データベースシステム 119－7
（1999．7．21）

文書画像からの分子系統樹データの抽出手法とその評価

廣中大雅† 田部智宏† 吉川孝伸†
松田秀雄† 橋本昭洋†

分子生物学の分野では各生物種の遺伝子やタンパク質の情報を解析する研究が盛んに行なわれ，多数の生物種における遺伝子データが急激に蓄積されつつある．遺伝子データの分子進化解析の手法は種々存在するが，その解析を行なう上で基本的なデータ表現の一つに分子系統樹と呼ばれる表現方法がある．しかし，その多くが学術雑誌等で図として表現されているので，内容の検索ができない．そこで，図形認識技術を用いて，紙面上の図形情報を Newick Standard 形式と呼ばれるテキスト形式に変換し，検索を可能にしようと試る．本研究では，画像情報を Newick Standard 形式に変換する部分に焦点をあて，認識プログラムによる自動変換を試みている．

A Method for Extracting Molecular Phylogenetic Tree Data from Document Images and Its Performance Evaluation

Hiromasa Hironaka,† Tomohiro Tabe,† Takanobu Yoshikawa,†
Hideo Matsuda† and Akihiro Hashimoto †

With the recent rapid progress of DNA sequencing technology, the amount of genetic data being made available has been increasing at a tremendous rate. New methods are required for comparing and examining these genetic data based on phylogenetic analysis. Here, we propose a phylogenetic tree database. Phylogenetic trees represent one of the major methods for representing the result of molecular phylogenetic analysis. However, serchable phylogenetic tree databases are difficult to implement, because the tree are generally only made available as figures in published papers. In this report, we try to transform phylogenetic trees in published papers into a text format representing its structure, called the Newick format.

1. はじめに

分子生物学の分野では各生物種の遺伝子やタンパク質の情報を解析する研究が盛んに行なわれ，多数の生物種における遺伝子データが急激に蓄積されつつある．このため，これらの遺伝子データを分子進化解析により比較・考察する要求が高まってきている．遺伝子データの分子進化解析の手法は種々存在するが，その解析を行なう上で基本的なデータ表現の一つに分子系統樹と呼ばれる表現方法がある．現在までに数多くの分子進化解析結果が学術雑誌等の文献中に図として添付されている．しかし，図として添付されているので，既存の文献データベース等ではその内容が検索できない．そこで，我々は Newick Standard というテキスト形式の系統樹の表記法を用いることで，系統樹をデータベースに蓄積し検索を可能にしようとしている．データベースには系統樹データの他に出典文献名などの付加情報をつけて検索をしやすくする．しかし，大量にある系統樹の文書画像を全て手作業で入力するのは不可能である．そこで，文書画像を自動的に認識し，Newick Standard 形式に変換する要求が生じる．

本研究では，分子系統樹データベースに蓄積するデータを作成するために，文献中の系統樹をデータベースに格納できる形に変換することを目的とする．

2. 分子系統樹データベース

2.1 分子系統樹とは

分子生物学の発展にともない，生命の設計図である遺伝情報を手に入れることができるようになった．この離散的な遺伝情報をもとに生命の進化を分子レベルで表現したものが分子系統樹である．分子系統樹の例を図 1 に示す．この 2 つの分子系統樹は両方とも同一の解析結果を示しており，左は無根木 (unrooted tree)，右は有根木 (rooted tree) と呼ばれる系統樹の表現の仕方である．葉節点は主に現生の生物種あるいは遺伝子を示す．中間節点はアミノ酸配列やその遺伝子の配列から推定される中間種あるいは共通祖先を示

† 大阪大学大学院基礎工学研究科情報数理系専攻
Graduate School of Engineering Science, Osaka University

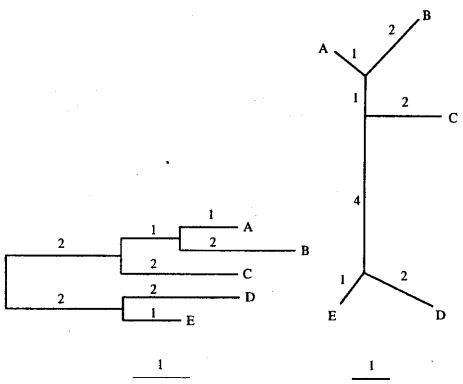

図1　有根木（左），無根木（右）

す．各節点間の長さは遺伝的な距離を示す．

2.2　分子系統樹データベースの必要性

現在，公表されている分子系統樹はその多くが学術雑誌等の文献中に図として添付されている．そのため，系統樹間の比較や演算は容易なことではなかった．さらに，近年のゲノム解析技術の進歩に伴い，そのデータ量が飛躍的に増え，大量にある分子系統樹を単純に遺伝子や生物種名などを検索するだけでなく，ある系統樹に似ている系統樹を検索するなどの，系統樹の構造に着目した検索の要求も高まってきている．しかし系統樹の構造が直接検索できる形で格納されているデータベースが存在しないことが，その障害となっている．

これらの問題点を解決するために系統樹のデータをデータベースに蓄積することが求められるが，そのまま取り込むと画像データとなってしまい，系統樹の構造の検索には不向きである．そのため，系統樹を計算機が処理しやすい形で蓄積する必要が生じる．その蓄積手段として，次に示すNewick Standard形式というテキスト形式で系統樹を表記した分子系統樹データベースを提案する．この形式は系統樹の構造を簡潔に表現しているため，構造を比較する処理に向いている．

2.2.1　Newick Standard とは

Newick Standard とは，系統樹の表記法の一つであり，木構造を括弧 '(',')' と ',' を使い構造を一意に決定できる表記法である．枝の長さも情報として持たせることも可能である．例えば，図1右をNewick Standard 形式で表すと，次のようになる．

```
((A:1 , B:2):1 , C:2 , (E:1 , D:2):4 );
```

表記中のコロン ':' の後ろの数字は枝の長さを表している．一つの木構造から最初に注目する葉ラベルや内部節点の違いでいくつかのNewick Standard 形式が得られるが，一つのNewick Standard 形式の記述からは木構造は一意に決まる．

2.3　既存の分子系統樹データベースとその問題点

今までにも，次に示すようにインターネット上で分子系統樹を検索できるサイトがある．

- UCMP Exhibit Halls , Internet address: http://www.ucmp.berkeley.edu/exhibit/phylogeny.html[1)]
- A new version of the RDP(Ribosomal Database Project), Internet address: http://www.cme.msu.edu/RDP/analyses.html[2)]
- The Tree of Life: A multi-authored,distributed Internet project containing information about pylogeny and bildiversity, Internet address: phylogeny.arizona.edu/tree/phylogeny.html[3)]
- JUNGLE: Phylogenetic Tree Database, Internet address: http://smiler.lab.nig.ac.jp/jungle/jungle.html[4)]

しかし，上記のデータベースでは利用者が文献名や，図の表題などのキーワードで検索するインターフェースを提供している場合がほとんどで，分子系統樹同士を直接比較できる機能を提供していない．

3.　系統樹データへの変換

3.1　変換手順概要

前処理

入力される図形には系統樹だけではなく，文字や記号などの画素も含まれている．しかし，文字や記号などが含まれていると系統樹の接続関係を抽出する際に，系統樹との区別ができないので，誤った抽出をすることになる．そこで，いったん，文字や記号を消去し，種名や遺伝子名など，データベースを検索する際に必要となる情報は，あとから，入力することにする．また，一般にスキャナなどを用いて入力される線図は線幅をもち，以後の処理量を軽減するために細線化を施して線幅が1画素の図形に変換する．この二つの処理を合わせて前処理と呼ぶことにする．

スケールバーと系統樹の読みとり

前処理を施した図形には，系統樹とスケールバー（線幅がいずれも1画素）のみが含まれている．スケールバーについては，その長さと対応する遺伝的な距離を読みとった後で消去する．

系統樹については，画素の集合から，始点と終点をもつ線分へ変換する．抽出された線分の集合から，線分の交点を捜し出して，交点の集合を作る．この時点で系統樹の接続関係と線分の長さはすべてて得られるはずだが，実際には，線分が短すぎて，抽出できなかったり，交点を誤ってしまったりするので，補正してやる必要がある．

出力ファイル

最後に系統樹から得られた接続関係と線分の長さと，スケールバーから得られた線分の長さと遺伝的な距離から，Newick Standard と呼ばれる系統樹の記述方式に変換する．さらに，足らない情報（文献名等）をお

ぎなって最終出力ファイルとして出力する．

3.2 各手順の詳細

3.2.1 前処理（細線化）

一般に，図面をスキャナで読みとり画像データに変換する際には，線の太さより細かな解像度で入力する必要がある．その結果ディジタル画像データ上の線は数画素程度の幅をもつことになり，どのようにして画素中の中心の位置を決めるかが問題となる．これを決定する方法として，本研究では，Hildich の細線化[5]を用いている．

3.2.2 前処理（不要な画素の消去）

対象とする図形には，記号や数字が文字列領域以外に，系統樹のなかにも多数含まれている，さらに，記号が系統樹と連結していたり，逆に系統樹が途切れ途切れになっていたりするので，これを，自動で認識するのは困難だと思われた．そこで，画像中に散らばっている文字や記号を消去する方法として，連結画素ごとに系統樹でであるかどうかをオペレータが判断することにした．

連結画素というのは，その8近傍にお互いに画素がある塊で，その連結する画素の塊に対して，同じ番号を割り当てる操作（ラベル付け）を行う．そして，その番号ごとに，系統樹であるかそうでないのかを対話的に指定することにする．

3.2.3 直線の抽出

実際の画像では，必ずしも線分が連結しているとは限らず，かすれて切れ切れになっている場合がある．このような状況においても，線分を抽出する手段としてハフ変換[6]がある．

簡単な例として2箇所に点のある画像を使ってハフ変換を説明する．この2点の座標 (x,y) について

$$\rho = x\cos\theta + y\sin\theta \qquad (1)$$

という変換式を用いると，この2点は（θ，ρ）平面上の2曲線を表すことになる。この曲線は**図2**の様になる．この2曲線の交わる点について式 (1) を書きなおした式

$$y = -\frac{x}{\tan\theta} + \frac{\rho}{\sin\theta} \qquad (2)$$

この変換式に交点の座標を代入すると直線に変換される。この線が元の画像の2点を通る式を表すことになり、直線パラメータが抽出できることになる．

この例は2点だったが，画像全体にこの変換を行ない，曲線の重なりを累積することにより，局所的最大点が直線を表すパラメータとなる．

3.2.4 交点の抽出

直線の検出で，始点と終点の座標がわかった線分の集合が得られ，その線分には全てユニークな番号が決められているとする．その線分の集合から，以下に示す，抽出ケース1～ケース4の場合に交点を作成し，その交点にどの線分が接続しているかを決定する．

抽出ケース1 線分 a,b が直角の関係で，それらの交点が存在せず，なおかつ，a の端点と b の端点の距離が一定値以下なら，交点 V(a,b) を作る（**図3**）

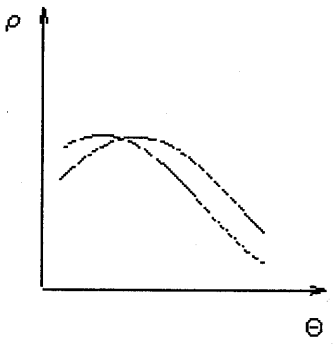

図2 2点を$\theta-\rho$平面で表したときの図

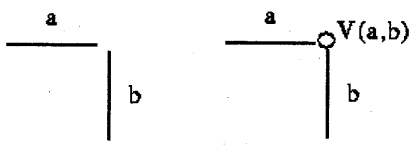

図3 抽出ケース1

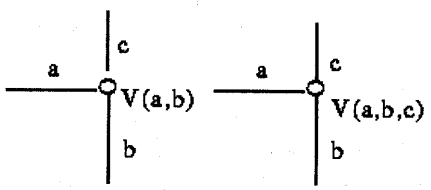

図4 抽出ケース2

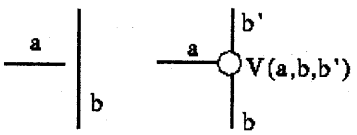

図5 抽出ケース3

抽出ケース2 線分 a と線分 b が既に交点 V(a,b) を持ち，交点 V(a,b) と線分 c の距離が，一定値以下なら，交点に新たに線分を加えて V(a,b,c) にする.（**図4**）

抽出ケース3 線分 b の端点以外のところに線分 a が直角の関係にあり，線分 a の端点と線分 b の距離がある値以下なら，線分 b を線分 a との交点で分断して新しい線分と交点 V(a,b,b') を作る（**図5**）

抽出ケース4 線分 a,b がクロスしている時，交点 V(a,b,a',b') を作る（**図6**）

抽出ケース5 線分の端点が交点を持たない場合，

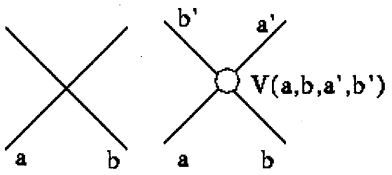

図6　抽出ケース4

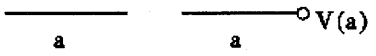

図7　抽出ケース5

交点に接続している線分の数が1の交点として，扱えるので交点として抽出する (図7).

3.2.5　Newick Standard での出力

前節で述べた交点の抽出処理によって，系統樹の接続関係が抽出されたとし，交点を v_i，v_iの次数を $d(v_i)$ と表すことにする.

この接続関係から，Newick Standard 形式に変換するアルゴリズムを図8に示す.

4. 実　　装

4.1　システム構成

ユーザの入力にともなう，内部データの変化を図9に示す. 図中の「外部記憶データ」とは物理的にはファイルのことを指す.「内部記憶データ」のビットマップ情報とは，各ピクセルごとに連結画素の番号などを含んだ，ビットマップ形式のポインタ列である. また集合は配列で実現している.

4.2　データの種類

線分情報を格納する構造体

線分情報には，大きく分けて3つの情報がある.

- 線分の抽出処理によって得られた線分の両端の座標
- この両端の座標から，交点の抽出処理によって得られる線分の両端が接続している交点の番号
- 線分の長さから求められた系統樹の枝長. これは，Newick Standard 形式に変換される時に用いられる

交点情報を格納する構造体

交点情報には，大きく分けて3つの情報がある.

- 交点の抽出処理によって得られる交点の座標
- 交点の抽出処理によって得られる交点に接続している線分の情報. 線分の数と，番号を配列出格納している
- 葉節点の場合，葉ラベルをもつので，その文字列へのポインタ

4.3　前　処　理

前処理は，細線化と，系統樹とスケールバー以外の画素の消去，の2つの処理に分けられる. Hidich の

procedure printNewick(v_i)
　v_iに接続している，各々の線分に対して，
　v_i以外の交点の集合を v_Lとする.
　'(' を出力
　foreach $v_l \in V_L$
　　if ($d(v_l) == 1$)
　　　葉ラベルを出力
　　　':' と枝長を出力
　　else
　　　printNewick1(v_i, v_l)
　　$V_L = V_L - \{v_l\}$
　　if ($V_L! = \phi$)
　　　',' を出力
　');' を出力

procedure printNewick1(v_p, v_i)
　v_iに接続している，各々の線分に対して，
　v_i, v_p以外の交点の集合を v_Lとする.
　'(' を出力
　foreach $v_l \in V_L$
　　if ($d(v_l) == 1$)
　　　葉ラベルを出力
　　　':' と枝長を出力
　　else
　　　printNewick1(v_i, v_l)
　　$V_L = V_L - \{v_l\}$
　　if ($V_L! = \phi$)
　　　',' を出力
　')' を出力
　':' と枝長を出力

図8　Newick Standard 形式に変換するアルゴリズム

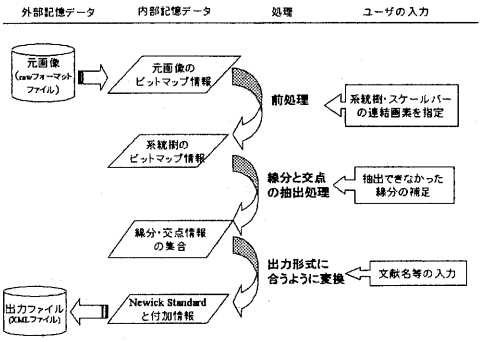

図9　処理の流れ

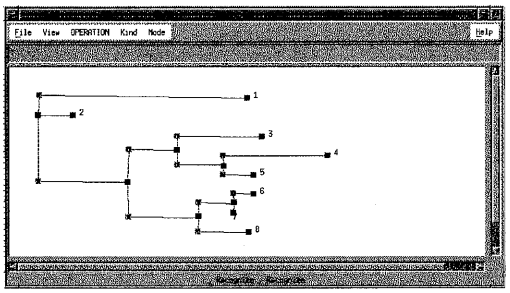

図 10　交点を抽出した時の画像

細線化[5]を行ない画像上の線を 1 画素の幅にする。その後、系統樹とスケールバー以外の不要な画素を消去する。

4.4　線分の検出

線分の検出は 3 段階に分けられる．まず，画像中のすべての画素にたいして式 (1) の変換式を適用し．曲線の重なりを累積することで，2 次元配列のヒストグラムを作る．

次に，ある閾値より大きい値を直線のパラメータとして格納する．最後に，検出された直線の始点と終点を決めるために，ビットマップ画像を直線にそってなぞりながら，画素と直線が重なる領域を決定する．

この際に，閾値を高い値を設定すると，長い直線しか抽出できなくなり，逆に低い値を設定すると，長い直線の「山」に短い直線の「山」が重なり，短い直線が抽出できなくなる．このため，閾値を下げながら，直線が検出されなくなるまで，3 つの処理を繰り返すことで，短い直線を検出することできるようにした．

4.5　交点の検出

交点の検出の際に，最も重要なパラメータが交点を検索する範囲である．このパラメータが小さいと間違う可能性は小さくなるが，交点が抽出できない可能性が大きくなる．逆に，パラメータを大きくすると，交点を抽出できない可能性は小さくなるが，交点はない所を交点として検出する可能性が大きくなる．実装では，経験的に最もうまくいく値を設定しているが，画像の質によってはこの値を大きくする必要があるので，値を変更できるようにした．

図 10 は交点を抽出したときの画面である．四角は抽出した交点を表す．また，数字は葉ラベルを対話的に入力するためにつけた葉節点の仮の ID である．

5.　結果とその考察

実験方法

Jounal of Molecular Evolution Vol. 46 に掲載されている系統樹 55 個について，正しく Newick Standard 形式に変換されるかどうかを実験した．特に，線分と交点の抽出処理では，検出できなかった線分を自動修正だけですべて修正できたものと，自動修正だけでは修正できなかったものに分けた．

表 1　抽出処理の実験結果

全体 55			
変換に成功 51			エラー 4
線分の修正なし 23	自動で修正 5	手動で修正 22	

表 2　補足操作の回数

線分を修正して抽出できたもの 11			
1 回 ～ 5 回	6 回 ～ 10 回	11 回 ～ 15 回	16 回以上
13	4	3	2

表 3　線分の修正回数が多かったものの内訳

修正した回数	線分数	葉節点数
11	160	49
13	163	42
15	51	16
17	131	38
31	128	36

結果

表 1 中の数字は系統樹の数を表す．たとえば，実験をした系統樹が 55 個で，そのうち，正しく変換されたものが 51 個，途中の処理でエラーを起こして処理が止まったものが 4 個であることを表す．

抽出できなかった線分を追加した回数と，誤って抽出した線分を削除した回数を足したものを修正操作の回数として数えた．修正操作の回数を 1 回から 5 回，6 回から 10 回，11 回から 15 回，それ以上，に分け，**表 2** にまとめた．

考察

- 手動の修正回数について

特に修正回数が多かった 5 つについて，その葉節点節点数と線分数を**表 3** に示す．

修正回数の多いものは，線分の数も多く，著しく抽出率が悪い訳ではないことが分かった．しかし，画像の質が悪いものにおいては，葉節点の数が平均と同じでも，修正回数が多いものがあった．

- エラーについて

Hough 変換を行なう際に，線分の角度を 1 度単位で検出しているので，長い線分は 1 度単位では表せきれなくなる．その結果，1 個の線分が 2 個（あるいはそれ以上）に分断される．これを，解決するために，Hough 変換で抽出された線分のうち，傾きと両端の位置を比較して，同一と思われる線分を検索し，1 個の線分にまとめるという処理がある．

- 修正が必要な場合
  - 平行な線分の間がせますぎて，同一の線分線として抽出される場合．

- 線分が短すぎて抽出できなく，自動修正もできなかった場合．
- 線分が途切れ途切れになっていて，その間隔が大きく，別の線分として抽出された場合．

6. おわりに

分子系統樹データベースに蓄積するデータを作成するために，文献中の系統樹を自動で認識しデータベースに格納できる形に変換する方法について提案した．

提案した方法は，手入力でのデータベース化と比較して，特に大規模な系統樹の場合に有用であるということが分かった．しかし，多くの自動認識システムにおいてもそうだが，本研究で作成した認識プログラムにおいても最終的には人による補正，確認が必要となる．そのため，実際に使いやすいプログラムになるためには，自動認識の性能と同時にユーザインターフェースも重要な役割を果たす．本研究において作成したプログラムはこの2点に重きをおいて作成した．

今後は，認識精度を向上させ線分の抽出率をあげることにより，さらに使いやすいプログラムを目指す予定である．

参考文献

1) UCMP Exhibit Halls: Phylogeny, In The Museum of Palenotology, University of California,Berkeley, Internet address: http://www.ucmp.berkeley.edu/exhibit/phylogeny.html
2) Maidak, B.L. et al: A New Version of the RDP (Ribosomal Database Project), *Nucleic Acids Research*, Vol. 27, No. 1, pp. 171–173 (1999). Internet address: http://www.cme.msu.edu/RDP/analyses.html
3) Maddison, D. R. and Maddison, W. P.: The Tree of Life: A Multi-Authored, Distributed Internet Project Containing Information about Phylogeny and Bildiversity (1998). Internet address: http://phylogeny.arizona.edu/tree/phylogeny.html
4) JUNGLE : Phylogenetic Tree Database, Internet address: http://smiler.lab.nig.ac.jp/jungle/jungle.html
5) Hilditch, C. J.: Linear Skeletons from Square Cupboards, *Machine Intelligence*, 4 (Meltzer, B.(ed.)), University Press, Edinburgh, pp. 403–420 (1969).
6) Hough, P.: Method and Means for Recognizing Complex Patterns, U. S. Patent 3,069,645 (1962).