近畿大学・農学部・生命情報学

マルチプルアライメントと分子系統学基礎

2007年5月8日 (火)

奈良先端大・情報・蛋白質機能予測学講座
川端 猛
takawaba@is.naist.jp

http://isw3.naist.jp/IS/Kawabata-lab/home-ja.html

マルチプルアライメント

(multiple sequence alignment
多重配列整列)

マルチプルアライメント (多重配列整列) とは

3本以上の配列を進化的な対応関係に従って並べること

```
>1nshA
SRPTETERCIESLIAVFQKYAGKDGHSVTLSKTEFLSFMNTELAAFTKNQKDPGVLDRMMKKLDLNSDGQLDFQEFL
NLIGGLAVAESFVKAAPPQKRF
>1j55A
MTELETAMGMIIDVFSRYSGSEGSTQTLTKGELKVLMEKELPGFLDAVDKLLKDLDANGDAQVDFSEFIVFVAAITS
ACHKYFEKAL
>1ig5A
KSPEELKGIFEKYAAKEGDPNQLSKEELKLLLQTEFPSLLKGPSTLDELFEELDKNGDGEVSFEEFQVLVKKISQ
>1qx2A
MKSPEEIKGAFEVFAAKEGDPNQISKEELKLVMQTLGPSLLKGMSTLDEMIEEVDKNGDGEVSFEEFLVMMKKISQ
```

```
CLUSTAL W (1.83) multiple sequence alignment

1nshA      SRPTETERCIESLIAVFQKYAGKDGHSVTLSKTEFLSFMNTELAAFTKNQKDPGVLDRMM
1j55A      --MTELETAMGMIIDVFSRYSGSEGSTQTLTKGELKVLMEKELPGFLD------AVDKLL
1ig5A      -------KSPEELKGIFEKYAAKEGDPNQLSKEELKLLLQTEFPSLLKG---PSTLDELF
1qx2A      ------MKSPEEIKGAFEVFAAKEGDPNQISKEELKLVMQTLGPSLLKG---MSTLDEMI
                  .   :   *. ::..:* .  ::* *:  .::.  ..: .      .:*.::

1nshA      KKLDLNSDGQLDFQEFLNLIGGLAVACHESFVKAAPPQKRF
1j55A      KDLDANGDAQVDFSEFIVFVAAITSACHKYFEKAGL-----
1ig5A      EELDKNGDGEVSFEEFQVLVKKISQ----------------
1qx2A      EEVDKNGDGEVSFEEFLVMMKKISQ----------------
           :.:* *.*.::.*.** ::  ::
```

マルチプルアライメントの目的

```
1nshA      SRPTETERCIESLIAVFQKYAGKDGHSVTLSKTEFLSFMNTELAAFTKNQKDPGVLDRMM
1j55A      --MTELETAMGMIIDVFSRYSGSEGSTQTLTKGELKVLMEKELPGFLD------AVDKLL
1ig5A      -------KSPEELKGIFEKYAAKEGDPNQLSKEELKLLLQTEFPSLLKG---PSTLDELF
1qx2A      ------MKSPEEIKGAFEVFAAKEGDPNQISKEELKLVMQTLGPSLLKG---MSTLDEMI
                  .   :   *. ::..:* .  ::* *:  .::.  ..: .      .:*.::
```

- **ファミリ内の機能的重要部位の検出**
- **ファミリを特徴付けるモチーフの発見**
- **プロフィール法による遠縁のホモログ発見**
- **分子系統解析の第一ステップとして不可欠**
- **進化的追跡法**(evolutionary trace method)

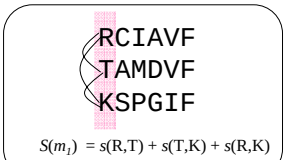
多重整列のスコア

(1) SP (sum-of-pairs)**スコア**

複数の文字列間のスコアを
ペアワイズのアミノ酸置換スコア$s(a,b)$**の和で表す**

$$S(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

m_i^k : k **番目の配列 の** i**番目の文字**



理論的にはおかしい: $S(A,B)+S(B,C)+S(A,C) = \log\frac{P(A,B)P(B,C)P(A,C)}{P(A)^2P(B)^2P(C)^2} \neq \log\frac{P(A,B,C)}{P(A)P(B)P(C)}$

```
#  BLOSUM62
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
A  4 -1 -2 -2  0 -1 -1  0 -2 -1 -1 -1 -1 -2 -1  1  0 -3 -2  0 -2 -1  0 -4
R -1  5  0 -2 -3  1  0 -2  0 -3 -2  2 -1 -3 -2 -1 -1 -3 -2 -3 -1  0 -1 -4
N -2  0  6  1 -3  0  0  0  1 -3 -3  0 -2 -3 -2  1  0 -4 -2 -3  3  0 -1 -4
D -2 -2  1  6 -3  0  2 -1 -1 -3 -4 -1 -3 -3 -1  0 -1 -4 -3 -3  4  1 -1 -4
C  0 -3 -3 -3  9 -3 -4 -3 -3 -1 -1 -3 -1 -2 -3 -1 -1 -2 -2 -1 -3 -3 -2 -4
Q -1  1  0  0 -3  5  2 -2  0 -3 -2  1  0 -3 -1  0 -1 -2 -1 -2  0  3 -1 -4
E -1  0  0  2 -4  2  5 -2  0 -3 -3  1 -2 -3 -1  0 -1 -3 -2 -2  1  4 -1 -4
G  0 -2  0 -1 -3 -2 -2  6 -2 -4 -4 -2 -3 -3 -2  0 -2 -2 -3 -3 -1 -2 -1 -4
H -2  0  1 -1 -3  0  0 -2  8 -3 -3 -1 -2 -1 -2 -1 -2 -2  2 -3  0  0 -1 -4
I -1 -3 -3 -3 -1 -3 -3 -4 -3  4  2 -3  1  0 -3 -2 -1 -3 -1  3 -3 -3 -1 -4
L -1 -2 -3 -4 -1 -2 -3 -4 -3  2  4 -2  2  0 -3 -2 -1 -2 -1  1 -4 -3 -1 -4
K -1  2  0 -1 -3  1  1 -2 -1 -3 -2  5 -1 -3 -1  0 -1 -3 -2 -2  0  1 -1 -4
M -1 -1 -2 -3 -1  0 -2 -3 -2  1  2 -1  5  0 -2 -1 -1 -1 -1  1 -3 -1 -1 -4
F -2 -3 -3 -3 -2 -3 -3 -3 -1  0  0 -3  0  6 -4 -2 -2  1  3 -1 -3 -3 -1 -4
P -1 -2 -2 -1 -3 -1 -1 -2 -2 -3 -3 -1 -2 -4  7 -1 -1 -4 -3 -2 -2 -1 -2 -4
S  1 -1  1  0 -1  0  0  0 -1 -2 -2  0 -1 -2 -1  4  1 -3 -2 -2  0  0  0 -4
T  0 -1  0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1  1  5 -2 -2  0 -1 -1  0 -4
W -3 -3 -4 -4 -2 -2 -3 -2 -2 -3 -2 -3 -1  1 -4 -3 -2 11  2 -3 -4 -3 -2 -4
Y -2 -2 -2 -3 -2 -1 -2 -3  2 -1 -1 -2 -1  3 -3 -2 -2  2  7 -1 -3 -2 -1 -4
V  0 -3 -3 -3 -1 -2 -2 -3 -3  3  1 -2  1 -1 -2 -2  0 -3 -1  4 -3 -2 -1 -4
B -2 -1  3  4 -3  0  1 -1  0 -3 -4  0 -3 -3 -2  0 -1 -4 -3 -3  4  1 -1 -4
Z -1  0  0  1 -3  3  4 -2  0 -3 -3  1 -1 -3 -1  0 -1 -3 -2 -2  1  4 -1 -4
X  0 -1 -1 -1 -2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -2  0  0 -2 -1 -1 -1 -1 -1 -4
* -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4  1
```

多重配列のスコア（続き）

（2）配列への重み付きのSum-of-pair関数 (ClustalW)

$$S(m_i) = \sum_{k<l} w_k \cdot w_l \cdot s(m_i^k, m_i^l)$$

w_k

0.1 LGVLF
0.1 LGILF
0.3 LAALF
0.5 LAAAL

（3）エントロピー関数の最小化

各サイトのアミノ酸の頻度$p_i(a)$を推定し、そのエントロピーの和を求める

$$S(m_i) = -\sum_a p_i(a) \log p_i(a)$$

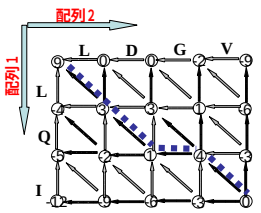
12345
LGVLF
LGILF
LAALF
LAAAL

サイト	Pi(a)	$S(m_i)$
1	$P_1(L)=1.0$,	0.00
2	$P_2(G)=0.5$,$P_2(A)=0.5$	0.69
3	$P_3(V)=0.25$, $P_3(I)=0.25$, $P_3(A)=0.5$	1.04

（4）対アライメントライブラリの重複による部位特異的スコア (T-COFFEE)

どうやって並べるか？
多次元DPによる多重配列の厳密解

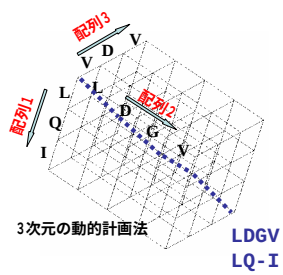
2本の配列のアライメント



2次元の動的計画法 LDGV / LQ-I

メモリ・計算時間 $O(L^2)$

3本の配列のアライメント



3次元の動的計画法 LDGV / LQ-I / VD-V

メモリ・計算時間 $O(L^3)$

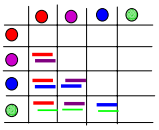
N本の配列のアライメントのメモリ・計算時間は$O(L^N)$→非現実的

長さ100の2本のアライメントが1秒でできても、10本に増やすと100^8秒かかる。

プログレッシブ・アライメント
(progressive alignment, 累進法)

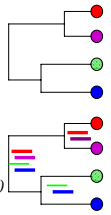
Feng and Doolittle (1987)

（1）全ての配列ペアのペアワイズアライメントを計算する



（2）ペアワイズアライメントによる距離行列を計算し、樹形図を計算する。

（3）樹形図の葉から、ペアワイズアライメントを組み上げていく



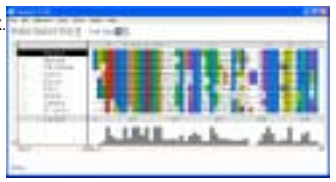
ステップ1に最も計算時間がかかる。全体の計算量はほぼ$O(NL^2)$

ClustalW / ClustalX

UNIX/Mac版 ftp://ftp.ebi.ac.uk/pub/software/unix/clustalw
Windows版 ftp://ftp.ebi.ac.uk/pub/software/dos/clustalx
WEBサーバ：http://www.ebi.ac.uk/clustalw

・現在、最も一般的な多重整列のプログラム
・アルゴリズムは累進法。ペアワイズアライメントはグローバルアライメントを用い、ガイド木はNJ法で 作成。スコアは配列の重みを導入したSum-of-pairs。置換スコア行列の選択、ギャップペナルティ等に様々な経験的な工夫が見られる。

・CUI版はClustalW, GUI版はClustalX. UNIX, Windows, MACでも動作する。



・NJ法による系統樹作成機能付き。

Thompson, J.D., Higgins, D.G., Gibson T.J. "CLUSTALW : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice". Nucleic Acids Reseach, 1994, 22, 4673-4680.

T-COFFEE

http://igs-server.cnrs-rs.fr/~cnotred/Projects_home_page/t_coffee_home_page.html

アルゴリズム

(1)対アライメントのライブラリを作成する
・グローバルアライメントとローカルアライメントの両方を用いる
・それぞれの対アライメントの重複性から、対アライメントライブラリの重みを計算
・3つ以上の対アライメントを組み合わせて、新しい対アライメントを作成
(2)これらの対アライメントから、位置特異的スコア行列を作成
(3)累進法で、多重アライメントを作成。

・様々な手法で、ペアワイズアライメント群を作成し、それらの重複性からスコア行列を作成しようとするアイデア。

・最終的な出力はグローバルアライメントだが、ローカルアライメントも考慮される。

・計算時間はClustalWの2〜3倍かかるが、アライメントの精度は高いとされる。

Notredame, C., Higgins, D., Heringa,J. "T-Coffee: A novel method for multiple sequence alignments". *J. Mol.Biol.* (2000), Vol 302, 205-217

マルチプルアライメントを行う上での注意点

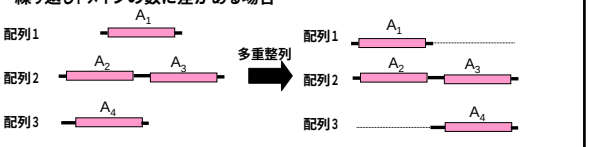
（1）対象とする配列群が相同であることの確認
・他と全く似ていない配列が混入していると意味のない比較になる

（2）対象とする配列群のほぼ全長どうしが対応することの確認
・ClustalW等主要な多重整列プログラムはグローバルアライメントなので、全長どうしが対応することがアルゴリズムの前提
・マルチドメイン構造、繰り返し構造になっていないかをチェック
・そもそも、配列長が著しく異なる場合は、ほぼ間違いなく問題が生じる
・配列の一部しか、対応しないなら、その部分だけ切り出して入力する

（3）計算されたマルチプルアライメントの結果の吟味
・既知の機能部位がきちんと保存されているか
・長すぎるギャップはないか（マルチドメインの可能性）
・保存部位が、非保存の配列はないか（ホモログでない可能性）
・立体構造が既知のものが含まれているなら、立体構造アライメントも参照

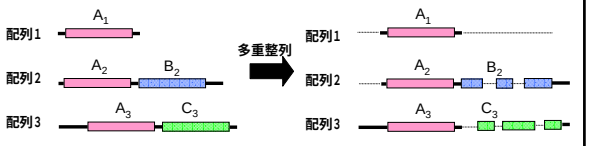
マルチドメインのときのアライメントの問題点

繰り返しドメインの数に差がある場合



全ての配列が並ぶサイトがない！

全く異なるドメインが接続されている場合



おかしなアライメント！

マルチプルアライメントから何を読み取るか？

```
5p21-  MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSY
1ctqA  MTEYKLVVVGAGGVGKSALTIQLIQNHFVDEYDPTIEDSY
1c1yA  MREYKLVVLGSGGVGKSALTVQFVQGIFVEKYDPTIEDSY
1kao-  MREYKVVVLGSGGVGKSALTVQFVTGTFIEKYDPTIEDFY
1huqA  --QFKLVLLGESAVGKSSLVLRFVKGQFHEYQESTIGAAF
1g16A  ----KILLIGDSGVGKSCLLVRFVE---DKFNPI--DFK
1ek0A  VTSIKLVLLGEAAVGKSSIVLRFVSNDFAENKEPTIGAAF
3rabA  ---FKILIIGNSSVGKTSFLFRYADDSFTPAFVSTVGIDF
1mh1-  ----KCVVVGDGAVGKTCLLISYTTNAFPGEYIPTVFDNY
2ngrA  MQTIKCVVVGDGAVGKTCLLISYTTNKFPSEYVPTVFDNY
1tx4B  ----KLVIVGDGACGKTCLLIVNSKDQF---YVPTVFENY
1i2mA  --QFKLVLVGDGGTGKTTFVKRHLKKYVATEVHPLVFHTN
1d5cA  --KYKLVFLGEQAVGKTSI-ITRFYDTFDNNYQSTIGDFL
```

サイトごとに保存の度合いに差がある。
サイトごとにアミノ酸の出現傾向に差がある

[AG]-x(4)-G-K-[ST]

モチーフ解析

- 正規表現風のパターンで、局所的な配列のパターンを表現。
  PROSITE(http://www.expasy.ch/prosite/)が有名

1.進化的に保存している局所配列パターン
- ・マルチプルアライメント由来
- ・保存しているサイト→機能的に重要なサイト→活性部位

2.機能的な局所配列パターン
- ・リン酸化サイト、N-ミリストイル化サイトなど

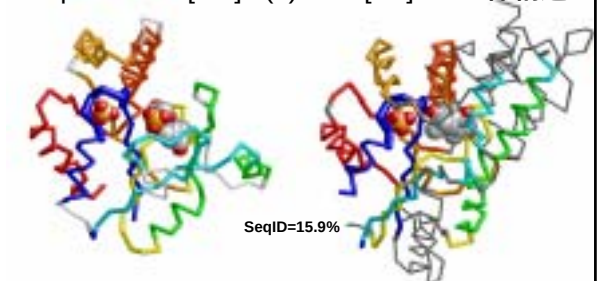
PROSITEのモチーフの記述法

(例) ATP_GTP_A :
[AG]-x(4)-G-K-[ST]

2FE2S FERREDOXIN:
C-{C}-{C}-[GA]-{C}-C-[GAST]-{CPDEKRHFYW}-C

x	:任意のアミノ酸
x(n)	:n個の任意のアミノ酸
x(n,m)	:nからm個の任意のアミノ酸
[ACD]	:AかCかDのいずれかのアミノ酸
{ACD}	:AでもCでもDでもないアミノ酸

P-loopモチーフ: [AG]-x(4)-G-K-[ST] の立体構造



1gky:Guanilate Kinase
(8-15:GPSGTGKS)

1e2kA:Thymidine Kinase
(56-63:GPHGMGKT)

- ・P-loopモチーフは、ヌクレオチドのリン酸基結合サイトに対応
- ・モチーフ以外の領域も、立体構造は似ている

ProSiteモチーフの問題点

False positiveが多く、ファミリの認識能力は高くない。

[AG]-x(4)-G-K-[ST]

```
5p21-  MTEYKLVVVGAGGVGKSAL
1ctqA  MTEYKLVVVGAGGVGKSAL
1c1yA  MREYKLVVLGSGGVGKSAL
1kao-  MREYKVVVLGSGGVGKSAL
1huqA  --QFKLVLLGESAVGKSSL
1g16A  ----KILLIGDSGVGKSCL
1ek0A  VTSIKLVLLGEAAVGKSSI
3rabA  ---FKILIIGNSSVGKTSF
1mh1-  ----KCVVVGDGAVGKTCL
2ngrA  MQTIKCVVVGDGAVGKTCL
1tx4B  ----KLVIVGDGACGKTCL
1i2mA  --QFKLVLVGDGGTGKTTF
2efgA  -RLRNIGIAAHIDAGKTTT
```

1. パターンの表現能力の限界
2. 客観的にパターンを生成するのが難しい。
3. もっと大域的な領域も淡く似ているはず

プロフィール法

マルチプルアライメントから<u>サイトごとのスコア行列</u>を作成。これに対して動的計画法等を用いて配列をアライメント。

サイトごとのスコア行列
↓
プロフィール(Profile)
PSSM(Position Specific Score Matrix)

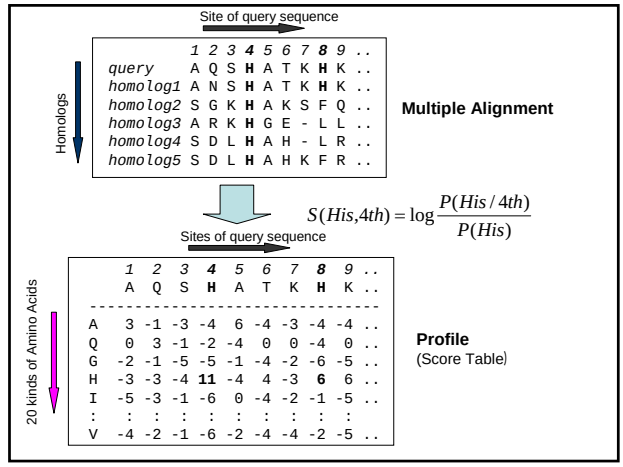
	1	2	3	4	5	6	..
A	3	-1	-3	-4	6	-4	..
Q	0	3	-1	-2	-4	0	..
H	-3	-3	-4	**11**	-4	4	..
:	:	:	:	:	:	:	
V	-4	-2	-1	-6	-2	-4	..

HMMer

マルチプルアライメントを入力とする。隠れマルコフモデル(HMM)を使用しているため、表現力はPSI-BLASTより高いはずだが、計算速度は遅い。Pfamは HMMerを採用している。

PSI-BLAST

BLASTの拡張版。反復的にデータベース検索を行うことで、厚いマルチプルアライメントを生成する。



Pfam : 蛋白質ファミリのデータベース

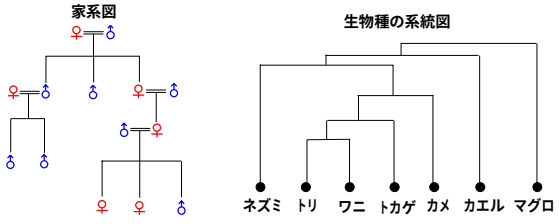
http://www.sanger.ac.uk/Software/Pfam/

各蛋白質のファミリのHMMのプロフィール、マルチプルアライメントを集めたデータベース

分子系統学基礎

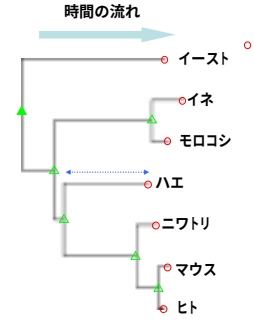
系統樹(phylogenetic tree)

対象物が生成される過程（歴史、進化史）を木構造で示したもの



- 何を対象にするかはいろいろ（個体、生物種、染色体、遺伝子）
- 「系統樹を書く」 → 「過去（歴史）を推定する」
- 「分類」（似ているものをまとめること）と「系統推定」の手続きは似ている
- 様々な「分類法」が在り得るが、「系統樹」には唯一つの歴史的真実があるはず。

系統樹の用語



トリオースリン酸異性化酵素のアミノ酸配列の分子系統樹

○ **葉(leaf)**. 現在観察される対象が位置するノード。対象のことを**OTU**（Operational Taxonomy Unit)と呼ぶ。個体、生物種、染色体、遺伝子、蛋白質、ドメインなど何でもよい。

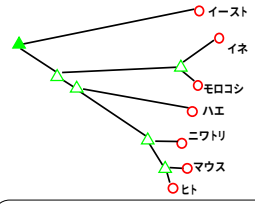
△ **祖先ノード(ancestral node)**。2つの枝が交わる点。その下にあるOTUの共通祖先を示す。

▲ **ルート、根 (root)**。木の中で最も過去にあるノードのこと。

枝長(branch length)。進化距離(evolutionary distance)に比例して書かれる。

枝長を無視したノードと枝の接続関係のことを**トポロジー(topology)**という。

系統樹(二分岐樹)のデータ構造



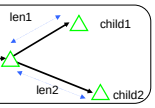
ノード(node)と枝(branch)からなるグラフ

- ノードには葉(leaf)ノードと祖先ノード(ancestor)ノードの2種がある。
- 祖先ノード(ancestor)ノードから2つの子孫ノードへ枝が引かれる
- 葉(leaf)ノードは、子孫ノードを持たない。
- ルートノードは、親ノードを持たない。

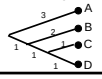
各ノードが、2つの子ノードへのポインタと、枝長を持つ。

```
struct NODE{
 struct NODE *child1,*child2;
 double len1, len2;};
```

ルートノードからスタートして再帰呼び出しすれば全ノードをスキャンできる。



- Newick(New Hampshire)フォーマット:系統樹を括弧やカンマで記述

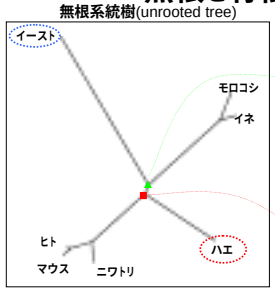


枝長なし `(A,(B,(C,D)));`

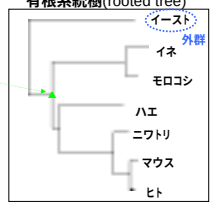
枝長つき `(A:3,(B:2,(C:1,D:1):1):1);`

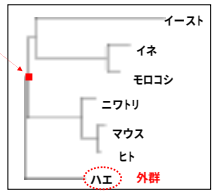
無根と有根の系統樹

無根系統樹(unrooted tree)



有根系統樹(rooted tree)

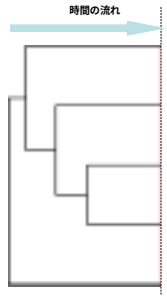




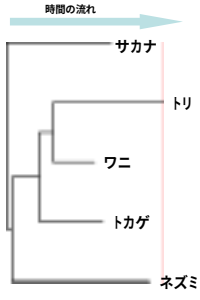
- NJ法等のアルゴリズムは、根を指定しない無根系統樹を生成する
- どの枝に根を置くかによって、様々な有根系統樹が生成可能。
- 根は適当な**外群(out group)**の選択で決める。
  外群:他の全てのOTUと十分遠いと考えられるOTU

進化速度の同一を仮定する場合・しない場合

進化速度 =[進化距離] / [時間]



進化速度が一定の場合
(UPGMA法で作成)
全てのOTU(葉ノード)が一列に揃う

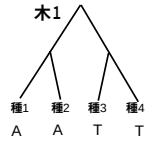


進化速度が一定でない場合
(NJ法で作成)
OTU(葉ノード)は一列に揃わない

分子配列からの系統樹の推定法

方法	解析方法	出力する木	計算速度	特徴
最節約法	サイト(特徴)単位	有根	遅い	アイデアは単純。分子データ以外の質的特徴にも適用可能
UPGMA法	距離行列	有根	速い	分子速度の一定性を仮定。重心間距離のクラスター解析と等価。
近隣結合法	距離行列	無根	速い	最小進化の法則を距離行列に適応。分子速度の一定性を仮定しない。
最尤法	サイト単位	有根	遅い	分子進化の確率モデルに従う。数学的な厳密さは高い。

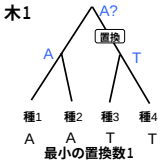
最節約法(maximum parsimony)

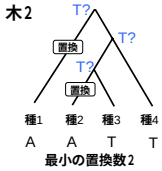


4つの生物種のある1つのサイトのDNA配列がわかったとする。

どちらの木が尤もらしいか?

(1) 総置換数が最小になるように、祖先形質を推定
(2) 総置換数が最小の木が尤もらしいとする





木1のほうが、置換数が少ない
→木1のほうが木2より尤もらしい

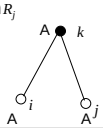
最節約の考え(最小進化の法則)
現在の生物の形質を表現する仮説(系統樹)の中で、進化による変化の回数が最も少ない仮説が正しい。

最小進化の法則(minimum evolution principle)、オッカムの剃刀(Ockham's razor)

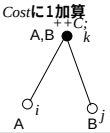
最節約法のアルゴリズム(traditional parsimony)

[初期化]
$Cost=0$, $k=2n-1$(ルートノード)
[再帰的実行]
kが葉ノードなら、
$R_k = x_k$
kが葉ノードでないなら、i,jをkの子ノードとすると、
子ノードのR_i, R_jが計算されていないなら、
R_i, R_jを計算(再帰呼び出し)。
計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, $Cost$に1加算
[終了処理]
$Cost$が最小コスト

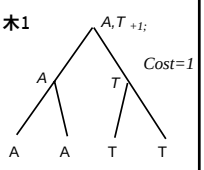
$R_i \cap R_j$が空でないなら、
$R_k=R_i \cap R_j$



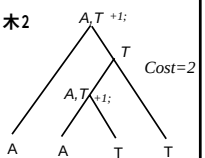
$R_i \cap R_j$が空なら、
$R_k=R_i \cup R_j$, $Cost$に1加算



木1



木2



最節約法のアルゴリズムのキーポイント

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

「∩」、「∪」、「空である」 ：などは集合の専門用語

A∩B： 積集合。共通部分。２つの集合A,Bの共通要素

例 (a,b,c)∩(b,c,d) = (b,c), (a,b,c)∩(a) =(a), (a)∩(b)=空

A∪B： 和集合。合併集合。２つの集合A,Bのどちらかに属する要素

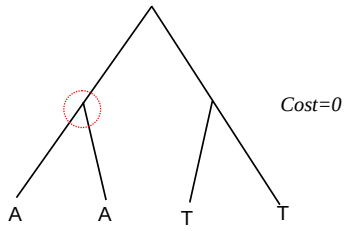
例 (a,b,c)∩(b,c,d) = (a,b,c,d), (a,b,c)∩(a) =(a,b,c,d), (a)∩(b)=(a,b)

Aが空である： 集合Aに属する要素が一つもないこと。

置換数の推定の例:木1(1)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

木1

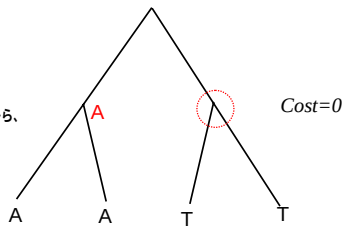


置換数の推定の例:木1(2)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

木1

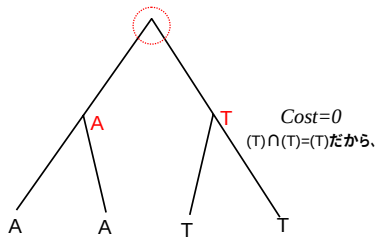
(A)∩(A)=(A)だから、



置換数の推定の例:木1(3)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

木1



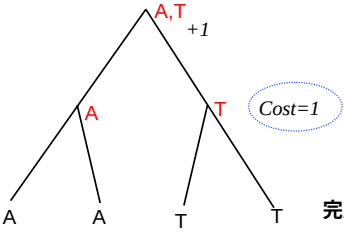
(T)∩(T)=(T)だから、

置換数の推定の例:木1(4)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

(A)∩(T)=空だから、(A)∪(T)=(A,T)を祖先形質とする。コストを1増やす

木1

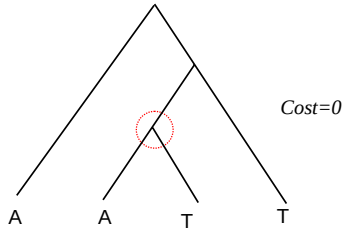


完成！

置換数の推定の例:木2(1)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

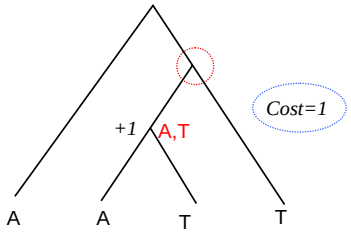
木2



置換数の推定の例:木2(2)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

木2

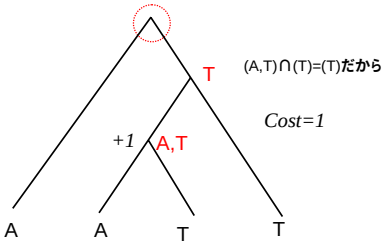


(A)∩(T)=空だから、(A)∪(T)=(A,T)を祖先形質とする。コストを1増やす

置換数の推定の例:木2(3)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

木2

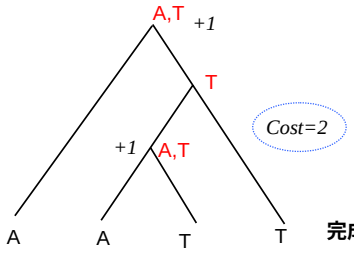


置換数の推定の例:木2(4)

子ノードのR_i, R_jが計算されているなら、以下のようにR_kを計算
$R_i \cap R_j$が空でないなら、$R_k=R_i \cap R_j$
$R_i \cap R_j$が空なら、$R_k=R_i \cup R_j$, *Cost*に1加算

(A)∩(T)=空だから、(A)∪(T)=(A,T)を祖先形質とする。コストを1増やす。

木2



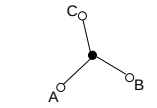
完成！

可能な木のトポロジーの数

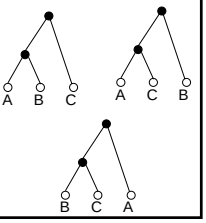
$\prod_{k=3}^{N}(2k-5)$　　$\prod_{k=3}^{N}(2k-3)$

OTU数 N	無根系統樹	有根系統樹
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10395
8	10395	135135
9	135135	2027025
10	2027025	34459425

N=3の場合の無根系統樹のトポロジー

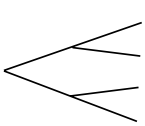


N=3の場合の有根系統樹のトポロジー



最節約法の特徴

- 分子データに限らず、様々な形質に対して適用可能
  骨、化石など生物の形態から系統樹を推定する唯一の方法
- 「最節約 / 最小進化」という考え方は、全ての系統推定の基本
- 各特徴が独立・無相関であることが前提
- 配列・特徴の数が増えた場合、膨大な計算時間が必要となる
  祖先形質の推定が必要。トポロジー探索は全回探索が基本。
- 原則として枝長の推定はできない
- 多重置換等、複雑な進化のモデルを扱えない



	塩基配列				羽毛	二足歩行	心臓	体温
種1	A	G	G	G	ない	不可能	1心房1心室	変温
種2	A	G	A	A	ない	不可能	2心房1心室	変温
種3	T	G	A	A	ない	不可能	2心房2心室	変温
種4	T	A	G	A	ある	可能	2心房2心室	恒温

距離行列法

なんらかの方法でOTU間の距離(進化距離)を定義し、距離行列を作成。
その距離をできるだけ満たすような木を計算する方法

アライメント

配列 1 AAAAA
配列 2 AAAAT
配列 3 TAATA
配列 4 TAATT

距離行列 d_{ij}
(不一致サイト数)

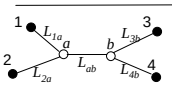
	1	2	3	4
1	0	1	2	3
2	1	0	2	2
3	2	2	0	1
4	3	2	1	0

とか

距離行列 d_{ij}
(p距離)

	1	2	3	4
1	0.0	0.2	0.4	0.6
2	0.2	0.0	0.4	0.4
3	0.4	0.4	0.0	0.2
4	0.6	0.4	0.2	0.0

$$\text{p距離} = \frac{\text{[不一致のサイト数]}}{\text{[比較したサイト数]}}$$



d_{12} $L_{1a}+L_{2a}$　d_{34} $L_{3b}+L_{4b}$
d_{13} $L_{1a}+L_{ab}+L_{3b}$　d_{14} $L_{1a}+L_{ab}+L_{4b}$
d_{23} $L_{2a}+L_{ab}+L_{3b}$　d_{24} $L_{2a}+L_{ab}+L_{4b}$

木の枝長の和が距離行列の値になるように木のトポロジーと枝長を推定

配列データからの進化距離の推定

進化距離：1サイトあたりに受けた置換の回数

分子時計：
DNAやアミノ酸配列の違いが生じる速度（進化速度）は近似的に一定であること。

分子進化の中立説（木村資生、1968）
DNAやアミノ酸配列が進化の過程で受ける変異の*ほとんど*は、
自然選択の上からは、よくも悪くもない“中立的”なものであるという仮説。

***p*-距離** ：最も単純な進化距離の推定法

$$p\text{-距離} = n_d / n$$

n：比較したサイトの数
n_d：配列が異なっていたサイトの数

GAALSTLLS
GGVVSTLVA

p-距離 = 4 / 10 = 0.4

多重置換の影響を考慮した距離

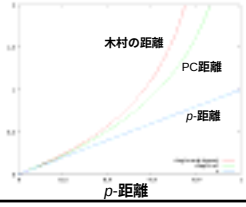
		p-距離
0	AAAAAAAAAA	0.0
1	AKAAAAAAAA	0.1
2	PKAAAAAAAA	0.2
3	PKAAMAAAAA	0.3
4	PKAAMAIAAA	0.4
5	PKAAMAIARA	0.5
6	PKAAMADARA	0.5
7	PKAAMADARR	0.6
8	PKAAMADATR	0.6
9	PKAAMADRTR	0.7
10	PKAANADRTR	0.7
11	PKAANADWTR	0.7
12	PKVANADWTR	0.8
13	PKVAAADWTR	0.7
14	NKVAAADWTR	0.7

階層

多重置換 ：進化時間が長いときに、同じサイトに複数回の置換が起こること。

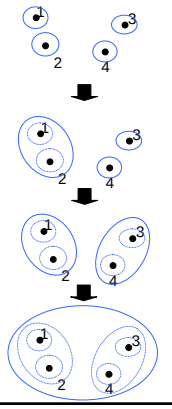
PC距離 （Poisson Correction ）= $-\log(1-p)$

木村の距離 = $-\log(1 - p - 0.2p^2)$



UPGMA法

Unweighted Pair-Group Method with Arithmetric mean



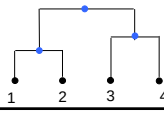
[初期化]
全ての配列間の距離d_{ij}を計算。それぞれの配列iが一つのクラスタC_iを構成するとする。

[反復]
(1) 全てのクラスタのペアの中で距離d_{ij}が最小のペアC_iとC_jを選び、融合して新しいクラスタ$C_k = C_i \cup C_j$を作る。このとき、C_iとC_jを子にもつ親ノードを枝長の高さ が$d_{ij}/2$ になるように作る

(2) 距離行列を更新する。クラスタ間の距離は、属する配列間の平均距離で定義する。

$$d_{ij} = \frac{1}{|C_i \| C_j|} \sum_{p \in C_i, q \in C_j} d_{pq}$$

クラスタ数が1つになるまで反復する。

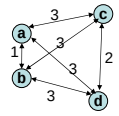


重心間距離を用いたクラスター解析と同じ

UPGMA法による系統樹の計算例

不一致文字数を距離とする

配列a GACT
配列b GTCT
配列c CCAT
配列d CGTT



距離行列

	a	b	c	d
a	0	1	3	3
b		0	3	3
c			0	2
d				0

最小距離のペアを選んで融合

距離行列

	a,b	c	d
a,b	0	**3**	**3**
c		0	2
d			0

(3+3)/2=3　(3+3)/2=3

クラスタと配列の距離は、配列間平均の距離とする

最小距離のペアを選んで融合

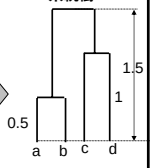
距離行列

	a,b	c,d
a,b	0	**3**
c,d		0

(3+3+3+3)/4=3

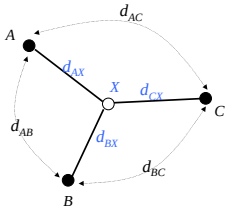
クラスタとクラスタの距離は、クラスタのメンバーの配列間の平均の距離とする

系統樹



距離の半分が枝長

Fitch-Margoliashの式



もとの距離行列d_{ij}を再現することを3つのOTUについて考える。

OTUが3つA,B,Cの場合、その間の3つの距離d_{AB}, d_{BC}, d_{AC}を満たすように、祖先ノードXを作成して、木を作成する。

連立1次方程式

$$\begin{cases} d_{AX}+d_{BX}=d_{AB} \\ d_{BX}+d_{CX}=d_{BC} \\ d_{AX}+d_{CX}=d_{AC} \end{cases}$$

を解くと、

$$d_{AX} = (d_{AB} + d_{AC} - d_{BC})/2$$

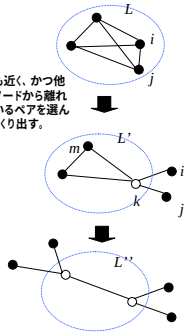
$$d_{BX} = (d_{AB} + d_{BC} - d_{AC})/2$$

$$d_{CX} = (d_{AC} + d_{BC} - d_{AB})/2$$

OTUが3つの場合、この式で、距離行列を完全に満たす枝長を求めることができる。

近隣結合法（Neighbor-Joining法、NJ法）

Saito.N., Nei.N. Mol.Biol.Evol. 4, 406-425,1987.



最も近く、かつ他のノードから離れているペアを選んでくくり出す。

[初期化]
L（相互結合したノード集合）をOTUの集合とする。

[反復]
(1) $d_{ij} - r_i - r_j$ が最小となるi,jをLから選択。

$$r_i = \frac{1}{|L|-2} \sum_{m \in L} d_{im}$$ 他のノードへの平均距離のような値

子ノードi,jを持つ親ノードkを作成し、Lに加える。
また、Lからノードi,jを除く。

(2) 距離行列を更新する。
新ノードkの距離行列は、Fitch-Margoliashの式から、

$$d_{mk} = (d_{im} + d_{jm} - d_{ij}) / 2$$
$$d_{ik} = (d_{ij} + d_{im} - d_{jm}) / 2$$
$$d_{jk} = (d_{ij} + d_{jm} - d_{im}) / 2$$

で定義。ただし、木の枝長となるd_{ik}, d_{jk}については、Lに属する全てのmについての平均の枝長を用いる。

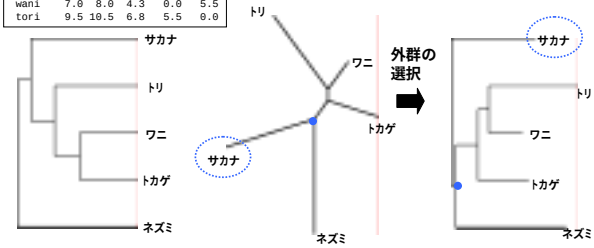
$$d_{ik} = \langle (d_{ij} + d_{im} - d_{jm}) / 2 \rangle_m = (d_{ij} + r_i - r_j) / 2$$
$$d_{jk} = \langle (d_{ij} + d_{jm} - d_{im}) / 2 \rangle_m = (d_{ij} + r_j - r_i) / 2$$

[終了処理]
Lが2つのノードを含むだけになったら終了
残ったノードのどちらかを木のルートノード（3分岐）とする。

UPGMA法とNJ法の樹形の違い

距離行列

sakana	0.0	9.0	7.3	7.0	9.5
nezumi	9.0	0.0	8.3	8.0	10.5
tokage	7.3	8.3	0.0	4.3	6.8
wani	7.0	8.0	4.3	0.0	5.5
tori	9.5	10.5	6.8	5.5	0.0



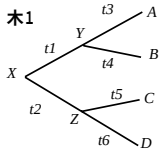
UPGMA法　NJ法（無根）　NJ法（有根）

・無根系統樹から有根系統樹への変換：OTUの中から適切な**外群**(out group)を選べばよい。

外群の選択基準：(1) 他の全てのOTUと相同、(2)他のどのOTUとも十分遠縁

最尤法(maximum likelihood)

分子進化に関する確率モデルを立て、葉ノードの形質を最もよく説明する（最も尤度が高い）系統樹を推定する。



$P_{ab}(t)$：時間tの間にaからbに変異する確率

木1が起こる確率Lは以下で表される。

$$L = P(G) \cdot P_{XY}(t1) \cdot P_{YA}(t3) \cdot P_{YB}(t4) \cdot P_{XZ}(t2) \cdot P_{ZC}(t5) \cdot P_{ZD}(t6)$$

・あるトポロジーについてLを最大化するように枝長($t1,t2,\ldots$)と祖先形質($X,Y,\ldots$)を計算

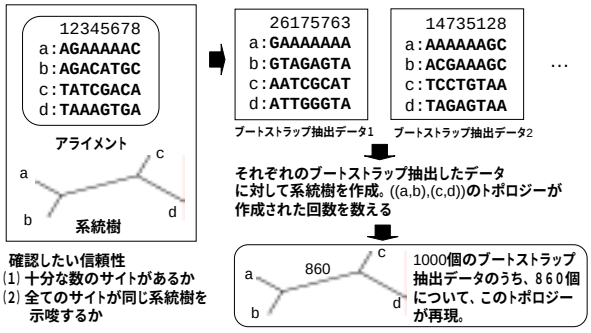
・尤度Lが最も高いトポロジーを探索する

・最節約法と同程度の長い計算時間を必要

系統樹のトポロジーの信頼性の検定

ブートストラップ(bootstrap)抽出を行い多数の擬似データを作成

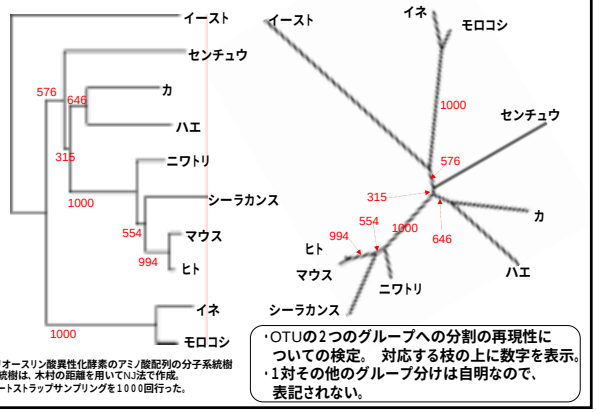
ランダムにサイトを元の数だけ選ぶ。同じサイトを複数回選んでもかまわない。



1000個のブートストラップ抽出データのうち、860個について、このトポロジーが再現。

確認したい信頼性
(1) 十分な数のサイトがあるか
(2) 全てのサイトが同じ系統樹を示唆するか

ブートストラップ値付きの系統樹の例



トリオースリン酸異性化酵素のアミノ酸配列の分子系統樹
系統樹は、木村の距離を用いてNJ法で作成。
ブートストラップサンプリングを1000回行った。

・OTUの2つのグループへの分割の再現性についての検定。 対応する枝の上に数字を表示。
・1対その他のグループ分けは自明なので、表記されない。

分子系統樹作成のためのソフトウエア

- ClustalW/ClustalX
  マルチプルアライメントのソフトだが、NJ法による系統樹作成の機能が付属。ブートストラップ計算にも対応。
- Phylip http://evolution.genetics.washington.edu/phylip.html
  様々な系統樹作成のためのプログラムのセット。最節約法、NJ法、最尤法など多くのアルゴリズムに対応。
  UNIX, DOS,Macに対応。
- MEGA http://www.megasoft.net
  様々な系統樹作成のためのプログラムのセット。最節約法、NJ法、など多くのアルゴリズムに対応。
  Windows/DOS/Macに対応。
- PAUP http://paup.csit.fsu.edu
  最節約法を中心とした系統樹作成ソフト。分子以外の形態データにも対応。有料。

分子系統樹表示のためのソフトウエア

- NJplot http://pbil.univ-lyon1.fr/software/njplot.html
  簡素な有根系統樹の描画ソフト。
- TreeView/TreeViewX
  http://taxonomy.zoology.gla.ac.uk/rod/treeview.html
  http://darwin.zoology.gla.ac.uk/~rpage/treeviewx/index.html
  多機能な系統樹の描画ソフト

参考文献

- 金久實　著　「ポストゲノム情報への招待」（2001） 共立出版
- Arthur M.Lesk(岡崎康司、坊農秀雄　監訳)「バイオインフォマティクス基礎講義 一歩進んだ発想をみがくために」(2003), メディカル・サイエンス・インターナショナル
- 長谷川政美、岸野洋久 「分子系統学」 岩波書店（1996）
- 根井正利、S.クマー「分子進化と分子系統学」 培風館 （2006）
- Durbin R.,Eddy.S.,Krogh A.,Mitchson,G. “Biological Sequence analysis”,Cambridge University Press, 1998.Chapter 7,8.
- R.Durbin 他著、阿久津達也他訳 「バイオインフォマティクス - 確率モデルによる遺伝子解析」医学出版、2001年、9800円