x'-y' 間で大きく距離がかけ離れていることが分かる．つまり，i の x' と y' を含む木構造パターンは反積木の候補と考えることができる．以下，反積木を持つ方の個体を表す候補系統樹を凸凹木（rugged tree），反積木を探すのに使われた候補系統樹を参照木（reference tree）と呼ぶことにする（図6では (a) と (b) で式(8)が最大になるのは E と G との間の距離においてである．$d_{(a)}(E,G) > d_{(b)}(E,G)$ であるから，(a) が凸凹木，(b) が参照木となる）．

Step 4. 参照木から x', y' を含む最小の部分木 t_{ref} を取り出す．また，凸凹木からは t_{ref} に現れる葉節点を取り除いた部分木 t_{rug} を作る．t_{ref} と t_{rug} とを連結させて新しい候補系統樹を作る．これが，交叉により得られた候補系統樹となる（図6(c)）．なお，t_{ref} と t_{rug} との連結ではそれらのどこをつなぐかが問題になるが，ここではあらかじめ基準点となる葉節点を決めておいて，元の凸凹木，参照木における基準点との接続関係が保存されるように連結することにする（図6(c)では葉節点 A を基準点にとっている）．

次に，突然変異については，内部節点どうしをつなぐ枝を1つ選んで，それにつながっている枝もしくは部分木どうしの入れ換えを行う（これを分枝交換と呼ぶ）．たとえば，図6(d)は(c)の葉節点 F につながる枝と葉節点 B, D を持つ部分木とを入れ換えている．

5. 実行結果

我々の提案する手法の有効性を調べるため，提案手法と従来の手法（平均距離法，近隣結合法，最大節約法，最尤法）で実際に分子系統樹を作成した結果をそれらの対数尤度により比較した（**表1**）．最尤法以外の手法では得られた分子系統樹を候補系統樹として3.1節で述べた方法により枝長と対数尤度を計算している．また，最尤法では，3.2節で述べた逐次付加法および星状系統樹分割法により対数尤度最大の候補系統樹の探索を行った結果と，我々のGAによる探索結果とを

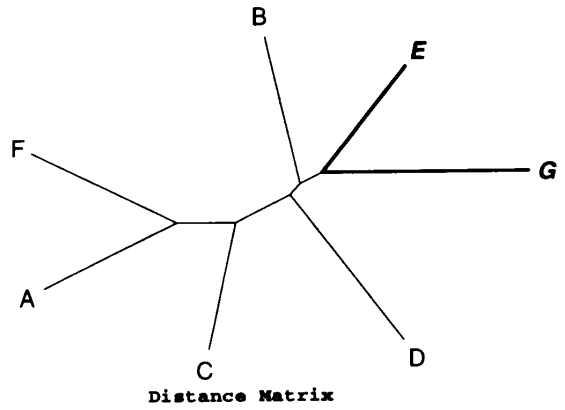


Distance Matrix

	A	B	C	D	E	F
A						
B	0.4548					
C	0.3656	0.3585				
D	0.6495	0.6053	0.5532			
E	0.4119	0.3676	0.3155	0.5232		
F	0.4435	0.5138	0.4246	0.7085	0.4709	
G	0.8621	0.8709	0.7817	1.0656	0.8280	0.9211

Ln Likelihood = -4470.4

(a) a rugged tree

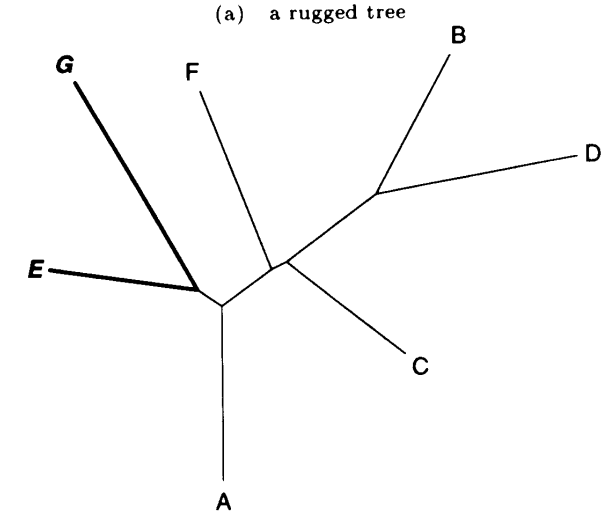


Distance Matrix

	A	B	C	D	E	F
A						
B	0.4673					
C	0.3543	0.3687				
D	0.6467	0.5936	0.5481			
E	0.4243	0.3621	0.3258	0.5507		
F	0.4431	0.5241	0.4111	0.7035	0.4811	
G	0.9037	0.8414	0.8051	1.0300	0.7758	0.9604

Ln Likelihood = -4468.9

(b) a reference tree

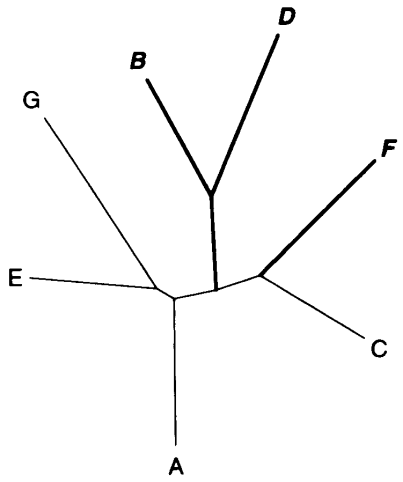


Distance Matrix

	A	B	C	D	E	F
A						
B	0.4565					
C	0.3672	0.3559				
D	0.6593	0.5096	0.5587			
E	0.3579	0.4044	0.3151	0.6072		
F	0.5048	0.4936	0.4043	0.6964	0.4527	
G	0.8520	0.8985	0.8092	1.1013	0.7672	0.9468

Ln Likelihood = -4461.9

(c) a tree constructed by crossover



Distance Matrix

	A	B	C	D	E	F
A						
B	0.4443					
C	0.3743	0.3624				
D	0.6481	0.5070	0.5662			
E	0.3556	0.3924	0.3225	0.5963		
F	0.5135	0.5016	0.3797	0.7055	0.4617	
G	0.8467	0.8835	0.8135	1.0873	0.7652	0.9528

Ln Likelihood = -4457.9

(d) a tree constructed by mutation

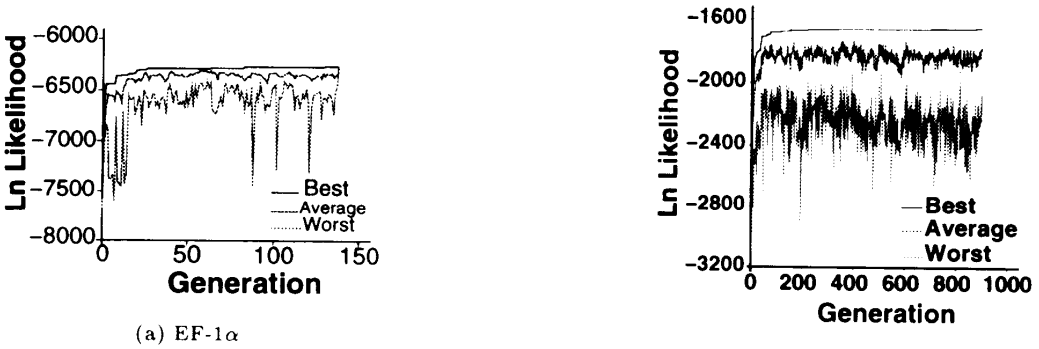
図6 交叉および突然変異の実行例
Fig. 6 Example of crossover and mutation.

比較している．また，配列データとしては，GenBank およびPIRから，タンパク伸長因子 (elongation factor) EF-1α のアミノ酸配列を 15 種類の生物について取り出したもの[11]（以下，EF-1α と略す）と，細菌における転写因子のひとつである σ 因子を細菌の種類および σ 因子の種類をそれぞれ変えて組み合わせた 21 種類の配列[12]とを使用した．

GA による探索の実行では，各世代の個体数と交叉

表1 対数尤度による手法ごとの結果の比較
Table 1 Comparison of tree-construction methods based on the log-likelihood scores of the resulting trees.

分子系統樹作成法	対数尤度（EF-1α）	対数尤度（σ 因子）
平均距離法	−6301.2	−1657.0
近隣結合法	−6301.8	−1655.7
最大節約法	−6267.7 .. −6270.3	−1653.7 .. −1657.0
最尤法（星状系統樹分割法）	−6309.6	−1658.7
最尤法（逐次付加法）	−6260.6 .. −6998.8	−1663.5 .. −2291.8
最尤法（GA）	−6260.7	−1652.0



(a) EF-1α

(b) σ 因子

図7 対数尤度の推移
Fig. 7 The improvement of log likelihood scores.

および突然変異の発生率の決定が重要である．個体数については，アミノ酸配列の数より大きい適当な数がよいと考えられるので，EF-1α については 20，σ 因子については 40 に設定した．また，交叉および突然変異の発生率については，具体的な決定方法がないため，交叉の発生率を 0.1 から 0.7 まで，突然変異の発生率を 0.1 から 0.6 まで，それぞれ 0.1 ずつ変化させたものについて予備実験を行い，それに基づいて設定を行った．予備実験では，新しい個体が 1000 個体発生するまでプログラムを実行し，その結果，候補系統樹の対数尤度が最大となった，交叉および突然変異の発生率（それぞれ，EF-1α では 0.5 および 0.2，σ 因子では 0.5 および 0.1）を用いることとした．それらの値に基づき，プログラムを実行した結果，EF-1α では 138 世代目にすべての個体が 1 つの分子系統樹に収束したのでそこで実行を打ち切り，σ 因子では 900 世代目までで実行を打ち切った．

表 1 で，最大節約法の結果と逐次付加法で探索したときの最尤法の結果の対数尤度値にばらつきがあるが，これは最大節約法の場合は置換回数最小の系統樹が複数得られたためと，逐次付加法の場合は付加するときの配列の選択順序に結果が依存するため順序をランダムに変えて 20 回実行したためである．

表 1 から分かるように，我々の手法により σ 因子では最も対数尤度の大きい結果が得られ，EF-1α では逐次付加法で探索した結果のうちの最も良いものよりは少し劣るが，それ以外では他の手法よりも良い結果が得られている．

図 7 は，世代交代に従って対数尤度がどのように変化したかを，各世代の対数尤度の最高値（Best），平均値（Average），最低値（Worst）について示したものである．図 7 (a) は，130 世代目で対数尤度の最高値が −6260.7 に達し，138 世代目ですべての個体が 1 種類の候補系統樹に収束したことを表している．これは，本手法の最適解への収束性を表しているものではなく，むしろ個体数が 20 と少ないことによる局所最適解への初期収束によるものと考えられる．一方，図 7 (b) では，877 世代目で対数尤度の最高値が −1652.0 に達しているが，その後も収束する様子はない．その理由としては，σ 因子では 21 個の配列から構成される候補系統樹を探索しており，15 個の配列を使っている EF-1α と比べて組合せの数が大きいことが考えられる．

なお，これらの結果を得るのに要した計算時間は，1 世代の結果を得るのに，日本電算機製ワークステーション JP4（CPU はクロック 100 MHz の PowerPC 604）での CPU 時間で，EF-1α では約 192 秒，σ 因子では約 351 秒となっている．

6. おわりに

我々は，分子系統樹を作成するための一手法である最尤法に遺伝的アルゴリズム（GA）を組み込み，従来の手法より良い結果を得ることに成功した．

最尤法は，与えられた配列から構成可能な候補系統樹を，分子進化の研究から得られているDNA塩基またはアミノ酸の置換頻度をもとに計算される対数尤度により比較する方法で，分子系統樹作成法の中では最も定量的な解析法として知られている．しかし，その反面，取り扱う対象生物の数が増えると候補系統樹の数が急激に増大することから，多数の局所最適解が存在することが分かっている．

一方，GAは，組合せ最適化問題の代表的な近似解法である．このGAを最尤法に適用することで，最尤法の欠点を補いつつ，従来の手法と比べてより対数尤度の大きい解を得ることが可能となった．

今後の課題としては，本手法においてはGAによる探索の一般的な傾向として収束に非常に時間がかかっているため，選択・交叉・突然変異について別方式の採用も含めたチューニングを行い，収束性能を向上させることがあげられる．

参考文献

1) 日本生化学会編：分子進化実験法，第23章 系統樹作成法，東京化学同人，pp.373–416 (1993).
2) Felsenstein, J.: Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach, *J. of Molecular Evolution*, Vol.17, pp.368–376 (1981).
3) Olsen, G.J., Matsuda, H., Hagstrom, R. and Overbeek, R.: fastDNAml: A Tool for Construction of Phylogenetic Trees of DNA Sequences using Maximum Likelihood, *Computer Applications in Biosciences*, Vol.10, No.1, pp.41–48 (1994).
4) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addisson-Wesley (1989).
5) Fukami-Kobayashi, K. and Tateno, Y.: Robustness of Maximum Likelihood Tree Estimation against Different Patterns of Base Substitutions, *J. of Molecular Evolution*, Vol.32, pp.79–91 (1991).
6) Dayhoff, M., Schwartz, R. and Orcutt, B.: A Model of Evolutionary Change in Proteins, *Atlas of Protein Sequence and Structure*, Vol.5, No.3, pp.345–352 (1978).
7) Jones, D., Taylor, W. and Thornton, J.: The Rapid Generation of Mutation Data Matrices from Protein Sequences, *Computer Applications in Biosciences*, Vol.8, pp.275–282 (1992).
8) Kishino, H., Miyata, T. and Hasegawa, M.: Maximum Likelihood Inference of Protein Phylogeny and the Origin of Chloroplasts, *J. of Molecular Evolution*, Vol.31, pp.151–160 (1990).
9) Felsenstein, J.: PHYLIP Manual Version 3.3, University Herbarium, University of Cariforia, Berkeley (1990).
10) Adachi, J. and Hasegawa, M.: MOLPHY: Programs for Molecular Phylogenetics I – PROTML: Maximum Likelihood Inference of Protein Phylogeny, Computer Science Monographs 27, Institute of Statistical Mathematics, Tokyo (1992).
11) Hasegawa, M., Hashimoto, T., Adachi, J., Iwabe, N. and Miyata, T.: Early Branchings in the Evolution of Eukaryotes: Ancient Divergence of Entamoeba that Lacks Mitochondria Revealed by Protein Sequence Data, *J. of Molecular Evolution*, Vol.36, pp.380–388 (1993).
12) Nakahigashi, K., Yanagi, H. and Yura, T.: Isolation and Sequence Analysis of RpoH Genes Encoding Sigma32 Homologs from Gram-negative Bacteria: Conserved mRNA and Protein Segments for Heat Shock Regulation, *Nucleic Acids Research*, Vol.23, No.21, pp.4383–4390 (1995).

(平成8年1月18日受付)
(平成8年4月12日採録)

川本 芳久

昭和42年生．平成3年大阪大学基礎工学部化学工学科卒業．平成5年同大学院基礎工学研究科物理系専攻情報工学分野修士課程修了．平成8年同大学院博士後期課程単位取得退学．同年同大学基礎工学部情報工学科助手．遺伝子情報処理，統合データベース，文書検索に関する研究に従事．

松田　秀雄（正会員）

昭和34年生．昭和57年神戸大学理学部物理学科卒業．昭和59年同大学院工学研究科システム工学専攻（修士課程）修了．昭和62年同大学院自然科学研究科（博士課程）修了．同年同大学工学部助手となり，同大学講師，助教授を経て，平成6年10月より大阪大学基礎工学部情報工学科助教授，現在に至る．この間，平成3年4月より10カ月間米国アルゴンヌ国立研究所客員研究員．学術博士．論理型言語による並列処理，遺伝子情報処理の研究に従事．電子情報通信学会，IEEE CS，ACM各会員．

橋本　昭洋（正会員）

昭和36年大阪大学工学部通信工学科卒業．昭和41年同大学院工学研究科博士課程修了．工学博士．同年NTT電気通信研究所に勤務．昭和44～46年イリノイ大学計算機科学科客員助教授．昭和60年NTTデータ処理研究部長，昭和62年情報科学研究部長．この間計算機の故障診断，自動設計，大型計算機DIPSの開発等に従事．平成1年大阪大学基礎工学部情報工学科教授，平成6年同大学情報処理教育センター長を併任，現在に至る．最近は分子生物学関連の情報処理技術の研究に従事．著書：計算機アーキテクチャ（平成7年・昭晃堂）．電子情報通信学会，IEEE，ACM各会員．