分子系統学演習

データセットの作成から仮説検定まで

田辺晶史

2012 年 4 月 22 日

目次

はじめに

本書は、2008 年 10 月の農林交流センターワークショップ「分子系統樹推定法：理論と応用」での講義用に執筆を開始しました。そのため、当初は受講者の復習と落選者の自習のためのものでした。2009 年 11 月、2010 年 8 月、2011 年 10 月にもワークショップがあり、それに合わせて加筆・修正を加え、現在の姿になりました。

分子系統推定法は、多くの分野で求められている技術となってきましたが、残念なことに未だに確立されたものではなく発展途上です。今回、私の担当する講義では「よく使われている方法」ではなく、(たぶん)「現状では最も良い方法」を提示することにしました。そのため、本書の内容は非常にアグレッシブな内容となっています。本書をお読みの皆さんの中には、既存の論文の中で使われている方法の解説を望まれる方がいらっしゃるかもしれません。おそらくそういう方にも多少は役に立つ情報は載っていると思いますが、本書の目的は「現状では最も良い (と私が勝手に思っている) 方法」を提示することにあることを予めご了承下さい。

本書が皆さんの役に立つことができましたら幸いです。この機会を与えて下さった農業環境技術研究所の三中信宏先生と、本書をお読みの皆さんに感謝します。

第 0 章

必要なソフトウェアのインストールと環境整備

Treefinder、Tracer および FigTree の動作には Java 実行環境が必要ですが、Windows には標準では最新の Java 実行環境が備わっていません。そのため
http://java.com/
から Java 実行環境を入手してインストールしておく必要があります。MacOS X は新しいものであれば、標準で十分に新しい Java 実行環境を備えていますので、Java 実行環境を別途インストールする必要はありません。Linux はディストリビューションによって Java への対応が異なります。Java 実行環境がインストールされていないなら、あらかじめインストールしておいて下さい。

Phylogears2 に含まれるコマンドの実行には Perl 実行環境が必要です。Windows では、あらかじめ ActivePerl などの Windows 用 Perl 実行環境をインストールしておく必要があります。ActivePerl は
http://www.activestate.com/activeperl
からダウンロードできます。インストール先は 32bit 版で標準の `C:\Perl` と仮定して話を進めます。他の場所にインストールした場合は適宜読み替えて下さい。MacOS X や Linux はほとんどの場合 Perl 実行環境は標準で備えています。なお、本書以外の日本語の書籍や、日本語版 Windows 上では `C:¥Perl` と表記・表示されているかもしれませんが、`C:\Perl` と同じ意味です。

また、Windows 環境では、エクスプローラ上で指定したフォルダをカレントとするコマンドプロンプトを簡単に起動できるようになる「ContextConsole Shell Extension」をインストールしておくことをお薦めします。
http://code.kliu.org/cmdopen/
から入手できます。このソフトをインストールすると、フォルダアイコンの右クリックメニューからコマンドプロンプトを起動できるようになります。もっと細かい設定を行うツールとして「いじくるつくーる」というソフトもあります。これを使うと、エクスプローラの右クリックメニューに様々な機能を加えることができます。

http://www.yoshibaworks.com/ayacy/inasoft/rnsf7.html
から入手できます。なお、「カレントフォルダ」というのは、その時点でプログラムを起動すると作業フォルダとして使われるフォルダのことです。プログラムによってはカレントフォルダを無視して任意のフォルダを作業フォルダとするものもあります。フォルダのことをディレクトリと呼ぶこともありますが意味は同じです。

Windows では、標準ではファイル名末尾の拡張子 (.fas とか.nex のこと) が表示されません。これはこの先大変不便なので、表示するように変更しておく必要があります。それにはまず、エクスプローラを起動 (Win キーと E の同時押しで可能です) し、**ツール**メニュー内の**フォルダ オプション**を開きます (Vista ではコントロールパネル内にもあります)。すると、表示されるダイアログに**表示**タブがありますのでそれを選択します。そして、**詳細設定**ペインの中に**登録されている拡張子は表示しない**という項目があり、チェックが入っているはずですので、そのチェックを外して OK を押すと、拡張子が表示されるようになります。また、Windows Vista/7 に搭載されているユーザーアカウント制御という機能は、セキュリティ上重要ではあるのですが、様々なソフトが正常に動作しないようにしてしまう困った機能ですので、もし何か問題があれば無効にして試してみて下さい。ウィルス対策ソフトも誤検出や暴走するものがありますので注意して下さい。実際、筆者は Symantec 社と TrendMicro 社の製品には何度も酷い目に遭わされました。

MacOS X では、一部 Xcode Tools が必要なものがあります。
http://developer.apple.com/xcode/
から事前にダウンロードしてインストールしておいて下さい。旧バージョン OS 用の Xcode は
http://developer.apple.com/
から辿ればダウンロードできますが、登録が必要です。ダウンロードが面倒なら、Mac に付属していた DVD の中に入っているはずですので、それをインストールしていただいてもいいでしょう。その上で、MacPorts という UNIX 関連ツールのパッケージ管理システムをインストールしておくと、様々なソフトが簡単なコマンドでインストールできるようになりますので、興味のある方はインストールしておくと後々役に立つと思います。MacPorts は以下の公式サイトから入手できます。
http://www.macports.org/
以下のページに簡単な使い方が説明されています。
http://d.hatena.ne.jp/hakobe932/20061208/1165646618
その他の日本語の情報は MacPortsWiki-JP
http://www.lapangan.net/darwinports/
に集まっていますのでこちらのページやここから辿れるページを見ればすぐに使えるようになるでしょう。MacPorts と同様のシステムとして Fink、および Homebrew というものもあります。公式サイトは下記になります。
http://www.finkproject.org/
http://mxcl.github.com/homebrew/
これらでも MacPorts と同様のことができますが、複数をインストールすると競合したりするかもしれませんの

でどちらか一方にしておく方が無難です。Fink はコンパイル済みバイナリをインストールしてくれるのでインストールは高速ですが、ソースコードからコンパイルしてくれる上にオプションが選択できる MacPorts の方が筆者の好みです。Homebrew は利用したいパッケージがあるかどうか調べられないのでインストールしていません。ところで、最近は MacOS X をターゲットにしたウィルスが増加し、MacOS X 用のウィルス対策ソフトもそれに伴って増えてきました。以前私の所属する研究室では iAntiVirus という製品をインストールしていましたが頻繁に暴走しました。このような製品にはご注意下さい。

分子系統推定で用いられるソフトウェアには、英語圏で制作されたものが多くあります。そのようなものはしばしば日本語の文字を含んだフォルダ名・ファイル名を正しく扱うことができません。しかし多くの OS でユーザー用のフォルダはユーザー名を含んでいます。そのため、ユーザー名に日本語 (に限らず英数字以外の文字) を用いていると問題が起きる可能性があります。的確なエラーメッセージが表示されれば原因は分かるし対策も打てるのですが、エラーメッセージを見ても原因が分からないことも頻繁にありますので、もしユーザー名に英数字以外を用いていた場合、新たに英数字以外の文字をユーザー名に含まないアカウントを作成してそちらのアカウントでログオンするようにして下さい。特定のファイルやフォルダの最上位のフォルダからの位置を正確に記したものを絶対パスとかフルパスと言います。フルパスにスペースを含んでいると正常に動作しないソフトウェアもあるかもしれませんので注意して下さい。特に、Windows XP ではデスクトップやマイドキュメントはフルパスにスペースが含まれていますので注意が必要です。

また、最近の OS には自動更新機能が搭載されていることがありますが、更新の際に強制的に再起動するものがあります。分子系統推定は非常に時間のかかる解析です。1 ヶ月かかる解析の最中に強制再起動が働いて、また最初からやり直し、などということになっては大変です。例えば、Windows では更新を定期的に確認し、もし見つかれば自動的にインストールして、必要があれば強制再起動するのがデフォルト設定になっています。というわけで、強制的に再起動されたりすることの無いように設定を確認しておいて下さい。処理の重いスクリーンセイバーや常駐ソフトウェアも解析の邪魔になりますので、解析中は無効にしておくことをお薦めします。

系統推定に関連する様々な解析は非常に負荷の高い処理がたくさんあります。バックグラウンドで走らせていると、フォアグラウンドの処理のレスポンスが悪くなってしまったりすることがあります。そういうときは、バックグラウンドの処理の優先度を下げることで改善することがあります。CPU コアを 1 つ常に余らせておくという対処法もありますが、これでは処理速度が大幅に低下します。優先度を下げることは大きな処理速度の低下無しにフォアグラウンド処理のレスポンスを向上できるため、有効な対策だと思います。Windows では、タスクマネージャで該当するプロセス名を右クリックして表示されるメニューから設定できます。MacOS X や Linux では、nice 値という値で優先度が設定できます。nice 値は-20~20 (または 19) の値を取り、-20 が優先度最高、20 (または 19) が最低です。この値は renice コマンドによって変更できます。起動の際に nice コマンドを経由させることでも設定できます。いずれの OS でも、優先度を設定したプロセスから派生する子プロセスにも設定が継承されます。

0.1　テキストエディタのインストール

後の操作の際に「正規表現検索・置換」に対応したテキストエディタがあると大変便利です。正規表現とは、「一定のルールに該当する文字列を検索する」ためのそのルールの記述方法のことです。例えば「2009/10/22」といった日付を全て探したい、別の文字列に置換したい場合に用います。ここでは無料で使える使いやすいテキストエディタをいくつか紹介しておきます。ちなみに Perl でも正規表現検索・置換が可能ですので、これで代用することもできます。

0.1.1　Windows の場合

Windows 用の無料テキストエディタで正規表現検索・置換ができるものとしてはサクラエディタがあります。
https://sourceforge.net/projects/sakura-editor/
からダウンロードできます。執筆時点の最新版 2.0.3 の場合、まず 1.6.6 をインストーラを使ってインストールし、2.0.3 の配布ファイルを展開して得られる sakura.exe を上書きします。さらに、2.0.3 の配布ファイルを展開して得られる QuickStartV2.zip も展開して、得られるファイル群を同じ場所に上書きして下さい。また、1.6.6 に含まれている `bregonig.dll` は古いため 2.0.3 では利用できません。
http://homepage3.nifty.com/k-takata/mysoft/bregonig.html
から Unicode 対応のものをダウンロード、展開して得られる `bregonig.dll` で上書きして下さい。

0.1.2　MacOS X の場合

MacOS X 用の無料テキストエディタでは、CotEditor や mi、TextWrangler が有名でしょう。それぞれ
http://sourceforge.jp/projects/coteditor/
http://mimikaki.net/
http://www.barebones.com/products/TextWrangler/
から入手できます。

0.1.3　Linux の場合

Linux 用であれば Emacs や vim もありますし、GUI を備えたものであれば gEdit (GNOME 用) や Kate (KDE 用) があります。いずれも各ディストリビューションのパッケージ管理システムからインストールできるでしょう。

0.2 EMBOSS のインストール

EMBOSS はオープンソースで開発されているバイオインフォマティクス関連データ解析用ソフトウェアパッケージです。様々なコマンドを含んでおり、データの操作・編集などに便利です。
http://emboss.sourceforge.net/
からダウンロードすることができます。

0.2.1 Windows の場合

Windows 用の EMBOSS は
ftp://emboss.open-bio.org/pub/EMBOSS/windows/
でインストーラが配布されていますのでこれをダウンロードして起動し、指示通りにインストールすれば完了です。

0.2.2 MacOS X・Linux の場合

MacOS X なら MacPorts や Fink から、Linux なら各ディストリビューションのパッケージ管理システムからインストールできることが多いと思いますが、一応ソースコードから入れる方法を書いておきます。まず、公式配布サイトから最新安定版のソースコードをダウンロードします。ファイルを保存した場所が~/temp と仮定すると、以下のようにコマンドを実行することでコンパイル・インストールできます (x.x.x はバージョン番号です)。

```
cd ~/temp
tar xzf ./EMBOSS-x.x.x.tar.gz
cd ./EMBOSS-x.x.x
./configure --prefix=/usr/local --without-x --without-java --without-pngdriver
make
sudo make install
make clean
```

0.3 ClustalW2/X2 のインストール

ClustalW2/X2 は最も一般的に利用されている多重配列整列 (multiple sequence alignment) 用ソフトウェアです。
http://www.clustal.org/

が公式サイトとなっています。ClustalW2 がコマンドライン版、ClustalX2 がグラフィカルインターフェイス版です。ClustalX2 では配列の編集機能もあります。

0.3.1　Windows の場合

どちらもインストーラが用意されていますのでこれをダウンロード・実行して質問に答えるだけでインストールされます。ClustalW2 はインストール後、インストール先にある `clustalw2.exe` を `C:\Perl\bin` にコピーして下さい。

0.3.2　MacOS X の場合

ClustalX2 は公式サイトで配布されているディスクイメージをダウンロードしてマウントすると、中にコマンドがありますのでこれを**アプリケーション** (`/Applications`) などの適当な場所へ移動したり、Dock へ登録するなどして下さい。ClustalW2 もディスクイメージ内にコマンドがありますが、これは適当な場所に置いてから、ターミナルで以下のコマンドを実行してインストールして下さい。ファイルを保存した場所が`~/temp` と仮定します。

```
cd ~/temp
sudo mv ./clustalw2 /usr/local/bin/
```

なお、ClustalX2 は MacPorts や Fink からもインストールできます。

0.3.3　Linux の場合

公式サイトからもバイナリが配布されていますが、各ディストリビューションのパッケージ管理システムからインストールするのが良いでしょう。

0.4　MAFFT のインストール

MAFFT は非常に高速に高精度な多重配列整列 (multiple sequence alignment) を行うことができるソフトウェアです。

http://mafft.cbrc.jp/alignment/software/

にて配布されています。

0.4.1 Windows の場合

All-in-one version というものが用意されていますのでそれをダウンロードして下さい。ファイルを展開して作成されるフォルダの中に、mafft.bat というファイルと ms というフォルダがありますので、それらを C:\Perl\bin に移動またはコピーして下さい。

0.4.2 MacOS X の場合

公式サイトで配布されている MacOS X 用のディスクイメージファイルをダウンロード・マウントして中にあるインストーラを実行して下さい。インストールには管理者権限が必要です。

0.4.3 Linux の場合

公式サイトで配布されているソースコードファイルをダウンロードします。ファイルを展開した場所が ~/temp と仮定すると、以下のようにターミナルでコマンドを実行することでコンパイル・インストールできます。

```
cd ~/temp/mafft-*/core
make
sudo make install
make clean
```

0.5 Unipro UGENE のインストール

Unipro UGENE は多重配列整列 (multiple sequence alignment) および配列データセットの編集用ソフトウェアです。マウスで操作可能なグラフィカルインターフェイスを持ちながら、Windows・MacOS X・Linux のいずれにおいても動作するのが特徴です。

http://ugene.unipro.ru/

からダウンロードすることができます。

0.5.1　Windows の場合

Windows 用配布ファイルを実行すれば簡単にインストールできます。インストール後に起動して、MAFFT コマンドの場所を設定しておくと、Unipro UGENE のメニューから MAFFT による整列を実行できるようになります。Settings メニューから Preferences... を選択し、表示されるダイアログで External Tools を選びます。右側のペイン内に外部コマンドの設定画面が出るので、MAFFT の行にある... を押して C:\Perl\bin\mafft.bat を指定します。同様に clustalw2.exe を指定すれば ClustalW2 による整列も可能になります。

0.5.2　MacOS X の場合

MacOS X 用配布ファイルはディスクイメージになっています。マウントすると、ugeneui という実行ファイルがあります。これを**アプリケーション** (/Applications) などの適当な場所へ移動したり、Dock へ登録するなどして下さい。Windows 版と同様に MAFFT と ClustalW2 の実行ファイルの場所を設定します。MacOS X 版では、Preferences... が ugeneui メニュー内にあるので注意して下さい。

0.5.3　Linux の場合

依存するソフトを全てインストールした上で、公式サイトのバイナリをインストールするか、Ubuntu か Fedora であれば、ディストリビューションに付属のパッケージ管理システムからインストールして下さい。インストール後に Windows 版と同様に MAFFT と ClustalW2 の実行ファイルの場所を設定します。

0.6　Kakusan4 のインストール

Kakusan4 は、筆者が開発した塩基置換モデルの選択を行うためのソフトウェアです。
http://www.fifthdimension.jp/products/kakusan/
からダウンロードすることができます。

0.6.1　Windows の場合

Windows 環境の場合、Windows 用のインストーラが用意されていますので、それをダウンロードしてインストールを実行します。Windows 用インストーラは kakusan4-4.x.yyyy.mm.dd_for_Windows.exe というファ

イル名で配布されています (x はマイナーバージョン番号で yyyy.mm.dd はリリースされた年月日です)。インストーラを実行すると、対話形式のウィンドウが現れますので、指示に従ってインストールを行って下さい。実行する際には、スタートメニューなどのショートカットから起動するか、データファイルを右クリックして現れる**送る**メニューから起動します。入力したいデータファイルが複数ある場合、それらのファイルを全て選択した状態で右クリックの**送る**メニューから起動することで、全てのファイルを Kakusan4 に読み込ませることができます。終了するには Ctrl+C を入力して下さい。

0.6.2 MacOS X の場合

MacOS X 向けには、kakusan4-4.x.yyyy.mm.dd_for_MacOSX.zip という専用のファイルが用意されています (x はマイナーバージョン番号で yyyy.mm.dd はリリースされた年月日です)。これをダウンロードして展開すると、Kakusan4 という実行ファイルが現れます。そのまま実行しても構いませんが、必要に応じてこれを**アプリケーション** (/Applications) などの適当な場所へ移動したり、Dock へ登録するなどして下さい。Kakusan4 を起動する際、ターミナルが起動していない状態ではターミナルが起動されてから Kakusan4 が起動します。終了するには Ctrl+C を入力してからウィンドウを閉じて下さい。

0.6.3 Linux の場合

Linux 用には専用のファイルが用意されていませんので kakusan4-4.x.yyyy.mm.dd.zip をダウンロードして展開して下さい (x はマイナーバージョン番号で yyyy.mm.dd はリリースされた年月日です)。このファイルには尤度計算用の改造版 baseml の実行ファイルが含まれていませんので、配布ファイル中のソースコードからコンパイルする必要があります。配布ファイルを展開してできる Makefile.*を用いてコンパイルを行って下さい。baseml の実行ファイルは Kakusan4 のスクリプトファイル kakusan4.pl と同じ場所に置いておくようにして下さい。配布ファイルには ReadSeq や PHYLIP などの必要なコマンドも含まれていませんので、これらのインストールも必要です。ReadSeq は
ftp://ftp.bio.indiana.edu/molbio/readseq/java/readseq.jar
からダウンロードできる Java 版の実行可能ファイル readseq.jar を Kakusan4 のスクリプトファイル kakusan4.pl と同じ場所に置いておけばいいでしょう (ただし、Java 実行環境が必要です)。PHYLIP は多くのディストリビューションのパッケージ管理システムからインストール可能なはずです。公式サイトからソースコードをダウンロードしてくればコンパイルしてインストールすることもできます。また、Statistics::Distributions と Statistics::ChisqIndep という 2 つの Perl モジュールも必要です。配布ファイルに同梱されていますので Perl モジュールのフォルダに置くか、CPAN からインストールして下さい。起動するには、ターミナル上で以下のコマンドを実行して下さい。終了するには Ctrl+C を入力して下さい。

```
kakusan4.pl --interactive=enable
```

0.7　Aminosan のインストール

Aminosan は、筆者が開発したアミノ酸置換モデルの選択を行うためのソフトウェアです。
http://www.fifthdimension.jp/products/aminosan/
からダウンロードすることができます。

0.7.1　Windows の場合

Windows 環境の場合、Windows 用のインストーラが用意されていますので、それをダウンロードしてインストールを実行します。Windows 用インストーラは aminosan-x.x.yyyy.mm.dd_for_Windows.exe というファイル名で配布されています (x.x はバージョン番号で yyyy.mm.dd はリリースされた年月日です)。インストーラを実行すると、対話形式のウィンドウが現れますので、指示に従ってインストールを行って下さい。実行する際には、スタートメニューなどのショートカットから起動するか、データファイルを右クリックして現れる**送る**メニューから起動します。入力したいデータファイルが複数ある場合、それらのファイルを全て選択した状態で右クリックの**送る**メニューから起動することで、全てのファイルを Aminosan に読み込ませることができます。終了するには Ctrl+C を入力して下さい。

0.7.2　MacOS X の場合

MacOS X 向けには、aminosan-x.x.yyyy.mm.dd_for_MacOSX.zip という専用のファイルが用意されています (x.x はバージョン番号で yyyy.mm.dd はリリースされた年月日です)。これをダウンロードして展開すると、Aminosan という実行ファイルが現れます。そのまま実行しても構いませんが、必要に応じてこれを**アプリケーション** (/Applications) などの適当な場所へ移動したり、Dock へ登録するなどして下さい。Aminosan を起動する際、ターミナルが起動していない状態ではターミナルが起動されてから Aminosan が起動します。終了するには Ctrl+C を入力してからウィンドウを閉じて下さい。

0.7.3　Linux の場合

Linux 用には専用のファイルが用意されていませんので aminosan-x.x.yyyy.mm.dd.zip をダウンロードして展開して下さい (x.x はバージョン番号で yyyy.mm.dd はリリースされた年月日です)。配布ファイルに

は ReadSeq や PHYLIP などの必要なコマンドは含まれていませんので、これらのインストールも必要です。ReadSeq は
ftp://ftp.bio.indiana.edu/molbio/readseq/java/readseq.jar
からダウンロードできる Java 版の実行可能ファイル `readseq.jar` を Aminosan のスクリプトファイル `aminosan.pl` と同じ場所に置いておけばいいでしょう (ただし、Java 実行環境が必要です)。PHYLIP は多くのディストリビューションのパッケージ管理システムからインストール可能なはずです。公式サイトからソースコードをダウンロードしてくればコンパイルしてインストールすることもできます。また、Statistics::Distributions と Statistics::ChisqIndep という 2 つの Perl モジュールも必要です。配布ファイルに同梱されていますので Perl モジュールのフォルダに置くか、CPAN からインストールして下さい。起動するには、ターミナル上で以下のコマンドを実行して下さい。終了するには `Ctrl+C` を入力して下さい。

```
aminosan.pl --interactive=enable
```

0.8 Treefinder のインストール

Treefinder は最尤系統推定に関連した様々な処理を実行できるソフトウェアです。
http://www.treefinder.de/
から Windows・Linux 用の配布ファイルを得ることができます。

0.8.1 Windows の場合

Windows 環境用の Treefinder は `tf-*-windows.exe` というファイル名で配布されています (*はリリースされた月・年が入ります)。これをダウンロードして実行すればインストールできます。現在は公式サイトがストライキ中だそうですが、URL を直接指定すればダウンロード可能です。最新版のファイル名は `tf-march2011-windows.exe` ですので、公式サイト URL の後ろにファイル名を足して直接指定して下さい。

さらに、`Treefinder` フォルダ内の `tf.exe` を `C:\Perl\bin` へコピーしておいて下さい。

また、Microsoft の「Visual C++ 2010 SP1 再頒布可能パッケージ」がインストールされている必要があります。インストールされていない場合は Microsoft から入手してインストールしておいて下さい。

0.8.2 MacOS X の場合

MacOS X 用のファイルは `tf-*-mac.zip` というファイル名で配布されています (*はリリースされた月・年が入ります)。現在、最新版は配布されておらず旧版もリンクがありませんが、ファイル名

を直接指定すればダウンロードできます。最終バージョンのファイル名は tf-october2008-mac.zip です。公式サイト URL の後ろにファイル名を繋げて指定して下さい。ダウンロードしたファイルを展開し、得られる Treefinder フォルダを**アプリケーション** (/Applications) フォルダに移動して下さい。比較的新しい PowerPC G5 搭載マシンや、Intel 製 CPU 搭載マシンでは、tf-october2008-mac-g5-binary.zip や tf-october2008-mac-intel-binary.zip をダウンロード・展開し、得られた tf.bin を/Applications/Treefinder/tf.bin に上書きして下さい。特に新しい Intel CPU 搭載 Mac では必ず実行して下さい。その上で、ターミナルを起動して以下のようにコマンドを実行します。

```
cd /Applications
chmod -R 755 ./Treefinder
sudo mkdir -p /usr/local/bin
sudo ln -f /Applications/Treefinder/tf /usr/local/bin/tf
sudo ln -f /Applications/Treefinder/treefinder /usr/local/bin/treefinder
```

起動するには、ターミナルで以下のコマンドを実行します。

```
treefinder
```

0.8.3 Linux の場合

Linux 用の配布ファイルは tf-*-linux.zip です (*はリリースされた月・年が入ります)。これをダウンロードして展開し、得られる Treefinder ディレクトリのあるディレクトリをカレントにして以下のようにコマンドを実行することでインストールを行います。

```
chmod -R 755 ./Treefinder
sudo mkdir -p /opt
sudo mv ./Treefinder /opt/
sudo ln -s /opt/Treefinder/tf /usr/local/bin/tf
sudo ln -s /opt/Treefinder/treefinder /usr/local/bin/treefinder
```

なお、最新版のファイル名は tf-march2011-linux.zip です。公式サイト URL に続けてファイル名を指定してアクセスすればダウンロード可能です。

起動するには、X 上のターミナルで以下のコマンドを実行します。

```
treefinder
```

0.9 PAUP*のインストール

PAUP*は非常によく用いられている最節約・最尤系統推定ソフトウェアです。
http://paup.csit.fsu.edu/
の情報に従って購入する必要があります。お持ちでない場合はこの節は読み飛ばして下さい。

0.9.1 Windows の場合

PAUP*の配布ファイル展開・更新を行うと、win-paup4b10-console.exe または paup.exe というファイルが得られます。win-paup4b10-console.exe だった場合は paup.exe という名前に変更してから、paup.exe だった場合はそのまま、C:\Perl\bin へ移動またはコピーして下さい。使用する際は、コマンドプロンプトから以下のコマンドを実行して起動します。Quit というコマンドを入力して実行することで終了できます。

```
paup
```

0.9.2 MacOS X・Linux の場合

UNIX 用の配布ファイルの中から適切なものを選択してダウンロードします。そして、配布ファイルを展開して得られた paup4b10-*を paup という名前に変更してインストールします。展開した場所が~/temp と仮定すると、以下のようにターミナルでコマンドを実行して下さい。

```
cd ~/temp/paup4b10-*
chmod 755 ./paup4b10-*
sudo mkdir -p /usr/local/bin
sudo mv ./paup4b10-* /usr/local/bin/paup
```

以降、ターミナルで以下のようにコマンドを実行することで PAUP*が起動します。Quit というコマンドを入力して実行することで終了できます。

```
paup
```

0.10 TNT のインストール

TNT は超高速な最節約系統推定ソフトウェアです。
http://www.zmuc.dk/public/phylogeny/TNT/

からダウンロードすることができます。

TNT では配列名を 31 文字までしか認識することができません。本来もっと多くの文字数を配列名に用いることができる形式の配列ファイルでも、31 文字までしか認識しないので注意が必要です。

0.10.1　Windows の場合

配布されているファイルがいくつかありますが、コマンドラインインターフェイス版の Win (charmode) を選んで下さい。ダウンロードされた zipchtnt.zip を展開すると、tnt.exe が得られますので、これを C:\Perl\bin へ移動またはコピーして下さい。使用する際は、コマンドプロンプトから以下のコマンドを実行して起動します。

```
tnt
```

なお、初回起動時にはライセンス認証を求められますのでライセンスをよく読んで同意しておいて下さい。

0.10.2　MacOS X・Linux の場合

自分の環境に合わせて適当なバイナリをダウンロードして展開して下さい。ファイルを展開すると、tnt.command (MacOS X の場合) または tnt (Linux の場合) が得られます。MacOS X 環境で、展開した場所が~/temp と仮定すると、以下のようにターミナルでコマンドを実行してインストールします。Linux の方はファイル名が異なりますが適宜読み替えて下さい。

```
cd ~/temp
chmod 755 ./tnt.command
./tnt.command
sudo mv ./tnt.command /usr/local/bin/tnt
```

コマンド実行中に使用許諾契約が表示されたら、y をタイプするごとに続きが表示され、最後の質問に I agree とタイプして答えます (もちろん内容に同意する場合だけです)。上記のコマンドを全て実行することでインストールは完了です。以降、ターミナルで以下のようにコマンドを実行することで TNT が起動します。

```
tnt
```

なお、使用許諾契約にはユーザーごとに同意しておく必要があります。インストールしたユーザーとは別のユーザーでも系統解析の必要があるのなら、そのユーザーでログインした状態で再度 TNT を起動して使用許諾契約への同意を済ませておきます。

0.11 Phylogears2 のインストール

Phylogears2 は筆者が開発した系統解析用の Perl スクリプト集です。配列データの編集・変換や Treefinder などの並列実行を行うことができます。
http://www.fifthdimension.jp/products/phylogears/
からダウンロードできます。

0.11.1 Windows の場合

公式サイトから Windows 用の ZIP ファイルをダウンロードして下さい。Windwows 用の配布ファイル名は `phylogears-x.x.yyyy.mm.dd_for_Windows.zip` となっています (`x.x` はバージョン番号、`yyyy.mm.dd` はリリース年月日です)。ZIP ファイルを展開すると、`phylogears-x.x.yyyy.mm.dd_for_Windows` というフォルダ内の `bin` フォルダにたくさんのコマンドがありますが、これらを全て `C:\Perl\bin` へ移動またはコピーして下さい。

また、疑似乱数生成用の Perl モジュールとして、Math::Random::MT::Perl か Math::Random::MT::Auto をインストールしておく必要があります (両方ある場合は Math::Random::MT::Auto を優先します)。Math::Random::MT::Auto をインストールするには、コマンドプロンプト上で以下のようにコマンドを実行して下さい。インターネットから自動的に必要なファイルをダウンロードしてきてインストールが完了します。Math::Random::MT::Perl を利用する場合は適宜変更して下さい。

```
ppm install Math-Random-MT-Auto
```

インストールする Perl のバージョンによってどちらか一方しか提供されていないことがありますので、エラーが出たら別の方をインストールしてみて下さい。インストール後、以下のコマンドを実行して何もエラーが表示されなければインストールは正常に完了しています。

```
perl -e "use Math::Random::MT::Auto"
```

χ^2 独立性の検定を行う `pgtestcomposition` コマンドを使いたい場合は、さらに Statistics::ChisqIndep というモジュールも必要になってきます。Windows では先程と同様に、以下のコマンドでインストールが完了します。依存するモジュールも含めてインストールされます。

```
ppm install Statistics-ChisqIndep
```

インストールを確認するには、以下のコマンドを実行してみます。

```
perl -e "use Statistics::ChisqIndep"
```

エラーが出なければ問題ありません。

0.11.2　MacOS X・Linux の場合

汎用の配布ファイル phylogears-x.x.yyyy.mm.dd.zip をダウンロード・展開して下さい (x.x はバージョン番号、yyyy.mm.dd はリリース年月日です)。展開した場所が~/temp と仮定すると、以下のようにコマンドを実行してインストールを完了します。

```
cd ~/temp/phylogears-x.x.yyyy.mm.dd/bin
chmod 755 ./*
sudo mkdir -p /usr/local/bin
sudo mv ./* /usr/local/bin
cd ../share
sudo mv ./phylogears /usr/local/share/
```

MacOS X・Linux でも、疑似乱数生成用 Perl モジュール Math::Random::MT::Auto を別途インストールしておく必要があります。インターネットにアクセスできる状態にして、以下のコマンドを実行すると、依存モジュールも含めてインストールされます。

```
sudo -H cpan -i Math::Random::MT::Auto
```

cpan コマンドの初回実行時にはいくつかの質問に答える必要がありますので、適切に答えを入力して下さい。インストール後、以下のコマンドを実行してエラーが表示されなければインストールは正常に完了しています。

```
perl -e "use Math::Random::MT::Auto"
```

χ^2 独立性の検定を行う pgtestcomposition コマンドを使いたい場合は、さらに Statistics::ChisqIndep というモジュールも必要になります。このモジュールをインストールするには下記のコマンドを実行します。

```
sudo -H cpan -i Statistics::ChisqIndep
```

0.12　CONSEL のインストール

CONSEL は座位ごとの尤度のブートストラップリサンプリング (resampling of estimated log-likelihoods, RELL) を用いて仮説間の検定を行うソフトウェアです。Treefinder にもこの機能は備わっていますが、Treefinder

は一部制限があるため、より柔軟に検定を行うにはこのソフトウェアが必要です。以下の URL が公式サイトです。

http://www.is.titech.ac.jp/ shimo/prog/consel/

0.12.1 Windows の場合

公式サイトでは旧版の実行バイナリしか配布されていませんが、筆者がコンパイルした実行バイナリを

http://www.fifthdimension.jp/documents/molphytextbook/consel-0.2.zip

に置いてありますので、これをダウンロードして下さい。展開すると、consel-0.2 フォルダ内に i686-bin フォルダがあります。その中のファイル群を C:\Perl\bin へ移動またはコピーして下さい。64bit 版の Windows では x86_64-bin フォルダの中のファイルを使って下さい。

0.12.2 MacOS X・Linux の場合

どちらも公式サイトのソースコードをダウンロード・展開してコンパイル・インストールすることで利用できるようになります。配布ファイルの保存先を~/temp と仮定すると、以下のようにターミナルでコマンドを実行して下さい。

```
cd ~/temp
tar xzf ./cnsls*.tgz
cd ./consel/src
make
make install clean
cd ../bin
sudo mv * /usr/local/bin/
```

0.13 trimAl のインストール

trimAl は整列が信用できない領域を推定し、除去するためのソフトです。以下の Web サイトからダウンロードできます。

http://trimal.cgenomics.org/

0.13.1 Windows の場合

公式サイトから Windows 用の配布ファイルをダウンロード・展開すると、bin フォルダの中に実行ファイルがあります。これを C:\Perl\bin へ移動またはコピーして下さい。

0.13.2　MacOS X・Linux の場合

どちらも公式サイトのソースコードをダウンロード・展開してコンパイル・インストールすることで利用できるようになります。配布ファイルの保存先を~/temp と仮定すると、以下のようにターミナルでコマンドを実行して下さい。

```
cd ~/temp
tar xzf ./trimal.*.tar.gz
cd ./trimAl/source
make
sudo mv trimal /usr/local/bin/
sudo mv readal /usr/local/bin/
make clean
```

0.14　MrBayes5D のインストール

MrBayes5D は最もよく利用されているベイジアン系統推定ソフトウェアである MrBayes の改造版です。
http://www.fifthdimension.jp/products/mrbayes5d/
からダウンロードできます。オリジナルは
http://mrbayes.csit.fsu.edu/
から入手することができます。

0.14.1　Windows の場合

配布ファイルをダウンロード・展開して得られる mrbayes5d.exe を C:\Perl\bin へ移動またはコピーして下さい。利用する際には、コマンドプロンプトで以下のようにコマンドを入力することで起動できます。

```
mrbayes5d
```

0.14.2　MacOS X の場合

配布ファイルをダウンロード・展開して得られる mrbayes5d_osx.command をどこか適当な場所に置いて下さい。起動の際はこのファイルを直接実行して下さい。ターミナルのウィンドウが開いて本プログラムが起動します。MacOS X が古いと、このファイルを実行することができません。もしも起動できなかった場合は以下に述べる Linux の場合と同様にソースコードからコンパイルする必要があります。

0.14.3 Linux の場合

配布ファイルを保存した場所が~/temp と仮定すると、以下のようにコマンドを実行してインストールします(x.x.x はオリジナル版のバージョン番号、yyyy.mm.dd は改造版のリリースされた年月日です)。

```
cd ~/temp
unzip ./mrbayes5d-x.x.x.yyyy.mm.dd.zip
cd ./mrbayes5d-x.x.x.yyyy.mm.dd
make
sudo mkdir -p /usr/local/bin
sudo mv ./mrbayes5d /usr/local/bin/mrbayes5d
make clean
```

MPI を用いた並列版をコンパイルするには、LAM/MPI や OpenMPI といった MPI をサポートしたフレームワークをあらかじめインストールしておきます。既に準備が済んでいるならば、以下のようにコマンドを実行することで MPI 版をコンパイルすることができます。

```
cd ~/temp
unzip ./mrbayes5d-x.x.x.yyyy.mm.dd.zip
cd ./mrbayes5d-x.x.x.yyyy.mm.dd
MPI=yes make
mv ./mrbayes5d ~/mrbayes5d-mpi
make clean
```

これで、~/に mrbayes5d-mpi ができます。起動するには以下のようにコマンドを実行します。

```
mpirun -np 利用するCPU数 ~/mrbayes5d-mpi
```

なお、MPI フレームワークとして LAM/MPI をインストールした場合は mpirun で起動する前に lamboot -v を実行しておく必要があります。解析後は lamhalt を実行しておきます。OpenMPI であればこれらは必要ありません。

0.15 Tracer のインストール

Tracer は MrBayes・MrBayes5D の出力ファイルを解析して収束具合を判断するために用いるソフトです。
http://tree.bio.ed.ac.uk/software/tracer/
から入手することができます。

0.15.1　Windows の場合

Windows 用配布ファイルをダウンロード・展開して得られる、Tracer v*.exe を実行することで起動します (*はバージョン番号)。この実行ファイルを含むフォルダごと適当な場所 (C:\Program Files など) に置き、ショートカットをデスクトップやスタートメニューに登録しておくとよいでしょう。

0.15.2　MacOS X の場合

MacOS X 用の.dmg イメージファイルをダウンロードしてマウントすると、実行ファイルが入っていますので**アプリケーション** (/Applications) などの適当な場所へ移動したり、Dock へ登録するなどして下さい。

0.15.3　Linux の場合

JAR 形式ファイルを含む圧縮ファイルをダウンロードして下さい。ファイルを保存した場所が~/temp と仮定すると、以下のようにコマンドを実行することで展開・インストールできます (*はバージョン番号です)。

```
cd ~/temp
tar xzf Tracer_v*.tgz
cd ./Tracer_v*/bin
chmod 755 ./tracer
sudo mkdir -p /usr/local/bin
sudo mv ./tracer /usr/local/bin
cd ../lib
sudo mkdir -p /usr/local/lib
sudo mv ./tracer.jar /usr/local/lib
```

起動する際はターミナルで以下のコマンドを実行して下さい。

```
tracer
```

0.16　FigTree のインストール

FigTree は系統樹の描画・編集を行うためのソフトウェアです。

http://tree.bio.ed.ac.uk/software/figtree/

で配布されています。インストール方法は Tracer とほぼ同様です。

第 1 章

配列データセットの作成

1.1 配列データファイルの形式と相互変換

各種データベースやソフトウェアでは、様々なデータファイル形式が用いられており、利用時には相互に変換する必要がしばしば生じます。以下ではまず各種ファイル形式について簡単に解説した後、相互変換方法について述べます。

1.1.1 各ファイル形式の特徴

GenBank 形式

Web 上の配列データベースにおけるスタンダードなファイル形式です。配列データ以外に、その配列に関する様々な注釈 (annotation) 情報を加えることができます。それらの情報に基づいたデータの加工処理もソフトウェアを用いて簡単に行うことができるため大変便利です。人間にとってもプログラムにとっても可読性の高いファイル形式と言えるでしょう。最も単純な場合は以下のような形式です。

ファイルの内容 1.1　GenBank 形式配列

```
LOCUS       ABC1234       60 bp
DEFINITION  TaxonA 18S small subunit ribosomal RNA gene, partial sequence.
ORIGIN
        1 AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
//

LOCUS       ABC1235       60 bp
DEFINITION  TaxonB 18S small subunit ribosomal RNA gene, partial sequence.
ORIGIN
        1 AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
//

LOCUS       ABC1236       60 bp
DEFINITION  TaxonC 18S small subunit ribosomal RNA gene, partial sequence.
ORIGIN
        1 AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
//
```

ほとんどの場合はもっと様々な情報を含んでいるので、これほどシンプルではありません。

FASTA 形式

Web 上の配列データベースは、この形式でのデータ出力にも対応していることが多いと思います。しかし、注釈 (annotation) 情報はありませんので、それらの情報を用いた加工を行いたい場合には不適です。また、塩基配列決定を行った場合には、波形の編集や複数の配列を結合 (assemble) した後、このファイル形式に配列データを書き出すことが多いでしょう。ほとんどの多重配列エディタ (multiple sequence editor) においてもスタンダードなファイル形式であり、いずれのソフトにおいても入力の互換性は高いと言えます。実際の配列編集を行う際にはこのファイル形式で作業することが多いでしょう。ClustalW/X では配列データキャラクタとして?に対応していないため、もし?があるなら N などに置換しておく必要があります。以下に典型的な FASTA 形式ファイルを示します。

ファイルの内容 1.2　FASTA 形式配列

```
1 >TaxonA
2 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
3 >TaxonB
4 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5 >TaxonC
6 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Clustal 形式

ClustalW/X において多重配列アライメント (multiple sequence alignment) を行った際に出力されるデフォルトファイル形式です。オプション設定により他の形式での出力も可能です。

ファイルの内容 1.3　Clustal 形式配列

```
1 CLUSTAL 2.0.12 multiple sequence alignment
2
3
4 TaxonA           AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
5 TaxonB           AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
6 TaxonC           AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
7                  ************************************************************
```

PHYLIP 形式

系統解析ソフトウェアにおいて最も多く利用されているファイル形式の一つです。単純なファイル形式ですが方言が多くあり、解析ソフトウェアごとにマニュアルを良く読んで確認する必要があるのがやっかいです。配列名の文字数に制限があり、元々は 10 文字しか使えませんでした。しかし、これを拡張して配列名と配列の間をスペースで区切ることにして配列名の文字数制限を緩めたものも多く使われています。最大の問題は、元々の配列名文字数 10 文字の仕様では配列名と配列との間をスペースで区切る必要が無かったため、配列名が 10 文字

ぴったりの場合に両者に互換性が無いことです。よって、この形式を用いる際には配列名を 10 文字以内にした上で必ず配列名と配列の間をスペースで区切るようにし、元々の PHYLIP 形式の仕様に準拠したものとするのが安全です。閲覧・編集に適した interleaved 形式もあり、テキストエディタでの操作に適しています。PHYLIP では配列内にはスペースが含まれていても問題ありませんが、ソフトウェアによっては配列は一続きの文字列であることを仮定しているものもあります。interleaved 形式に対応していないソフトウェアもあります。また、1 行空けてさらに同じ形式でデータを続けることで、ブートストラップリサンプリングしたりした多数のデータセットを 1 ファイルに格納することもできます。GenBank・Clustal・FASTA ではそのようなことはできません。

non-interleaved と interleaved の違いは実際のファイルの中身を見ていただくのが分かり易いでしょう。以下が non-interleaved の PHYLIP 形式配列ファイルです。

ファイルの内容 1.4　non-interleaved PHYLIP 形式配列

```
3 60
TaxonA     AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
           AAAAAAAAAA
TaxonB     AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
           AAAAAAAAAA
TaxonC     AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
           AAAAAAAAAA
```

そして、これが interleaved の PHYLIP 形式ファイルです。

ファイルの内容 1.5　interleaved PHYLIP 形式配列

```
3 60
TaxonA     AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
TaxonB     AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA
TaxonC     AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA

           AAAAAAAAAA
           AAAAAAAAAA
           AAAAAAAAAA
```

どちらも 50 座位で折り返しているのですが、non-interleaved 形式ではそれぞれの配列ごとに折り返しているのに対して、interleaved 形式では全配列をセットで折り返しています。前述のように interleaved 形式に対応していないソフトもありますが、non-interleaved なのに折り返しがあるファイルに対応していないソフトもありますので注意が必要です。

NEXUS 形式

系統解析ソフトウェアにおいて最も多く利用されているもう一つのファイル形式です。様々な「ブロック」を記述することができ、対応しているソフトウェア用のコマンドを記述しておくことができます。その「ブロック」に非対応のソフトウェアではその中の内容は無視されますので通常問題は生じません。配列も `Data` ブロックというブロック内に記述します。本形式にも閲覧・編集に適した interleaved 形式があり、テキストエディタでの操作に向いています。また、PHYLIP 形式と同様、さらに `Data` ブロックを作成することで、ブートストラッ

プリサンプリングしたりした多数のデータセットを 1 ファイルに格納することもできます。GenBank・Clustal・FASTA ではそのようなことはできません。

ファイルの内容 1.6　NEXUS 形式配列

```
#NEXUS

Begin Data;
  Dimensions NTax=3 NChar=60;
  Format DataType=DNA Interleave Missing=? Gap=-;
Matrix
TaxonA   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TaxonB   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
TaxonC   AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

TaxonA   AAAAAAAAAA
TaxonB   AAAAAAAAAA
TaxonC   AAAAAAAAAA
  ;
End;
```

1.1.2　データ形式の相互変換

配列名には、基本的に英数字とアンダースコア以外は使わないようにした方が無難です。その他の特殊記号を用いていてうまくいかない場合には一時的に特殊記号を他の文字列に置き換えておくとよいでしょう。しかし、そのような文字は解析ソフト側でも問題が発生しやすいのでできるだけ使用は避けましょう。

seqret によるデータ変換

seqret は EMBOSS に含まれている配列ファイル入出力コマンドです。ほとんどの形式に対応しており、配列形式の相互変換に便利です。対応形式の一覧は
http://emboss.sourceforge.net/docs/themes/SequenceFormats.html
にあります。入力ファイルを PHYLIP/NEXUS 形式へ変換するには以下のようにコマンドを実行します。

```
seqret input_file phylip::output_file
seqret input_file nexus::output_file
```

入力ファイル形式がうまく認識されていないと思われる状況では、入力ファイル形式を以下のように指定することで改善することがあります。

```
seqret fasta::input_file phylip::output_file
```

ClustalW2/X2 によるデータ変換

ClustalW2/X2 は、FASTA・EMBL・Clustal・GCG などの形式を読み込み、これらに加えて PHYLIP・NEXUS 形式で書き出すことができます。ClustalX2 の場合は、まず起動して File メニューから Load sequences を選択して入力ファイルを指定します。ファイルの読み込みが完了したところで File メニューの Save sequence as... を選択することでダイアログが表示されます。出力したいファイル形式にチェックを入れて OK を押すことで変換後のファイルが保存されます。ファイル名は、Save Sequence As: (File extension will be appended) の欄に書かれた内容に、チェックを入れた形式の拡張子が付加されたものとなります。具体的には PHYLIP 形式は.phy、NEXUS 形式は.nxs が付加されます。

ClustalW2 では、以下のようにコマンドを実行することで PHYLIP/NEXUS 形式への変換が可能です。

```
clustalw2 -convert -output=phylip -outfile=output_file -infile=input_file
clustalw2 -convert -output=nexus -outfile=output_file -infile=input_file
```

出力ファイル名が指定されていない場合には入力ファイルの拡張子を変更したファイルが作成されます。Windows 版では、-convert ではなく/convert などと - ではなく / を使ってオプションを指定する必要がありますのでご注意下さい。コマンドラインオプションを一覧表示するには以下のように実行します。

```
clustalw2 -options
```

Phylogears2 によるデータ変換

Phylogears2 には、FASTA・NEXUS・PHYLIP・Treefinder の 4 形式の相互変換が可能な pgconvseq コマンドがあります。NEXUS と PHYLIP は多数のデータセットを 1 ファイルに格納できますが他の形式はそうではないので、多数のデータセットを格納している NEXUS や PHYLIP 形式を FASTA や Treefinder 形式に変換する場合は、独自ルールで書き出します。FASTA の場合、データセット間に空行を設けます。Treefinder では、% end of data というコメントを間に挟みます。これらを正しく解釈できるソフト (Phylogears2 の一部コマンドだけです) でしかこれらが多数のデータセットであることを認識できません。また、変換元の FASTA 形式配列に空行があると、NEXUS や PHYLIP 形式に出力した際に別データセットとされてしまうので注意が必要です。使い方は下記のようになります。

```
pgconvseq --output=PHYLIP input_file output_file
pgconvseq --output=NEXUS input_file output_file
pgconvseq --output=TF input_file output_file
```

なお、PHYLIP 形式では本来配列名は 10 文字以下でなくてはなりませんが、配列形式として PHYLIPex を指

定することで 11 文字以上の配列名も許容したファイルを作成することができます。PHYML・RAxML・PAML では、この形式で長い OTU 名を使うことができます。

Treefinder によるデータ変換

Treefinder は、独自形式に加えて PHYLIP・NEXUS・FASTA 形式の読み書きができます。

グラフィカルインターフェイスの場合、Utilities メニュー下にある Transform Sequence Data ... を選択して下さい。Sequence File に入力ファイルを指定し、Save As に出力ファイル名を指定します。その上で適宜出力ファイル形式を選択して OK を押せばファイルが出力されます。

コマンドラインから操作する場合には tf コマンドを用います。何も指定せずに tf を実行すると対話型インターフェイスが起動します。このインターフェイス上でファイルを変換して出力するには以下のように入力します。

```
TL> SaveSequences[LoadSequences["input_file"],"output_file",Format->"FASTA"]
```

以下のように入力しても同じ結果になります。

```
TL> "input_file",LoadSequences,"output_file",Format->"FASTA",SaveSequences
```

対話型インターフェイスを終了するには、Quit を実行します。

上述の入力コマンドをテキストファイルに保存した上で、以下のように実行してやれば、テキストファイル内のコマンドが実行されるため、同様に入力ファイルの内容が変換されて出力ファイルに保存されます。

```
tf command_file
```

1.2　目的の配列を入手する

以下では配列データベースから目的の配列を探し出して得る方法について述べます。

1.2.1　分類群・遺伝子の名前から探す

配列が欲しい分類群が分かっているなら、分類群名データベースから辿ることで目的の配列を得ることができます。

まず、NCBI Taxonomy のサイトを開きます。URL は下記です。

http://www.ncbi.nlm.nih.gov/taxonomy/

このページで表示される検索ボックスから正式な分類群名で検索すると、データベース内で見つかった分類群のリストが出ますので、目的の分類群のリンクをクリックします。すると、高次分類群であれば所属する下位の分類群の階層化リストが出ます。最上位の分類群名をクリックすると、NCBI の他のデータベース内にある当該分類群のデータエントリ件数のリストが表示されています。高次分類群でなく種であればすぐにこの表示になります。件数にリンクが設定されていますのでクリックしてリンク先に跳ぶと、選択したデータベース内での当該分類群のデータエントリがずらっと出てきます。この状態で検索ボックスに遺伝子名などを追加すれば絞り込むことができます。NCBI Taxonomy を使わずとも、Nucleotide や Protein のデータベースで分類群名で検索しても構いませんが、漏れや余計なものが入りやすいのでこちらの方法がおすすめです。

次に探したいデータの遺伝子名が分かっている場合です。この場合も分類群同様に遺伝子名データベースから辿ればよいでしょう。

まず、NCBI Gene のサイトを開きます。URL は下記になります。

http://www.ncbi.nlm.nih.gov/gene/

こちらの検索ボックスで目的の遺伝子名で検索します。ただ、それだけでは大量にヒットしてしまいますので、分類群名などを追加して絞り込むとよいでしょう。分類群名と同様、Nucleotide や Protein のデータベースで遺伝子名で検索してもよいでしょう。他のデータベースへのリンクはあるものの分類群と違ってあまり役に立たないのでその方が手っ取り早いかもしれません。

NCBI の Nucleotide や Protein のデータベースでは、それぞれのデータエントリにはそのデータ元の生物名、遺伝子名、配列長などの様々な情報が項目ごとに記載されています。ですから、それぞれの項目を指定してキーワード検索できれば余計なものが引っかかりにくくなったりして便利です。そのためには、以下のようなキーワードを書けばよいことになっています。

```
キーワード[項目指定語]
```

項目指定語の一覧は以下の URL で説明されています。

http://www.ncbi.nlm.nih.gov/books/NBK49540/

例えば、以下のようなキーワードを付加することで配列長が 100~1,000 のエントリのみに絞り込むことができます。

```
100:1000[Sequence Length]
```

これらの項目指定検索を組み合わせてやることで目的のエントリを見つけやすくなるでしょう。

目的のエントリが見つかったら、エントリ名をクリックすればいいですし、複数件ある場合は各エントリの頭にあるチェックボックスにチェックを入れてから検索結果リストの上にある Display プルダウンメニューから GenBank を選択すれば、チェックを入れたエントリの生データ、即ち GenBank 形式配列が表示されます。Show プルダウンメニューからは 1 ページに表示する件数、並び替えに使うもの (Sorted By) などを指定できます。Send to からは Text を選べばプレーンテキストで表示され、File ならローカルファイルへ保存するダイアログが出るはずです。つまり、GenBank 形式で表示している状態で Send to を File にすれば、表示している GenBank 形式データをごっそり手元のマシンに保存できます。

1.2.2　配列から類似配列を探す

NCBI BLAST から配列データベース中の類似配列を探索することができます。URL は下記です。
http://www.ncbi.nlm.nih.gov/BLAST/

BLAST の基本的な使い方はライフサイエンス統合データベースプロジェクトが運営する統合 TV にて動画で解説されていますのでそちらをご参照下さい。下記 URL からアクセスできます。
http://togotv.dbcls.jp/

1.3　GenBank 形式ファイルからの特定遺伝子配列の抽出

GenBank 形式では、配列中のそれぞれの領域がどういうものかという注釈 (annotation) が加えられています。この情報を応用すれば、長大な配列から特定の遺伝子領域のみを抽出することができます。

まず、GenBank 形式のデータファイルをテキストエディタで開いてみて下さい。以下のような内容になっているはずです。

ファイルの内容 1.7　*D. melanogaster* ミトゲノム完全長データ

```
1  LOCUS       NC_001709              19517 bp    DNA     circular INV 06-MAY-2009
2  DEFINITION  Drosophila melanogaster mitochondrion, complete genome.
3  ACCESSION   NC_001709
4  VERSION     NC_001709.1  GI:5835233
5  DBLINK      Project:164
6  KEYWORDS    .
7  SOURCE      mitochondrion Drosophila melanogaster (fruit fly)
8    ORGANISM  Drosophila melanogaster
9              Eukaryota; Metazoa; Arthropoda; Hexapoda; Insecta; Pterygota;
10             Neoptera; Endopterygota; Diptera; Brachycera; Muscomorpha;
11             Ephydroidea; Drosophilidae; Drosophila; Sophophora.
12 REFERENCE   1  (bases 1 to 408; 13319 to 19517)
13   AUTHORS   Lewis,D.L., Farr,C.L. and Kaguni,L.S.
14   TITLE     Drosophila melanogaster mitochondrial DNA: completion of the
15             nucleotide sequence and evolutionary comparisons
16   JOURNAL   Insect Mol. Biol. 4 (4), 263-278 (1995)
17    PUBMED   8825764
18 略
19 FEATURES             Location/Qualifiers
20      source          1..19517
```

```
                     /organism="Drosophila melanogaster"
                     /organelle="mitochondrion"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:7227"
     gene            1..65
                     /gene="trnI"
                     /nomenclature="Official Symbol: mt:tRNA:I | Name:
                     mitochondrial isoleucine tRNA | Provided by: FBgn0013696"
                     /note="tRNA[Ile]"
                     /db_xref="FLYBASE:FBgn0013696"
                     /db_xref="GeneID:261011"
     tRNA            1..65
                     /gene="trnI"
                     /product="tRNA-Ile"
                     /db_xref="FLYBASE:FBgn0013696"
                     /db_xref="GeneID:261011"
略
     gene            240..1263
                     /gene="ND2"
                     /nomenclature="Official Symbol: mt:ND2 | Name:
                     mitochondrial NADH-ubiquinone oxidoreductase chain 2 |
                     Provided by: FBgn0013680"
                     /note="URF2"
                     /db_xref="FLYBASE:FBgn0013680"
                     /db_xref="GeneID:192474"
     CDS             240..1263
                     /gene="ND2"
                     /note="TAA stop codon is completed by the addition of 3' A
                     residues to the mRNA"
                     /codon_start=1
                     /transl_except=(pos:1263,aa:TERM)
                     /transl_table=5
                     /product="NADH dehydrogenase subunit 2"
                     /protein_id="NP_008277.1"
                     /db_xref="GI:5835234"
                     /db_xref="FLYBASE:FBgn0013680"
                     /db_xref="GeneID:192474"
                     /translation="MFNNSSKILFITIMIIGTLITVTSNSWLGAWMGLEINLLSFIPL
                     LSDNNNLMSTEASLKYFLTQVLASTVLLFSSILLMLKNNMNNEINESFTSMIIMSALL
                     LKSGAAPFHFWFPNMMEGLTWMNALMLMTWQKIAPLMLISYLNIKYLLLISVILSVII
                     GAIGGLNQTSLRKLMAFSSINHLGWMLSSLMISESIWLILFFFYSFLSFVLTFMFNIF
                     KLFHLNQLFSWFVNSKILKFTLFMNFLSLGGLPPFLGFLPKWLVIQQLTLCNQYFMLT
                     IMMMSTLITLFFYLRICYSAFMMNYFENNWIMKMNMNSINYNMYMIMTFFSIFGLFLI
                     SLFYFMF"
略
ORIGIN
        1 aatgaattgc ctgataaaaa ggattacctt gatagggtaa atcatgcagt tttctgcatt
略
//
```

これを見れば、FEATURES という項目にどこからどこまでが何という領域か、といった情報が書かれているのが分かります。ORIGIN には実際の塩基配列があります。この FEATURES の内容を任意のキーワードで検索して、該当する領域の配列を ORIGIN の内容から切り出して出力すればいいわけです。これを行うためのコマンド extractfeat が EMBOSS に含まれています。

例えば *trnI* 領域を別ファイルに書き出すには、以下のようにターミナルやコマンドプロンプトでコマンドを実行します。

```
extractfeat -type tRNA -tag gene -value trnI input_file output_file
```

このコマンドを実行すると、tRNA 領域の中で遺伝子名に trnI を含む領域が出力ファイルに FASTA 形式で

書き出されます。同様に、*ND2* 領域を書き出すには以下のようにします。

```
extractfeat -type CDS -tag gene -value ND2 input_file output_file
```

データベースの注釈がきちんとなされていればこれでうまくいきますが、遺伝子名には微妙に表現が異なる記法が使われていることが頻繁にあります。そのような場合は、"ND2 | NAD2"などとスペースと|で区切って複数のキーワードを書き、ダブルクォートで囲ってやることでそれぞれのキーワードに一致する配列が出力されます。これは、複数の領域を一度に書き出したい場合にも使えます。ただし、16S ribosomal RNA などといった、上記のような区切り文字でないスペースを含んだキーワードは使用できません。そのような場合は、事前に配列ファイルを正規表現を用いた検索・置換などを用いて処理しておきます。

また、書き出した領域を増幅できるプライマーを設計したい場合には、その領域の前後 100bp ほどまで含めて書き出したいことがあります。その場合には、以下のように-before オプションと-after オプションを付加します。

```
extractfeat -type CDS -tag gene -value ND2 -before 100 -after 100 input_file output_file
```

1.4　多重配列整列

配列の準備ができたら、整列 (alignment) によって各配列間で相同 (homologous) な領域を検出して揃えてやる必要があります。これは、相同でない形質を比較しても系統樹の推定には役立たないためです。相同とは、「同じ祖先形質に由来する」という意味です。例えば、人間の眼と魚の眼は共通祖先が持っていた眼に由来すると考えられますが、イカやタコの眼はそうではありません。同様に、鳥の翼とコウモリの翼も相同ではありません。ただ、これらが相同でないというのは、我々が系統関係を知っているから分かるのであって、それが無ければそうとは分からないかもしれません。ですから、相同であるか否かと系統樹とは鶏と卵の関係に似ていると言えます。

配列の整列でも同じことが言えます。つまり、系統関係無しには正しい整列ができないのです。そこで、整列と系統推定を同時にやってしまおうという動きもあります (例えば Fleissner *et al.*, 2005; Lunter *et al.*, 2005; Redelings and Suchard, 2005, など) が、膨大な計算を要し、今のところ現実的ではありません。そこで、我々はそこそこ悪くないだろうと思われる「仮の系統樹」を作成し、それに基づいて整列を行い、系統関係に依存していると考えられる信頼性の低い領域は除去して系統推定に用いることにしています。

多重配列整列 (multiple sequence alignment) に最もよく用いられているのが、ClustalW2/X2 (Larkin *et al.*, 2007) ですが、最近は MUSCLE (Edgar, 2004) や MAFFT (Katoh *et al.*, 2005) という高速性や正確性で上回るプログラムが登場し、徐々にこれらへの移行が起きつつあります。ここでは MAFFT を用いた多重配列整列の方法

を説明します。

MAFFT はコマンドラインから実行するプログラムです。使用するには、コマンドプロンプトやターミナルで以下のようにします。入力ファイル・出力ファイル共に FASTA 形式です。

```
mafft --auto input_file > output_file
```

--auto オプションでは、MAFFT が備えているいくつかのアルゴリズムからデータサイズなどに応じて最適なものを自動的に選択してくれます。終了の際のメッセージにどのアルゴリズムを用いたのかが表示されますので、論文にする際にはどれが使われたのかできるだけ書いた方が良いでしょう。

1.4.1 タンパクコード塩基配列の多重配列整列

タンパクコード塩基配列を塩基配列のままで整列すると、翻訳後のアミノ酸の変異を考慮していないため、容易にフレームシフトを起こすギャップが挿入されてしまいます。しかし、現実にはそんな整列結果が妥当であることはほとんどありません。また、遺伝暗号やアミノ酸の物理化学的性質上、起こりやすい・起こりにくい変異はかなり情報が蓄積されていますが、塩基配列の整列ではそのようなことも考慮されません。そこで、いったんアミノ酸配列に翻訳して整列してから、それを逆翻訳 (正確には整列済アミノ酸配列を参照しながら塩基配列を整列) してやることで、多くの場合ただ単純に整列するよりも良い結果が得られます。ここでは多重配列整列に MAFFT を、逆翻訳に EMBOSS に含まれている tranalign を用いる方法を説明します。

まず、翻訳するには各配列でコドン位置が揃っている必要があるため、塩基配列のままで整列をします。

```
mafft --auto input_file > output_file
```

整列したファイルを Unipro UGENE や ClustalX2 などで表示して見てやると、大抵の場合第 3 コドン位置では同義置換ばかりで他のコドン位置よりも変異が激しいためすぐに分かります。変異の多い座位が 3 座位ごとにあるわけです。そこで、第 1 コドン位置が 1 座位目になるように編集して保存します。もし途中から非コード配列になるようであればその領域も削除しておきます。翻訳してから削除しても構いません。もしもコドン位置が分からなかったり、翻訳の向きが分からなかったら、以下のように EMBOSS の sixpack コマンドを使います。

```
sixpack input_file
```

コマンドを実行すると保存先のファイルを聞かれるので適当に名前を付けるかデフォルトのままで保存します。ここで、遺伝暗号が standard ではない場合は、-table オプションでそれを指示してやる必要があります。例えば昆虫のミトゲノム配列であれば invertebrate mitochondrial なので以下のようにコマンドを実行します。

```
sixpack -table 5 input_file
```

-table オプションに指定する番号と遺伝暗号との対応は以下のようになっています。

0. Standard (default)
1. Standard with alternative initiation codons
2. Vertebrate Mitochondrial
3. Yeast Mitochondrial
4. Mold, Protozoan, Coelenterate Mitochondrial and Mycoplasma/Spiroplasma
5. Invertebrate Mitochondrial
6. Ciliate Macronuclear and Dasycladacean
9. Echinoderm Mitochondrial
10. Euplotid Nuclear
11. Bacterial
12. Alternative Yeast Nuclear
13. Ascidian Mitochondrial
14. Flatworm Mitochondrial
15. Blepharisma Macronuclear
16. Chlorophycean Mitochondrial
21. Trematode Mitochondrial
22. Scenedesmus obliquus
23. Thraustochytrium Mitochondrial

sixpack コマンドで出力されるファイルのうち、FASTA 形式配列の方を開くと、入力ファイルの 1 つ目の配列で順方向 3 フレーム、逆方向 3 フレームの全 6 フレームでの翻訳がなされた結果得られた open reading frame (ORF) の配列が保存されています。ORF とは、開始コドンから終止コドンまでの配列です (ここでは実際には終止コドンで区切っただけの配列となっています)。これが最も長くなるのが正しい翻訳結果と考えられます。sixpack コマンドで出力されるもう一つのファイルには、入力ファイルの 1 つ目の配列で 6 フレーム翻訳を行った結果がテキストエディタで見やすく出力されていますのでこちらでも確認できます。末尾に 6 フレームそれぞれでできる ORF 数がありますので、これが少ない方が正しい可能性が高いでしょう。もし読み枠が逆方向だったら、revseq コマンドで必要に応じて逆相補配列に変換することができます。

正しくコドン位置を揃えることができたら、そのファイルから以下のように EMBOSS の degapseq コマンドでギャップを除去してやります。

```
degapseq input_file output_file
```

ギャップを除去したら、以下のように EMBOSS の transeq コマンドを用いてアミノ酸配列に翻訳してやります。ここでも standard 以外の遺伝暗号の場合は-table オプションで遺伝暗号を指定してやって下さい。

```
transeq input_file output_file
```

翻訳したアミノ酸配列ファイルを念のためテキストエディタや多重配列エディタで開いて確認したら、以下のように MAFFT で整列します。

```
mafft --auto input_file > output_file
```

そして、最後に EMBOSS の `tranalign` コマンドでアミノ酸配列から元の塩基配列へ逆翻訳してやります。ここでも standard 以外の遺伝暗号の場合は`-table` オプションで遺伝暗号を指示する必要があります。

```
tranalign nonaligned_nucleotide_sequences aligned_peptide_sequences output_file
```

なお、ここで述べたようにアミノ酸配列の整列に合わせて塩基配列を整列することが常に良いとは限りません。複数回のフレームシフトが起きている場合には塩基のまま整列した方が良いこともあります。そのため、塩基のまま整列した結果と必ず比較して確認するようにして下さい。

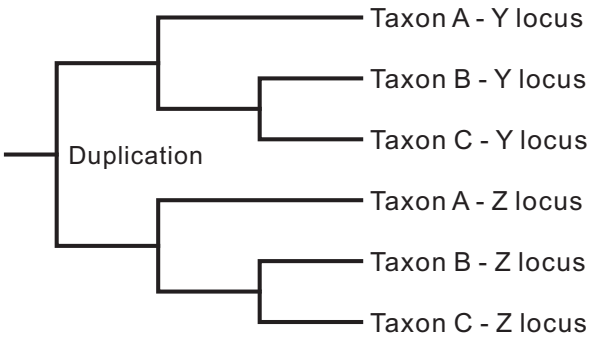
1.5　分子系統樹推定に不適な領域の除去

大抵の整列した配列データには、そのままでは分子系統樹推定に不適な部位を含んでいます。そのため、そのような部位を除去してから系統推定に用いる必要があります。ここでは系統推定に適している・適していないとはどういうデータかを説明した上で、その他の注意点を述べます。

1.5.1　オルソロガスとパラロガス

整列によって得られた相同 (homologous) な配列データセットでも、系統推定に使えるとは限りません。例えば図 1.1 のように Y locus と Z locus が遺伝子重複によって生じた場合を想定すると、Taxon A の Y locus と Taxon B の Y locus、Taxon C の Z locus は相同ではありますが、正しい系統関係 (Taxon B と Taxon C が単系統で Taxon A はその外側) を推定することはできません。このような関係をパラロガス (paralogous) と言います。それに対してそれぞれの Y locus どうしや Z locus どうしの関係はオルソロガス (orthologous) と呼ばれています。系統推定には、オルソロガスなデータセットを用いなけれ

図 1.1　オルソロガスとパラロガスの例



ばなりません。ですから、遺伝子重複が起きたことが分かっている領域はできるだけ系統推定には使わない方が無難です。ただし、重複した遺伝子の全配列を全ての OTU で揃えれば、正しい系統樹を推定可能です。ある程度なら配列が欠けていても何とかなる場合もあるでしょう。

問題は遺伝子重複が起きたかどうかをどうやって知るかですが、これは近縁種で全ゲノムデータが得られていれば、ゲノム内 BLAST で一致度の高い複数の領域が見つからないことを確認すればよいでしょう。BLAST の方が確実だと思いますが、Ensembl genome browser に登録されていれば、こちらでも確認することができます。Ensembl のサイトは以下の URL からアクセスして下さい。
http://www.ensembl.org/

近縁種のゲノムで重複が見つからないからといってオルソログスとは言い切れませんが、これ以上は確認のしようがないので致し方ないでしょう。より多くの領域を用いて系統推定することで信頼性を担保する以外に無いと思います。

1.5.2　仮定を満たしていないデータ

分子系統推定は、様々な仮定を置いて適当にでっち上げた基準で系統樹を評価し、最も良いものを選ぶというものです。ですから、基準そのものの妥当性はさておき、その基準できちんと評価をするにはデータが仮定を満たしている必要があります。この仮定は、最節約法よりも最尤法やベイズ法などのモデルベースの方法の方がより多くなっています。

まず、全ての方法で共通な仮定として、「1 座位の塩基・アミノ酸から 1 座位の塩基・アミノ酸への変異しか含まない」というものがあります (コドン置換モデルの場合は「1 つのコドンから 1 つのコドン」)。分子進化モデルは 1 座位のアミノ酸から複数座位のアミノ酸への変異など想定していませんし、この仮定を満たしていないと最節約法でも変化の回数を過大評価してしまいます。具体的には、開始・終止コドンから別のコドンへの変異とその逆 (1 コドンから複数コドンへの変異とその逆)、イントロン両端のスプライセオソーム認識配列から別の配列への変異とその逆 (非コード配列＝ 0 コドンから複数コドンへの変異とその逆)、フレームシフト・逆位 (複数座位から複数座位への変異)、挿入・欠失 (無から有とその逆) がそれに当たります。ただし、挿入・欠失は整列が信頼できるならギャップをただの missing data として取り扱うことで対処できます (ほとんどのソフトウェアがそういう実装になっています)。

次に、モデルベースの方法が仮定しているものとして「系統樹上で分子進化パターンが共通である」というものがあります。現状の分子系統推定法では系統樹全体で共通の分子進化モデルを当てはめているからです。ただ、そのような仮定をせずに系統樹上で分子進化モデルを変化させることが可能な推定方法もあるにはある (例えば Boussau and Gouy, 2006; Blanquart and Lartillot, 2006, 2008, など) のですが、計算量が膨大だったりするため現状ではほとんど使われていません。遺伝暗号やコドン使用頻度が OTU 間で共通でないタンパクコード塩基

配列はこれらの仮定を満たしていない可能性が極めて高いのでそのようなデータからモデルベースの方法で系統推定を行うのは避けた方が良いでしょう。また、塩基・アミノ酸頻度が OTU 間で共通でない塩基・アミノ酸配列も同様です。塩基・アミノ酸頻度が OTU 間で共通でない塩基・アミノ酸配列は、RY coding (Woese *et al.*, 1991) や Dayhoff coding (Hrdy *et al.*, 2004) を用いて情報を多少捨ててでも無理矢理塩基・アミノ酸頻度を共通にしてしまうか、不均質モデル (Blanquart and Lartillot, 2006, 2008) を当てはめれば解析は可能です。

最後に、既に述べたことともやや重複しますが、同じ分子進化モデルを当てはめた座位間では同じ分子進化パターンに従っていなくてはなりません。ですから、座位ごとに分子進化パターンが異なると予想される場合 (異なる遺伝子座など) には、異なる分子進化モデルを各座位に当てはめるべきです。しかし、異なる分子進化モデルを当てはめれば推定しなくてはならないパラメータが増加してしまいます。開始・終止コドンや、複数の遺伝子に共有されている座位の配列は他とは明らかに異なる選択圧にさらされているはずですから、当然分子進化パターンは異なると予想されます。とは言え、わざわざパラメータ数を増やしてまで異なるモデルを当てはめるほどの情報量は持っていないでしょうから、そのような座位は捨てた方が無難でしょう。

1.5.3　整列の信頼できない座位

偽遺伝子や遺伝子間領域、イントロン、rRNA/tRNA の loop 領域などの欠失や挿入の多い配列では、整列の信頼性が低くなってしまいます。誤って整列された座位は、系統樹推定の際のノイズとなってしまうため、除去した方がよいと言われています (Talavera and Castresana, 2007)。これまでのところ、そのような処理が研究者の経験と勘でなされることが多かったのですが、近年になって自動的に行ってくれるソフトウェアが登場してきました。それが Gblocks (Castresana, 2000)・trimAl (Capella-Gutiérrez *et al.*, 2009)・BMGE (Criscuolo and Gribaldo, 2010) です。ここでは trimAl を用いて整列の信頼できない座位をトリミングする手順を説明します。

trimAl が対応している入力ファイル形式は PHYLIP・FASTA・NEXUS などです。trimAl では、様々なパラメータをユーザーが設定することもできますが、ギャップをそれなりに残す設定とギャップを残さない設定、さらにその 2 つからデータに応じて自動的に選択させることもできます。それぞれの設定によるトリミングは以下のようにして行います。

```
trimal -gappyout -in input_file -out output_file
trimal -strict -in input_file -out output_file
trimal -automated1 -in input_file -out output_file
```

ただし、タンパクコード塩基配列では読み枠がずれないように、コドン単位でのトリミングをする必要があります。trimAl はそこまで考えて処理をしてくれませんが、Phylogears2 の `pgtrimal` コマンドを用いることでそれが可能です。`pgtrimal` は内部で trimAl を呼び出して除去しない座位を得た上で、読み枠がずれないように除去する範囲を拡大します。入力ファイルは NEXUS 形式でなくてはなりません。以下のようにして用います。

```
pgtrimal --frame=1 --method=gappyout input_file output_file
pgtrimal --frame=1 --method=strict input_file output_file
pgtrimal --frame=1 --method=automated1 input_file output_file
```

pgtrimal は--frame オプションがあると入力ファイルをタンパクコード塩基配列として扱います。--frame=1 は配列の 1 塩基目が第 1 コドン位置であるという意味です。--frame=2 であれば 2 塩基目が、--frame=3 であれば 3 塩基目が第 1 コドン位置であるということになります。ただ、この機能が働くということは、フレームシフトが入っているということになるので、そもそも使用するには適していないデータだと考えられます。該当領域を事前に手動でトリミングしておくべきでしょう。

1.5.4　その他の注意点

塩基配列データは、昔は RI、現在は蛍光を検出することで得ているはずです。そのようなデータは、検出されたシグナル強度の波形から読み取られているでしょう。しかし、しばしば波形が重なっていてどの塩基か特定できないことがあります。特に核ゲノムの配列をクローニングせずに直接読んでいる場合にヘテロな個体でよくあることだと思います。このような場合、解析ソフトは表 1.1 のような縮重コード表記を考慮してくれますので、何でもすぐに N にせずに R や Y も積極的に用いた方が良いと思います。ただし、そのような不確実なデータを使わないのが最も安全ではあります。また、ギャップやギャップかどうかもよく分からない missing data はそれぞれ-・?として区別できるようにしておいた方が良いでしょう。

表 1.1　塩基の縮重コード表記

文字	意味
M	A or C (amino)
R	A or G (purine)
W	A or T
S	C or G
Y	C or T (pyrimidine)
K	G or T (keto)
V	A or C or G
H	A or C or T
D	A or G or T
B	C or G or T
N	A or C or G or T

タンパクコード塩基配列の編集の際には、必ず読み枠と翻訳後のアミノ酸配列が変化しないように注意して下さい。読み枠がずれると、コドン位置ごとの異なるモデルの当てはめがうまくいきません。第 2・3 コドン位置と次のコドンの第 1 コドン位置を削除すると、読み枠はずれませんが、後になってアミノ酸配列に変換する必要が生じたときやコドン置換モデルを当てはめようとした場合にうまくいかなくなってしまいますし、ケアレスミスの元なのでこれも避けるべきです。

配列の編集では、削除した座位がすぐに分かるように記録を残しておくとやり直したり削除した座位を確認したりする際に役立ち、ミスを防いだりミスに気付きやすくなります。実際に解析に用いる配列ファイルとは別に、削除した座位を [] などで囲んだファイルを別に保存しておくとよいでしょう。グラフィカルインターフェイスを持った多重整列エディタは便利ですが、そのような記録を残す機能を持っていないものがほとんどでしょうから、個人的には画面内に収まるように配列を折り返した interleaved 形式で保存したファイルをテキストエディタで編集するのが最も良いと思います。多重整列エディタで編集したい場合は、少なくとも何も削除してい

ないファイルも保存しておき、削除後のファイルの配列と比較すればすぐに削除した部分が分かるようにしておくべきでしょう。

また、この先用いる解析ソフトでは、配列名には半角英数字とアンダースコア (_) しか使わない方が無難です。他の文字列を用いていたら、必ず別の配列が同一の名前にならないように注意しながら削除しておきます。後述する TNT では配列名は 31 文字までという制限もありますので注意して下さい。形質が最初の配列と同じであることをピリオド (.) で表す方法がありますが、これも使わない方が安全です。対応したソフトで別形式で書き出すなどして無くしておきましょう。ファイル名にも注意が必要です。やはり半角英数字とアンダースコアしか使わないようにした方が良いでしょう。

1.6 配列が完全一致する OTU の除去

系統解析では配列が完全に一致する複数の OTU (系統樹末端の生物およびその配列) を含んでいると、その OTU が他の OTU より大きく評価されることになり、推定結果に悪影響を及ぼしてしまいます。そのため、完全一致する配列はただ 1 つを残して他は除いておく必要があります。Phylogears2 の `pgelimdupseq` コマンドを用いることで簡単に処理できます。以下のように用います。

```
pgelimdupseq --type=DNA input_file output_file
```

アミノ酸配列では`--type=DNA` の代わりに`--type=AA` を指定して下さい。これによって完全一致する配列はただ 1 つを残して取り除かれます。残される配列の配列名 (OTU 名) は、除去された配列の名前を 2 連続のアンダースコア「__」で連結したものとなります。FASTA・NEXUS・PHYLIP・extended PHYLIP・Treefinder 形式の入力ファイルに対応しています。ただし、PHYLIP 形式は配列名が 10 文字までしか使えませんので特殊な処理を行っています。

ここで、縮重コード文字の取り扱いが問題になってきます。塩基配列では「`A` または `G`」という意味で「`R`」を用います。「`A` または `C` または `G` または `T`」の場合は「`N`」となります。この縮重コード文字がデータに含まれているときに、縮重コード文字をそのままにして全形質が一致しているものだけを完全一致配列とするのか、縮重コード文字を本来の意味通り「`A` または `G`」などと解釈して完全一致配列を探すのか、がまず問題となります。筆者の個人的な意見では後者が妥当であろうと思います。

後者を採用した場合、残す配列では形質を「`A`」とするのか「`R`」とするのかがさらなる問題となります。例えば「`AAA`」と「`ARA`」という配列があった場合、これらは完全一致となりますが、どちらを残すべきかということです。「`R`」が塩基配列決定の信頼性が低いために「`R`」とされているなら、残すのは「`AAA`」でよいでしょう。「`R`」となっている原因がノイズであり、ノイズを捨てることは何ら問題ではないからです。しかし、核 DNA を多数クローンで配列決定を行いコンセンサス配列をデータとしている、または核 DNA をクローニングせずに直

接配列決定して「A」と「G」の両方のシグナルが検出されたために「R」としているのであれば、「ARA」にすべきかもしれません。「R」はノイズによるのではなく意味があるのですから。ただし、「R」には意味があるというのであれば、(あまり好ましくありませんが)「AAA」と「ARA」はやはり両方残すべきということになるかもしれません。pgelimdupseq は、標準では「AAA」を残します。「ARA」を残したい場合は--prefer=degenerate というオプションを入力ファイル名の前に付けて実行して下さい。両方を残したい場合は--prefer=both とします。筆者は pgelimdupseq の標準設定を強く推奨します。なお、pgelimdupseq はギャップを意味する「-」を「?」(missing data,「-または N」の意) として取り扱います。ギャップを意味のある形質として取り扱うには、--gap=another をオプションとして指定します。

1.7　塩基・アミノ酸組成の均一性の検定とデータ改変による均一化

ほとんどの分子進化モデルでは、塩基組成やアミノ酸組成は OTU 間で均一であることが仮定されています。ですから、解析対象のデータがその仮定を満たしているかどうかは解析結果に大きな影響を及ぼします。塩基組成やアミノ酸組成が OTU 間で均一でない場合、本当は単系統ではない OTU 群の単系統性が非常に強く支持されてしまうことがしばしばあります。そのような、仮定を満たしていないデータに基づいてあり得ない単系統性を見いだしている論文が公表されることが未だに後を絶ちません。データ配列において塩基組成・アミノ酸組成の均一性が棄却されないことを確認しておけば、そのような論文を公表せずに済むはずです。Kakusan4・Aminosan もモデル選択前にこの検定を行いますが、Phylogears2 に含まれている pgtestcomposition を用いることで、検定だけを行うことができます。

組成の均一性を検証するにはいくつかの方法がありますが、pgtestcomposition では χ^2 統計量を用いた独立性の検定を利用します。「組成は均一である」が帰無仮説です。これは PAUP*(Swofford, 2003) の BaseFreqs コマンドに実装されているのと同じ方法です。ただし PAUP*では塩基配列にしか適用できませんが pgtestcomposition ではアミノ酸配列にも適用できます。また、PAUP*は「R」なら「A」と「G」がそれぞれ 0.5 回出現などとしてカウントすることで、縮重コード文字を検定統計量の算出に利用しますが、pgtestcomposition は縮重コード文字を一切用いません。この検定法よりも良いとされている Bowker の検定というものもあります (Ababneh *et al.*, 2006) が、その方法ではある条件下では p 値を算出できず、その条件を満たすデータがしばしばあるため今のところは独立性の検定を利用しています。pgtestcomposition でこの検定を行うには、以下のようにコマンドを実行します。

```
pgtestcomposition --type=DNA input_file output_file
```

アミノ酸配列では--type=DNA を--type=AA に置き換えて下さい。対応している入力ファイル形式は FASTA・NEXUS・PHYLIP・extended PHYLIP・Treefinder です。出力ファイルには以下のような情報が出力されます。

ファイルの内容 1.8 χ^2 検定の結果

```
Type of Nucleotides: 4
Number of Taxa: 配列数
Degree of Freedom: 自由度
Total Count: 座位数 * 配列数 - Gap・Missing Data・Ambiguous Dataの数
Chi-square Statistic: chi-square統計量
p-value: p値

               A       C       G       T       rtotal
OTU名      観測値  観測値  観測値  観測値         合計
           期待値  期待値  期待値  期待値

                        中略

ctotal       合計    合計    合計    合計       総合計
```

もしも均一性が棄却されてしまった場合、データを改変することで無理矢理均一化してしまうか、組成の不均一性を許容するモデル (Blanquart and Lartillot, 2006, 2008) を適用した解析を行う必要があります。また、データによっては正確な p 値が算出できないものがあります (Cochran, 1954)。そのようなデータではファイルの末尾にその旨が出力されます。また、この方法では配列が長い場合には過剰に均一性が棄却されやすくなってしまいますので、そのようなデータの取り扱いには注意が必要です。

タンパクコード塩基配列データの第 3 コドン位置や、多遺伝子座配列データのある 1 遺伝子座といった、データの一部分のみに範囲を絞って検定を行うこともできます。例えば以下のコマンドでは、入力ファイルの 1~100 塩基目の範囲だけで検定を行います。

```
pgtestcomposition --type=DNA 1-100 CG TA input_file output_file
```

第 3 コドン位置のみを対象としたい場合には以下のようにします。

```
pgtestcomposition --type=DNA 3-.\3 CG TA input_file output_file
```

ここで、`3-.\3` は、3 塩基目から末尾までの範囲において 3 塩基ごとに (2 塩基間隔で) 対象となる座位を選択するという意味です。

組成の均一性が棄却されてしまった場合、データを改変することで無理矢理均一化することができます。代表的な方法に RY コーディング (Woese *et al.*, 1991) があります。RY コーディングは、塩基配列を対象としたデータ変換の方法です。塩基配列において組成が不均一なのは、AT と GC の比率が OTU によって異なるためであることがよくあります。このようなデータであっても、AG と CT の比率はは OTU 間で均一になっている場合があります。これを利用すれば、形質状態を表す文字を「`A` または `G`」(つまり「`R`」) を表す文字と、「`T` または `C`」(つまり「`Y`」) を表す文字の 2 つだけにしてしまうことで、組成を均一化できます。この方法では `AG` 間、および `TC` 間の変異の情報は捨ててしまうことになりますが、従来の系統樹推定法をそのまま利用できます。Phylogears2 では、`pgrecodeseq` コマンドを用いることでこの処理が容易に可能です。以下のコマンドを実行することで、RY コーディングを適用した配列を得ることができます。

```
pgrecodeseq --type=DNA CG TA input_file output_file
```

このコマンドを実行すると、「C」は「T」へ、「G」は「A」へそれぞれ置換された配列が出力されます。このため、配列は「A」と「T」の 2 文字だけになります (ただし縮重コード文字や「-」と「?」は除く)。RY の 2 文字になるわけではありませんが、効果は同じです。対応している入力ファイル形式は FASTA・NEXUS・PHYLIP・extended PHYLIP・Treefinder です。アミノ酸配列に対して用いられる Dayhoff コーディング (Hrdy *et al.*, 2004) というものもありますが、これは以下のように実行することで適用できます。

```
pgrecodeseq --type=AA STGPNEQKHVILYW AAAADDDRRMMMFF input_file output_file
```

これにより、出力される配列は ADRMFC の 6 文字だけになります。変換後のデータは RAxML (Stamatakis, 2006) や Treefinder (Jobb *et al.*, 2004)、MrBayes (Ronquist and Huelsenbeck, 2003) で一般時間反転可能 (GTR) モデルを適用して解析することができます。絶対に WAG (Whelan and Goldman, 2001) や JTT (Jones *et al.*, 1992) などの経験的置換モデルやそれらの +F モデルを適用してはいけません。従って、Dayhoff コーディングを行ったデータではモデル選択は必要ありません。pgrecodeseq は縮重コード文字も適切に処理するように作成してありますので、縮重コード文字の含まれている配列にもお使いいただけます。ただし、置換前の文字列と置換後の文字列には縮重コード文字を用いることはできませんのでご注意下さい。これはプログラム作成上の都合と、pgtestcomposition が縮重コード文字を統計量の計算に用いないためです。

タンパクコード塩基配列データの第 3 コドン位置や、多遺伝子座配列データのある 1 遺伝子座といった、データの一部分のみにこの処理を適用することもできます。範囲の指定方法は pgtestcomposition と同様です。データの一部分にのみ均一化の処理を行った場合、処理した部分と処理していない部分は異なるパーティションとし、比例または分離モデルを適用する必要があります。そうしないと、塩基組成や置換速度のパラメータが正しく推定できなくなるためです。また、データの変換ができたら、pgtestcomposition を用いて組成が OTU 間で均一になっていることを確認してから実際の解析に用いるようにご注意下さい。

第 2 章

分子進化モデルの基礎

分子進化モデルは塩基配列データに当てはめられる塩基置換モデル (nucleotide substitution model) と、アミノ酸配列データに当てはめられるアミノ酸置換モデル (amino acid substitution model) に大別されます。タンパクコード塩基配列データにおいて同義置換 (synonymous substitution) と非同義置換 (nonsynonymous substitution) を区別するコドン置換モデル (codon substitution model) というものもあります。コドン置換モデルはパラメータが多く計算が大変なのと対応しているソフトウェアが少ないため今のところあまり使われていませんが、よりリアルな確率過程を表しているため、将来的にはタンパクコード領域ではコドン置換モデルが多用されるようになっていく可能性は高いでしょう。しかし、ここでは塩基置換モデルとアミノ酸置換モデルに絞って説明を進めていきます。

2.1 塩基置換モデル

2.1.1 塩基置換速度行列

塩基置換速度行列 (nucleotide substitution rate matrix) は、座位 (site) 内における、形質状態 (character state) 間の移行速度の不均質性 (heterogeneity) を表現するものです。表 2.1 のように表すことができます。

表 2.1 塩基置換速度行列

From ＼ To	A	C	G	T
A	-	$Rate_{AC}Freq_C$	$Rate_{AG}Freq_G$	$Rate_{AT}Freq_T$
C	$Rate_{AC}Freq_A$	-	$Rate_{CG}Freq_G$	$Rate_{CT}Freq_T$
G	$Rate_{AG}Freq_A$	$Rate_{CG}Freq_C$	-	$Rate_{GT}Freq_T$
T	$Rate_{AT}Freq_A$	$Rate_{CT}Freq_C$	$Rate_{GT}Freq_G$	-

ここで、$Rate_{XY}Freq_X$ は塩基 Y から塩基 X への移行速度で、$Freq_X$ は塩基 X の頻度です。ただし、$Rate_{XY} = Rate_{YX}$ とします (これを「時間反転可能」[time-reversible] と言います)。

$Rate_{AC} = Rate_{AG} = Rate_{AT} = Rate_{CG} = Rate_{CT} = Rate_{GT}$ であり、かつ $Freq_A = Freq_C = Freq_G = Freq_T$ のとき、最も単純な JC69 モデル (Jukes and Cantor, 1969) となります。$Rate_{AG} = Rate_{CT} \neq Rate_{AC} = Rate_{AT} = Rate_{CG} = Rate_{GT}$ であり、かつ $Freq_A = Freq_C = Freq_G = Freq_T$ なモデルは K80/K2P モデル (Kimura, 1980) です。$Rate_{AC} = Rate_{AG} = Rate_{AT} = Rate_{CG} = Rate_{CT} = Rate_{GT}$ であり、かつ $Freq_A \neq Freq_C \neq Freq_G \neq Freq_T$ なモデルは F81 モデル (Felsenstein, 1981) と呼ばれています。$Rate_{AC} \neq Rate_{AG} \neq Rate_{AT} \neq Rate_{CG} \neq Rate_{CT} \neq Rate_{GT}$ であり、かつ $Freq_A \neq Freq_C \neq Freq_G \neq Freq_T$ なモデル (Tavaré, 1986) は一般時間反転可能 (general time-reversible を略して GTR) モデルと呼ばれています (Posada and Crandall, 1998)。他にも様々なモデルがありますが、全て GTR モデルの下位互換なモデルとなっています。

この後説明する系統推定の際には、一般的に無根系統樹を仮定して系統推定を行います。そのため、分子進化モデルは時間反転可能なモデルでなくてはなりません (そうでないと尤度が定義できない)。これは、時間反転不能モデルは有根系統樹でしか適用できないのですが、そのためには数値計算の困難さと樹形空間の拡大などの問題があり現実的には難しいためと思われます。

2.1.2　座位間の置換速度不均質性

座位 (site) 間における置換速度の不均質性 (heterogeneity) があることが知られており、これを表すモデルがいくつか提案されています。これらは ARSV (among-site rate variation) モデルと呼ばれています。

配列データ内では、置換の滅多にない座位がほとんどであり、置換が頻発する座位は限られています。これに Γ 分布を当てはめるものが提案されています (Yang, 1993)。しかし、連続的な Γ 分布を当てはめるのは計算量が膨大になるため、Γ 分布に基づいて任意の数に座位をカテゴリ分けするモデル (Yang, 1994) が最もよく利用されています。これを + G とか + dG (discrete Gamma の意) などと表記します。カテゴリ分けする数を含めて + dG_4 などと表記することもしばしばあります。

また、置換の起きない座位 (invariable site) と置換が起きる座位 (variable site) の 2 つにカテゴリ分けするモデル (+ I と表記) や、 + G と + I を併用したモデルもあります。これらは一定の法則に従って自動的に行われるカテゴリ分けですが、解析者が任意のカテゴリ分け (partitioning) を指定することもできます (+ SS [site specific rate の意] と表記)。異なるコドン位置 (codon position) や遺伝子座などの置換速度は異なる可能性が高いので、これらがしばしばカテゴリとして指定されます。この場合、単に + SS と表記しても分かりづらいので、 + Codon Position Specific Rate とか + Gene Specific Rate と表記した方が良いでしょう。さらに、これらのカテゴリ内で + G や + I モデルを当てはめることも可能です (ただし、実際には + I モデルを併用できるソフトウェアは存在しません)。 + Codon Position Specific Rate と + G を併用する場合、コドン位置それぞれに Γ 分布を当てはめる

こともできます (+ 3 Different Gamma) し、共通の Γ 分布を当てはめることも可能です (+ 1 Shared/Common Gamma)。同様に、遺伝子座間でもそれぞれに Γ 分布を当てはめる場合 (+ N Different Gamma) と共通の Γ 分布を当てはめる場合 (+ 1 Shared/Common Gamma) があり得ます。隣接する座位間の置換速度の相関を + G モデルに取り入れた + adG (autocorrelated discrete Gamma の意) モデルもあります (Yang, 1995)。

2.1.3 Mixed model

前節では座位 (site) 間での置換速度不均質性のみを考慮していましたが、塩基置換速度行列および置換速度不均質性の不均質性を考慮することも可能です。つまり、任意の座位のグループ＝パーティション (partition) ごとに異なる塩基置換速度行列、異なる ASRV モデルを当てはめます。これを mixed model と呼んでいます。論文によっては区分モデル (partitioned model) と呼んでいることもあります。これに対して、パーティション間に共通の塩基置換速度行列と ASRV モデルを当てはめるものは非区分モデル (nonpartitioned model) と呼ばれます。

Mixed model には大きく分けて 2 つのモデルが含まれています。1 つ目はパーティション間での平均置換速度のばらつきを考慮した比例モデル (proportional model) で、もう 1 つはパーティション間で置換速度の変化が独立している分離モデル (separate model) です。比例モデルでは、系統樹の枝の長さがパーティション間で相似的になっていますが、分離モデルでは系統樹の枝長はパーティションごとに全く独立しています。比例モデルでは枝長パラメータは増加しませんが、パーティション数-1 個のパーティション間の枝長比=置換速度比パラメータの推定が必要になります。分離モデルは枝長パラメータがパーティション数倍の膨大な数となってしまうため、多くの場合比例モデルが用いられます。ASRV モデルの中で説明した + SS モデルは、パーティション間で置換速度行列も ASRV モデルも共通にしつつ比例モデルを当てはめたのと同じものになります。

2.2 アミノ酸置換モデル

2.2.1 Empirical model

塩基置換速度行列は 4x4 の行列でしたが、アミノ酸置換速度行列は 20x20 の行列となるため、$Rate_{XY}$ と $Freq_X$ の数は時間反転可能モデルでも 190 + 20 = 210 となり膨大です。そこで、既に系統関係の分かっている分類群間の系統樹において、大量のデータを用いてあらかじめ推定された $Rate_{XY}Freq_X$ の値を用いたモデルをアミノ酸置換モデルとして用います。これらは、実際のデータから観測された「経験的な」ものなので、empirical model と言います。核 (Dayhoff *et al*., 1978; Henikoff and Henikoff, 1992; Jones *et al*., 1992; Müller and Vingron, 2000; Whelan and Goldman, 2001; Veerassamy *et al*., 2003; Le and Gascuel, 2008)・ミトコンドリア (Adachi and Hasegawa, 1996; Cao *et al*., 1998; Abascal *et al*., 2007)・葉緑体 (Adachi *et al*., 2000)・レトロウィルス (Dimmic *et al*., 2002; Nickle *et al*., 2007) のアミノ酸配列用に様々なモデルが提案されています。$Rate_{XY}$ は既

存の empirical model の値を用い、アミノ酸頻度 $Freq_X$ はデータから推定するモデルも + F モデルと呼ばれて広く用いられています。

2.2.2 Mixed empirical model

Empirical model は所詮 empirical model に過ぎないため、たとえ選択されたとしても手元のデータに本当に適したモデルではない可能性が十分考えられます。だからと言って、20x20 の行列のパラメータを推定しながら系統推定を行うのは現在のコンピュータの演算能力では困難ですし、近い将来も無理でしょう。そこで、複数の empirical model を重み付け平均化するモデルが提案されています (Jobb, 2008)。平均化の際の重み付けはモデルのパラメータとしてデータから推定します。このモデルは今のところ Treefinder にしか実装されていません。しかし、MrBayes にも類似のアプローチが実装されています (Ronquist *et al*., 2005)。MrBayes では、empirical model を重み付け平均化するのではなく、model jumping という方法によって解析中に適用するモデルを変更していきます。それによって、各モデルの適用された事後確率が得られます。また、モデル平均化 (model averaging) によっても類似の効果を得られるかもしれませんが、上述の 2 つの実装では樹形探索を行いながら平均化パラメータを最適化する、もしくは適用するモデルを切り替えて最適化するというところが大きく異なります。

2.2.3 Mixed model

塩基置換モデルと同様に、ソフトウェアによってはパーティションごとに異なるアミノ酸置換モデルを当てはめる mixed model を適用することができます。枝長パラメータの扱いに応じて比例モデルと分離モデルがあるのも同様です。

第 3 章

分子進化モデルの選択

3.1 モデル選択の必要性

尤度計算の際に当てはめるモデルは複雑なものほど当てはまりは良くなりますが、実際にはデータにはノイズが含まれており、ノイズにまでフィットしてしまっても意味が無いどころか有害ですらあります。例えばデータを同じ母集団から採取し直したときに当てはまりが大きく低下してしまうようなら、そのモデルは母集団のパラメータを表しているとは言えません。そこで、パラメータを無制限に増やすのではなく、パラメータ数の増大というコストと尤度の向上という利益のバランスを取る必要が出てきます。それを実現したのが Akaike (1974) によって提案された赤池情報量規準 (Akaike information criterion を略して AIC) です。AIC は尤度を L、パラメータ数を k としたときに以下の式で表されます。

$$\mathrm{AIC} = -2\ln L + 2k \tag{3.1}$$

この AIC の値が最小となるモデルが最もバランスの取れたモデルであるということが理論的に導かれています。これを利用して最適なモデルを選択してやればよいわけです。しかし、AIC はサンプルサイズが無限大の理想的なデータを前提とした近似によって導かれています。実際のデータはサンプルサイズ無限大ということはあり得ませんので、サンプルサイズが小さいときに AIC がパラメータ数の増大コストを過小評価してしまうことを正規分布を仮定して補正した AICc が Sugiura (1978) によって提案されています。AICc はサンプルサイズを n としたときに以下のように表されます。

$$\mathrm{AICc} = -2\ln L + 2k \times \frac{n}{n-k-1} \tag{3.2}$$

ここで、分子進化モデルの選択に対する AICc の適用には「正規分布を仮定して補正した」という点が問題になります。分子進化モデルの誤差構造が正規分布ではないからです。また、$n-k-1$ が 0 以下のとき、AICc は算出することができません。これは、そのようなモデルはそのデータには適用すべきでないと考えるべきなのかもしれません。

また、「ベイズ的」と言われている BIC というものもあります (Schwarz, 1978)。これは以下の式で表されます。

$$\mathrm{BIC} = -2 \ln L + k \times \ln n \tag{3.3}$$

このように規準が複数あると、どれを使えばいいのかという話になりますが、筆者は最尤系統推定で用いるモデルの選択には AIC か AICc、ベイジアン系統推定で用いるモデルの選択には BIC を使うことにしています。AIC か AICc かは、それぞれ「サンプルサイズ無限大を前提としている」、「正規分布を仮定している」という問題があり、一長一短があります。筆者は全ての候補モデルで AICc が算出可能、つまり $n - k - 1 > 0$ であれば AICc を、そうでなければ AIC を使うことにしています。

本来、系統推定においては分子進化モデルの選択と系統樹の選択は同時に行われるべきですが、1 つの分子進化モデルにおいてさえ、系統樹の選択は大変な労力を要します。そのため、全ての分子進化モデルでそうするのは非現実的です。そこで、とりあえずそれほど悪くはないであろうと考えられる「仮の」系統樹に樹形を固定して (ソフトによっては簡易な樹形探索も行う)、各分子進化モデルにおける最大化対数尤度を計算し、それに基づく情報量規準によって分子進化モデルの選択を行います。その後、選択された分子進化モデルを適用して系統樹の選択を行います。つまり、現状の分子系統推定は「多重モデル選択」となっているわけです。model jumping などによりいつかはこれは解決されるかもしれませんが、しばらくの間はこの方法が使われ続けることになるでしょう。

この問題があるため、系統推定によって最終的に得られた系統樹でも、同じ分子進化モデルが選択されるのかを確認することが望ましいでしょう。もし異なるなら、選択された分子進化モデルを適用して再度系統推定を行う必要があります。ただ、何度やってもモデル選択結果と系統推定結果が一致しない可能性があります。その場合は複数の暫定最尤系統樹の中でモデル選択に用いた規準の値が最も小さいものを使うか、いずれにおいても共通している部分についてのみ考察するしかないでしょう。

3.2　Kakusan4・Aminosan による分子進化モデルの選択

Kakusan4 (Tanabe, 2007)・Aminosan は、配列データに対して最適な置換モデルを選択し、Treefinder や MrBayes (MrBayes5D) 用のモデル設定ファイルを書き出してくれるソフトウェアです。また、モデル選択に必要な尤度の計算は PAUP*・baseml・Treefinder のいずれか (Aminosan は Treefinder か codeml) に丸投げして計算させますが、この際に並列化して各ソフトウェアを起動するため、マルチ CPU やマルチコア CPU を搭載したコンピュータでは従来のソフトウェアよりも大幅に高速な処理が可能になっています。対応している入力データは、FASTA・NEXUS・PHYLIP・GenBank などの配列ファイルです。モデル選択に利用できる情報量規準は AIC (Akaike, 1974)・AICc (Sugiura, 1978)・BIC (Schwarz, 1978) となっています。

Kakusan4 と Aminosan は、以下のような処理を行っています。

1. χ^2 独立性の検定による塩基・アミノ酸頻度の均一性確認
2. 固定する仮の系統樹を作成 (指定も可能)
   - JC69 距離の近隣結合樹 (Kakusan4)
   - K83 距離の近隣結合樹 (Aminosan)
3. 領域・コドン位置ごとに候補モデルを当てはめて最大化対数尤度を求める
4. 領域・コドン位置ごとに情報量規準を算出してモデル選択
5. 領域・コドン位置ごとに選択されたモデルを適用した比例・分離モデルを全領域連結配列に当てはめて最大化対数尤度を求める
6. 情報量規準を算出して非区分・比例・分離モデルからのモデル選択を行う

このように、一旦各領域・各コドン位置ごとのモデル選択を行ってから連結配列での非区分・比例・分離モデルからのモデル選択をしているため、ここでも多重モデル選択になっています。また、比例・分離モデルは本来、各領域・各コドン位置に全候補モデルを当てはめる全ての組み合わせからモデル選択すべきですが、計算量的に非現実的なので、各領域・各コドン位置のモデル選択で選ばれたモデルを用いた比例・分離モデルを当てはめることで妥協しています。

Kakusan4・Aminosan には 2 つの動作モードがあります。1 つ目は誰でも簡単に利用できる (つもりで作った) 対話型の動作モードで、2 つ目は自動処理に適したコマンドラインから操作する動作モードです。ここでは対話型の動作モードでの操作方法を説明していきます。主に Kakusan4 を用いて説明していきますが、Aminosan では異なる点があれば適宜説明を加えていきます。また、現在のところ Aminosan は mixed empirical model には対応していません。

3.2.1 モデル選択の実行

いずれの環境においても、Kakusan4・Aminosan を起動すると標準で対話モードになります。対話モードでは最初に入力ファイルの名前を質問されます。

```
Kakusan4 4.0.2010.10.27
========================================================================

This is a script to select nucleotide substitution model for multi-
partitioned data set. Official web site of this script is
http://www.fifthdimension.jp/products/kakusan/ .
To know script details, see above URL.

Copyright (C) 2006-2010  Akifumi S. Tanabe

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License.

This program is distributed in the hope that it will be useful,
```

```
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Parsing command line options...
No input files are specified.
Entering interactive mode.
Specified options are ignored.
Specify an input file name.
Note that you can use wild card.
```

Windows (Vista 以外)・MacOS X 環境では、ここでファイルのアイコンを 1 つだけこのウィンドウにドロップすると、ファイルのフルパス名が入力されます。Windows Vista では、エクスプローラ上で Shift キーを押しながらファイルアイコンを右クリックして**パスとしてコピー**をしてから、タイトルバーを右クリックし、**編集**の中にある**貼り付け**を行って下さい。

```
Specify an input file name.
Note that you can use wild card.
"C:\Users\akifumi\Desktop\SampleData\CYTBnuc_P.fas"
```

そのまま Enter キーを押すとファイルが Kakusan4・Aminosan に読み込まれます。

```
"C:\Users\akifumi\Desktop\SampleData\CYTBnuc_P.fas"
"C:\Users\akifumi\Desktop\SampleData\CYTBnuc_P.fas" was accepted.
Specify an input file name or just press enter to leave input file specification.
```

複数領域データなどの場合、領域ごとに別のファイルとして用意しておき、この操作を繰り返して全ファイルを読み込ませます。同一の遺伝子配列でも、タンパクコード領域とそうでない領域は必ず別のファイルに分けて下さい。タンパクコードでない領域の中で、イントロンと 5'・3' の非翻訳領域は分けるべきかどうかは正直分かりません。場合によるでしょう。なお、タンパクコード領域塩基配列データではファイル名 (拡張子は含まない) が必ず_P で終わるようにして下さい。こうすることで、コドン位置ごとの置換速度の不均質性やコドン位置ごとに異なる塩基置換モデルを当てはめる mixed model が検討されるようになります。また、複数領域データでは、各ファイルでの配列名が統一されていなくてはなりませんので注意して下さい。入力ファイルの指定の際には*や?といったワイルドカードが使えます。ワイルドカードを用いることで一度に多数のファイルを読み込ませることができます。

また、Aminosan では、ファイル名が_mt で終わるようにすると、検討対象モデルを mtREV (Adachi and Hasegawa, 1996)・mtMam (Cao *et al*., 1998)・mtArt (Abascal *et al*., 2007)・mtZoa (Rota-Stabelli *et al*., 2009) のみに制限できます。同様に_nc で Dayhoff (Dayhoff *et al*., 1978)・JTT (Jones *et al*., 1992)・BLOSUM62 (Henikoff and Henikoff, 1992)・VT (Müller and Vingron, 2000)・WAG (Whelan and Goldman, 2001)・PMB (Veerassamy *et al*., 2003)・LG (Le and Gascuel, 2008) のみに、_cp で cpREV (Adachi *et al*., 2000) のみに、_rt で rtREV

(Dimmic *et al*., 2002)・HIVb・HIVw (Nickle *et al*., 2007) のみに検討モデルが絞られます。_**モデル名**で特定モデルのみに絞ることもできます。ただし、 + F, + G, + I の適用は検討されます。なお、Aminosan が検討する Dayhoff・JTT モデルは、Kosiol and Goldman (2005) による DCMut バージョンと呼ばれる若干の改善が施されたものであることに注意して下さい。

全てのデータファイルを読み込ませたら、何も入力せずに Enter キーを押します。

```
Specify an input file name or just press enter to leave input file specification.

OK. Input file specification have terminated.

Log, result and configuration files will be output to "C:\Users\akifumi\Desktop\
SampleData\CYTBnuc_P.fas.kakusan".
```

以上のメッセージの通り、最初に与えたファイルの存在するフォルダ内に「**最初に与えたファイルのファイル名**.kakusan」という名前のフォルダ (Aminosan の場合は末尾は aminosan になります) が作成され、そこに全ての結果が出力されます。続いて、どの系統推定ソフトウェア向けのモデル選択を行うのか尋ねてきます。

```
OUTPUT OPTIONS

Which is a target analysis software? (MrBayes/Treefinder/PAUP/PHYML/RAxML)
(default: Treefinder)
```

この質問では、選択した系統推定ソフトウェア向けのモデル設定ファイルが出力されるように設定されます。タンパクコード領域配列データを入力して Treefinder か RAxML か MrBayes を選択すると、コドン位置ごとに異なるモデルを当てはめる mixed model の検討が強制的に有効になります。RAxML の場合はコドン位置間で共通のモデルを当てはめることも強制的に検討されます。PAUP*または PHYML を選択した場合、全領域連結配列を区分せず共通なモデルを当てはめることが強制的に検討されます。これは PAUP*と PHYML が mixed model に対応していないからです。この後の質問は、これまでの返答によって内容が変化します。全ての質問に関して説明していきますが、表示されない質問がある場合がありますのでご注意下さい。

次の質問は、コドン位置ごとに異なる塩基置換モデルを当てはめる mixed model を検討するか否かに関するものです。ただし、タンパクコード領域データを入力していない場合や、強制的に検討される場合には質問が表示されません。また、当然ですが Aminosan でもこの質問はされません。

```
ANALYSIS OPTIONS

You input protein coding sequence.
Do you want to consider partitioning of codon positions? (y/n)
(default: n)
```

この質問に y と答えて Enter キーを押せば、コドン位置ごとに最適な塩基置換モデルの選択が行われます。

次の質問は、タンパクコード領域において全コドン位置を区分せず共通なモデルの検討を行うか否かに関する

ものです。ただし、タンパクコード領域データを入力していない場合や、コドン位置ごとに最適なモデル選択を行う設定が有効になっていない場合は表示されません。PAUP*や PHYML 用の設定ファイル出力が有効になっている場合にも表示されません。もちろん、Aminosan でもこの質問はされません。

```
You enabled partitioning of codon positions.
Do you want to consider nonpartitioning of codon positions? (y/n)
If you say yes, applying nonpartitioned models to all-codon position-concatenate
d sequences will be considered on each locus.
(default: n)
```

この質問に n と答えるか、何も入力せずに Enter キーを押した場合、タンパクコード領域において全コドン位置に共通なモデルの検討は行われません。y と答えれば検討されます。

次の質問は、複数領域データを与えている場合に、全領域連結配列に領域を区分しないモデルを当てはめることを検討するか否かに関するものです。複数領域データを与えていない場合や、PAUP*または PHYML 用の設定ファイル出力を有効にしている場合は強制的に有効になるので表示されません。

```
You input multiple files.
Do you want to consider nonpartitioning of loci? (y/n)
If you say yes, applying nonpartitioned models to all-loci-concatenated sequence
s will be considered.
(default: n)
```

この質問に y と答えて Enter キーを押せば、全領域連結配列に領域を区分しないモデルを当てはめることが検討されますが、n と答えれば検討されません。

次は複数領域データかタンパクコード領域データを与えたときに、全領域連結配列または全コドン位置連結配列における非区分・比例・分離モデル間の比較を行うかに関する質問です。この質問は複数領域データかタンパクコード領域データを与えていないと表示されません。PAUP*または PHYML 用設定ファイル出力を有効にしている場合は表示されません。また、RAxML 用設定ファイル出力を有効にしている場合は比例モデルが検討されないので文言が異なります。

```
You input multiple files or protein coding sequence.
Do you want to compare nonpartitioned, proportional and separate models on all-
loci concatenated sequences? (y/n)
Note that this function needs Treefinder.
(default: y)
```

y と答えるか、何も入力せずに Enter キーを押せば非区分・比例・分離モデル間の比較が行われます。分離モデルの対数尤度は各領域の対数尤度の和なので簡単に求められますが、比例モデルの尤度は実際に当てはめて計算しなくてはならないため、計算量が増加します。また、比例モデルの尤度は次の質問の内容にかかわらず Treefinder で計算されます。そして、Treefinder 以外で計算した尤度との互換性が厳密にあるのか何とも言えないので、他の尤度計算に Treefinder 以外が使われていた場合は Treefinder で尤度を計算し直すためさらに計算量

が増加します。また、Treefinder が + SS モデルには対応していないためこれは比較対象に含まれていません。つまり、非区分モデルやコドン位置間非区分モデルが選択されなかったとしても、 + SS モデルを検討していないせいである可能性があるので注意して下さい。

次の質問は、モデル選択に用いる尤度の値をどのプログラムで計算させるかというものです。PAUP*・baseml (Aminosan では codeml)・Treefinder のいずれかから選びます。

```
Which do you want to use the program for likelihood calculation? (baseml/tf/paup)
(default: baseml)
```

baseml と答えれば、baseml が各モデルの尤度最大化に使われます。tf と答えれば Treefinder が、paup と答えれば PAUP*が用いられることになります。Treefinder または MrBayes 用設定ファイル出力が有効で、非区分・比例・分離モデルの比較を行う設定にした場合には Treefinder がデフォルトで、それ以外の場合は baseml (codeml) がデフォルトです。PAUP*・PHYML 用の設定ファイル出力を行う場合は PAUP*を、MrBayes・Treefinder 用の場合は Treefinder を用いることを推奨します。RAxML 用の場合はいずれでも構いません。

次の質問は、塩基頻度パラメータを持つモデルにおいて、各塩基頻度パラメータを最適化するか、それともデータから得られる観測値を用いるのかに関するものです。

```
Do you want to optimize the parameters of base composition? (y/n)
(default: n)
```

n と答えるか、何も入力せずに Enter キーを押すと、最適化が無効になり、データから得た観測値が用いられます。最適化は行われません。y と答えると最適化が行われます。最適化を行うと時間はかかりますがより厳密な解析が行われます。しかし、塩基配列でデータが十分にある場合は最適化の効果はあまりありませんので無効にしても構わないでしょう。アミノ酸配列では形質状態が 20 もあるため、最適化した方が良いことも多いと思いますが、最適化ができるのは Treefinder で尤度を計算する場合のみです。その場合も Treefinder や MrBayes 用の設定ファイルを出力させるときしかこの質問は表示されません。

次に、座位間の置換速度不均質性に対する離散 Γ 分布の当てはめにおいて、離散化の際のカテゴリ数に関する質問がなされます。

```
How many rate categories of discrete gamma rate heterogeneity do you want to con
sider? (integer)
(default: 8)
```

この質問には、正の整数で答えます。少なくとも 4 以上の値を入力するようにして下さい。値を大きくするほど尤度は正確になりますが計算時間が延びていきます。

次の質問は、ASRV に + I モデルの当てはめを検討するか否かに関するものです。PAUP*か Treefinder で尤度を計算する設定のときにのみ表示されます。

```
Do you want to consider invariant model for among-site rate variation? (y/n)
(default: n)
```

デフォルトでは n ですが、検討させたい場合には y と答えて下さい。

次の質問は、領域・コドン位置ごとに異なる離散 Γ 分布の当てはめを行うモデルを検討するか否かに関するものです。なお、この質問は尤度最大化に baseml を用いる場合にしか表示されません。

```
Do you want to consider N-GAM model for among-site rate variation? (y/n)
Note that this model is very time-consuming.
(default: n)
```

y と答えて Enter キーを押せば、領域・コドン位置ごとに異なる離散 Γ 分布の当てはめを行うモデルが検討されますが、このモデルの尤度最大化には非常に時間がかかるため注意して下さい。この質問で n と答えても、比例モデルや分離モデルで領域・コドン位置ごとに異なる離散 Γ 分布を当てはめるモデルは検討されます。

次に、隣接座位間の置換速度自己相関を考慮した離散 Γ 分布の当てはめを行うモデルを検討するか否かに関する質問がなされます。なお、この質問は尤度最大化に baseml を用いる場合にしか表示されません。

```
Do you want to consider autocorrelated discrete gamma model for among-site rate
variation? (y/n)
Note that this model is very time-consuming.
(default: n)
```

y と答えて Enter キーを押せば、隣接座位間の置換速度自己相関を考慮した離散 Γ 分布の当てはめを行うモデルが候補モデルに含まれるようになりますが、このモデルの尤度最大化には非常に時間がかかるため注意して下さい。このモデルはデータによっては尤度の改善に大きな効果があるのですが、それ以上に計算に膨大な時間がかかってしまいます。計算時間がそれほど問題にならない小さめのデータセットでは有効にしてもよいでしょう。

次に、領域ごとに異なる樹形を用いて尤度最大化を行うか、共通の樹形を用いるかの質問がなされます。なお、この質問は複数領域のデータを与えた場合にしか表示されません。

```
Do you want to use different tree topology for parameter optimization on each lo
cus? (y/n)
(default: n)
```

この質問に y と答えて Enter キーを押せば、各領域で異なる樹形に基づいてモデル選択が行われますが、n と答えるか、何も入力せずに Enter キーを押した場合は全領域連結配列データから生成された樹形に基づいてモデル選択が行われます。領域間の不調和 (incongruence) について検討する場合には y と答えて下さい。樹形は次の質問で樹形ファイルを指定しない限り、JC69 距離 (Aminosan では K83 距離 [Kimura, 1983]) に基づいて近

隣結合法 (neighbor-joining [Saitou and Nei, 1987]) によって生成されます。対話モードではなくコマンドラインから用いる場合は他の方法も用いることができます。

次に、尤度最大化に用いる樹形を指定するか否かに関する質問です。

```
If you want to give tree(s) for parameter optimization, specify an input file na
me.
Otherwise, just press enter.
```

もしも尤度最大化に用いる樹形を指定したい場合には、ここで Newick か NEXUS 形式の樹形ファイルを指定して下さい。その必要が無ければ、そのまま Enter キーを押して下さい。

最後に、同時に起動するプロセスの数に関する質問がなされます。

```
How many processes do you want to run simultaneously? (integer)
(default: 1)
```

ここで、任意の正の整数を入力して Enter キーを押すと、入力した数だけプロセスが同時起動されます。指定する値は、基本的にはお使いの PC が搭載している CPU(コア) の数と同数にして下さい。そうすることで、PC の演算能力を最大限に生かすことができます。

以上の全ての質問に答え終えると、以下のような表示がなされます。

```
All configurations have been completed.
Just press enter to run!
```

心の準備ができたら Enter を押して解析を始めて下さい。解析は場合によっては長時間かかってしまいますが、気長に待っていて下さい。

3.2.2 モデル選択結果を見る

既に述べた通り、最初に与えたファイルの存在するフォルダ内に「**最初に与えたファイルのファイル名.kakusan**」(Aminosan からの出力では末尾は aminosan) という名前のフォルダ (以降、「出力フォルダ」と呼びます) が作成され、そこに全ての結果が出力されます。下図のように、出力フォルダ内には Chisq・Results・MrBayes・PAUP・PHYML・RAxML・Treefinder・Scores・Logs というフォルダが作成され、さらにその中に様々なファイルが出力されます。

出力フォルダ
├ Chisq
│ ├ chisq_partition.txt（**各領域のカイ二乗検定の結果**）
│ └ ...
├ Results
│ ├ partition_criterion.txt（**各領域におけるモデル選択の結果**）
│ ├ whole_criterion_comparemix.txt（**連結配列における非区分・比例・分離モデルからの選択結果**）

```
│  └ ...
├ MrBayes
│  ├ partition_criterion_xxx.nex（各領域データと選択されたモデルを適用するコマンドの書かれたNEXUSファイル）
│  └ ...
├ PAUP
│  ├ partition_criterion.nex（各領域データと選択されたモデルを適用するコマンドの書かれたNEXUSファイル）
│  └ ...
├ PHYML
│  ├ partition.phy（各領域データ）
│  ├ partition_criterion_singlesearch.bat（単一の樹形探索を行うバッチファイル）
│  ├ partition_criterion_shotgunsearch.bat（ショットガン樹形探索を行うバッチファイル）
│  ├ partition_criterion_bootstrap.bat（ブートストラップ解析を行うバッチファイル）
│  ├ partition_criterion_shotgunbootstrap.bat（ショットガンブートストラップ解析を行うバッチファイル）
│  └ ...
├ RAxML
│  ├ partition.phy（各領域データ）
│  ├ partition_criterion_xxx.partition（各領域データに選択されたモデルを適用する設定ファイル）
│  ├ partition_criterion_xxx_singlesearch.bat（単一の樹形探索を行うバッチファイル）
│  ├ partition_criterion_xxx_shotgunsearch.bat（ショットガン樹形探索を行うバッチファイル）
│  ├ partition_criterion_xxx_bootstrap.bat（ブートストラップ解析を行うバッチファイル）
│  └ ...
├ Treefinder
│  ├ partition_xxx.tf（各領域データ）
│  ├ partition_criterion_xxx.model（各領域データに選択されたモデルを適用する設定ファイル）
│  ├ partition_criterion_xxx.rates（比例・分離を指定する設定ファイル）
│  ├ partition_criterion_comparemodels.tl（非区分・比例・分離モデル間の比較を行うTreefinder Languageスクリプト）
│  ├ partition_criterion_xxx_singlesearch.tl（単一の樹形探索を行うTreefinder Languageスクリプト）
│  ├ partition_criterion_xxx_shotgunsearch.tl（ショットガン樹形探索を行うTreefinder Languageスクリプト）
│  ├ partition_criterion_xxx_bootstrap.tl（ブートストラップ解析を行うTreefinder Languageスクリプト）
│  └ ...
├ Scores
│  ├ partition_model.txt（各領域における各モデルの最大化対数尤度）
│  └ ...
└ Logs（その他のログファイルの出力されるフォルダ）
   └ ...
```

partition はパーティション名 (入力ファイル名)、criterion はモデル選択規準、xxx は非区分・比例・分離モデルの適用状況を示しています。全領域連結配列は whole という名前のパーティションとなっています。非 Windows 環境ではバッチファイルの代わりにシェルスクリプトが作成されます。

χ^2 検定の結果 (chisq_partition.txt) の内容は、pgtestcomposition の出力と同じ形式です。p 値が 0.05 以下のとき、OTU 間の塩基・アミノ酸組成に有意な差があると考えられます。ただし、この p 値が信頼できるデータには条件があり、それを満たしていない場合は末尾にその旨を示すメッセージが出ています。もしも塩基・アミノ酸組成に有意な差があったのであれば、データ改変による均一化を検討して下さい。他にも、系統樹上で組成が変化することを許容する不均質モデル (Blanquart and Lartillot, 2006, 2008) の適用を検討するのも良いですが、このモデルを適用できる nhPhyloBayes はかなり解析が遅いので、大規模データに適用するのは難しいと思います。

次に、各領域・コドン位置のモデル選択結果 (partition_criterion.txt) をテキストエディタで開いてみると以下のような内容となっています。

ファイルの内容 3.1　モデル選択の結果

```
1 model                          criterion  weight                -LnL  nparam
2 SYM_GeneCodonPos1Gamma   5.237279083000e+004 0.98496 2.606139541500e+004    125
3 J2ef_GeneCodonPos1Gamma  5.238115467800e+004 0.01504 2.606757733900e+004    123
4 SYM_Gamma                5.288409574800e+004 0.00000 2.631904787400e+004    123
5 以下略
6 モデル名                       criterionの値       weight  -LnLの値             パラメータ数
```

GeneCodonPos1Gamma というのは、領域間・コドン位置間に異なる速度を当てはめた上で、領域・コドン位置に共通の Γ 分布モデルを当てはめたものです。AICc や BIC に基づいたモデル選択の結果では、上記の内容に加えてサンプルサイズの値が記述されています。AICc と BIC の計算に用いるサンプルサイズの値は複数考えられるため、それぞれをサンプルサイズに用いてモデル選択を行った結果が出力されています。各出力ファイルで使われているサンプルサイズは以下のようになっています。

AICc1・BIC1: 系統樹上での最小塩基置換数 (最節約樹長)

AICc2・BIC2: 各座位における最小塩基置換数の合計

AICc3・BIC3: 各座位における形質状態の合計

AICc4・BIC4: 座位数 (配列長)

AICc5・BIC5: 変異のある座位数

AICc6・BIC6: 座位数 × 配列数

最もよく使われているサンプルサイズは AICc4・BIC4 の座位数です。

ここで重要なのは、**このファイルで最上位になっているモデルが実際の解析で適用されるとは限らない**ということです。というのも、ここでは比較に用いた候補モデル全ての順位が示されているのであって、たとえ最上位でも解析ソフトの側が対応していなければ適用できないからです。実際に適用されるモデルは、必ず解析ソフトで用いる設定ファイルを直接開いて確認して下さい。

Results フォルダに作成される whole_criterion_comparemix.txt は、連結配列における非区分・比例・分離モデル間の比較結果です。内容は以下のようなものです。

ファイルの内容 3.2 非区分・比例・分離モデルからの選択結果

```
1 model                                    AIC                -LnL  nparam
2 Separate_CodonProportional       1.286036307191e+004 6.373181535953e+003     57
3 Proportional_CodonProportional   1.286895735412e+004 6.385478677060e+003     49
4 Separate_CodonSeparate           1.288258125450e+004 6.352290627248e+003     89
5 Proportional_CodonNonpartitioned 1.401815088065e+004 6.983075440327e+003     26
6 Separate_CodonNonpartitioned     1.402149556766e+004 6.976747783830e+003     34
7 Nonpartitioned                   1.413466486467e+004 7.049332432334e+003     18
8 モデル名                         criterionの値       -LnLの値             パラメータ数
```

このファイル内のモデルはそれぞれ以下のようなものです。

- 領域間分離・コドン位置間比例モデル
- 領域間比例・コドン位置間比例モデル
- 領域間分離・コドン位置間分離モデル
- 領域間比例・コドン位置間非区分モデル
- 領域間分離・コドン位置間非区分モデル
- 非区分モデル

なお、Kakusan4・Aminosan は MrBayes (MrBayes5D) と Treefinder 用の比例・分離モデルを適用する設定ファイルを書き出すことができますが、Kakusan4・Aminosan が複数領域データにおいて実際に行っているのは、既に述べたように「それぞれの領域」での最適モデルの選択と、それぞれの領域で選択されたモデルを用いた非区分・比例・分離モデル間の比較だけです。これは、領域ごとに当てはめるモデルが多数あるとき、その組み合わせはさらに多数になってしまい、全ての比較を現実的な時間で処理することが不可能だからです。ただし、分離モデルはただそれぞれの領域で最大化した対数尤度を足し合わせたものですので、モデル選択に AIC を用いる場合は全ての組み合わせで正攻法で尤度を計算してモデル選択した結果と完全に一致します。AICc や BIC は相加的ではないため完全には一致しない可能性があります。

Kakusan4・Aminosan では、このようにして選択された分離モデルに対して全ての領域・コドン位置で枝長が比例するように制約を課すことで比例モデルの設定ファイルを作成しています。当然、実際には領域・コドン位置間で枝ごとの置換速度のパターンが異なる場合には、部分的に分離モデルを適用し部分的に比例モデルを当てはめたモデルがより良い可能性はありますが、そのような比較は行っていませんし設定ファイルも作成されません。また、非区分・比例・分離モデル間の比較を Treefinder で行っている場合、Treefinder が対応していないモデルは比較対象に入っていません。比較対象に入っていないモデルがベストである可能性は常に残っていることに注意して下さい。

3.3　非区分・比例・分離モデル間の比較

Kakusan4･Aminosan は出力フォルダ下の Treefinder フォルダに partition_criterion_comparemodels.tl というファイルを書き出します。このファイルの解析を Treefinder で実行してできるログファイルを見れば、どのモデルが最適なのか分かります。ですから、Kakusan4・Aminosan で非区分・比例・分離モデル間の比較を行っていなくても、後から実行することができます。また、Kakusan4 は連結配列での非区分・比例・分離モデル間の比較しか行いませんが、このファイルを使えばタンパクコード領域塩基配列でコドン位置間比例・分離とコドン位置間非区分モデル間の比較も行うことができます。ただ、ログファイルの見方が難しいので、ここで説明しておきます。

partition_criterion_comparemodels.tl を実行してできるのは､partition_criterion_comparemodels.log というファイルです。これをテキストエディタで開いて中身を見ます。以下のような内容になっているはずです。

ファイルの内容 3.3　partition_criterion_comparemodels.log の内容

```
1 {
2  {Likelihood->最大化対数尤度,Phylogeny->系統樹,SubstitutionModel->最適化した分子進化モ
       デル,OSubstitutionModel->最適化されていない分子進化モデル(与えたモデルの設定),
       OEdgeOptimizationOff->枝長最適化設定,NSites->座位数(配列長),NParameters->パラメー
       タ数,AIC->AICの値,AICc->AICcの値,HQ->HQの値,BIC->BICの値,Checksum->データのチェッ
       クサム,PartitionKeys->各パーティションのID,PartitionRates->最適化した枝長モデル,
       OPartitionRates->最適化されていない枝長モデル,NSitesPartitionwise->各パーティショ
```

```
      ンの座位数,FilterNames->パーティションを示している項目名,LikelihoodTime->計算に要
      した時間,LikelihoodMemory->計算に要したメモリ量},
以下略
 ()
}
```

これは、各モデルを適用して尤度最大化した結果です。ファイル名に含まれる情報量規準の値に応じて並び替えられており、先頭に書かれているものがベストなモデルを当てはめたものです。なお、Treefinder における AICc・BIC は、Kakusan4・Aminosan における AICc4・BIC4 に当たります。

この中で、OPartitionRates->と PartitionKeys->の項目を見ます。そもそもこれらの項目が存在しないなら、非区分モデルが使われています。OPartitionRates->Optimum となっていれば比例モデルが使われています。OPartitionRates->{1:1.,2:1.,3:1.,4:1., **以下略**} となっていれば分離モデルです。

PartitionKeys->の項目数は区分しているパーティション数を示しますので、例えば 10 領域データで 10 項目あれば領域間比例または領域間分離でコドン位置間は共通のモデルが当てはめられているか、またはそもそもタンパクコード領域がデータに含まれていないということになります。10 領域のうち 1 領域がタンパクコードデータであるとき、PartitionKeys->の項目数が 12 であれば、領域間だけでなくコドン位置間も比例または分離モデルが当てはめられていることになります。タンパクコードの 1 領域のみのデータで、3 領域に分けられているなら、コドン位置間比例またはコドン位置間分離モデルが使われています。

OPartitionRates->{1:1.,2:Optimum,2:Optimum,2:Optimum,3:1., **以下略**} のように、**数**:Optimum と **数**:1. が混在しているなら、領域間分離・コドン位置間比例モデルを当てはめたモデルです。

以上のようにして、領域間比例・分離やコドン位置間比例・分離のモデルを比較して最適なものを選択することができます。

第 4 章

最尤系統推定

4.1 最尤系統推定とは何か

今更ではありますが、そもそも尤度とは「あるモデルが正しいと仮定した状況で手元のデータが得られる確率」のことです。これは、データに対するモデルの当てはまりの良さを表します。ここで、10 回のコイントスを行って表が 1 回、裏が 9 回出たときの状況を考えましょう。すると、「このコインを使ったコイントスでは表と裏が 1:9 の比率で出る」というモデルの尤度 L_1 は以下のようになります。

$$\begin{aligned} L_1 &= \frac{1}{10} \times \left(\frac{9}{10}\right)^9 \\ &= 0.0387 \end{aligned} \tag{4.1}$$

「このコインを使ったコイントスでは表と裏が等確率で出る」というモデルの尤度 L_0 は以下のようになります。

$$\begin{aligned} L_0 &= \left(\frac{1}{2}\right)^{10} \\ &= 0.000977 \end{aligned} \tag{4.2}$$

このように、$L_1 > L_0$ であることから、前者のモデルの方が当てはまりが良いことになります。ただし、前者は「表と裏が 1:9 の比率」ということをデータから推定していると考えられますので、パラメータが 1 つありますが、後者にはデータから推定しているパラメータがありませんので、AIC は以下のようになります。

$$\begin{aligned} AIC_1 &= -2 \times \left\{ \ln\left(\frac{1}{10}\right) + \ln\left(\frac{9}{10}\right) \times 9 \right\} + 2 \times 1 \\ &= 8.50 \end{aligned} \tag{4.3}$$

$$\begin{aligned} AIC_0 &= -2 \times \left\{ \ln\left(\frac{1}{2}\right) \times 10 \right\} + 2 \times 0 \\ &= 13.86 \end{aligned} \tag{4.4}$$

ここでも $AIC_1 < AIC_0$ であることから、やはり前者のモデルの方が良いということになります。

最尤系統推定は、分子進化モデルは固定 (パラメータ値はそうでない) にして、最も尤度が高くなるような系統モデル=系統樹を選択するというものです。系統モデルは枝長パラメータと樹形から成りますが、検討すべき樹形数は配列数=系統樹の端点数=OTU (operational taxonomic unit) 数に応じて劇的に膨れ上がってしまいます。そのため最尤系統推定では、網羅的探索 (exhaustive search) は計算時間から見て非現実的です。そこで、ほとんどの場合は発見的探索 (heuristic search) を行います。これは、近隣結合法 (neighbor-joining [Saitou and Nei, 1987]) や段階的配列付加法 (stepwise/sequential sequence addition [Swofford and Begle, 1993]) などで生成した初期系統樹 (initial/starting tree) と、それを枝交換 (branch swapping) によって樹形改変 (topology rearrangement) してできる系統樹の尤度を計算し、より尤度の高い系統樹が見つかればそれを初期系統樹としてまた同じことを繰り返す、というものです。

4.2　Treefinder による発見的探索

最尤系統推定に用いられるソフトは各種ありますが、ここでは比較的高速で多機能な Treefinder (Jobb *et al.*, 2004) を用いて説明していきます。他にも大規模データでは RAxML (Stamatakis, 2006) が非常に高速で使いやすいと言われています。樹形探索範囲も Treefinder より RAxML の方がずっと広いと思われます。ある程度以上の規模 (100 OTU 以上または塩基 10k bp 以上アミノ酸 5k aa 以上) になると RAxML でしか解析できないものもあります。ただし、RAxML は塩基配列データでは置換速度行列は GTR モデルしか使えない上、分離モデルは使えるものの比例モデルはサポートしていません。基本的に最も複雑なモデルにしておけばいいだろうという方針のようです。膨大なパラメータ数に見合うだけの膨大なデータがある場合はこれでも問題は無いのかもしれません。Treefinder には解析不能な大規模データでは RAxML を検討されるとよいでしょう。また、分離モデルが選択された場合、Treefinder でも適用できるものの解析に非常に時間がかかります。そのような場合は RAxML を使った方が良いでしょう。RAxML の使い方は ver.7.0.4 のマニュアルに詳しく書かれていますが内容が古いので、最新版で`-h` オプションを付けて実行したときに表示されるメッセージを参照するようにして下さい。

Kakusan4・Aminosan で分子進化モデルの選択を行った場合、出力フォルダ下の `Treefinder` フォルダに
`partition_criterion_xxx_singlesearch.tl`
というファイルが作成されているはずです。`partition` はパーティション名、`criterion` はモデル選択規準、`xxx` は mixed model の適用状況を示しています。連結配列データは `whole` という名前のパーティションとなっています。作成されるファイルは入力されたデータが複数領域データか、タンパクコード領域データかによって異なりますが、例えば、
`whole_AIC_proportional_codonproportional_singlesearch.tl`
(AIC をモデル選択規準として領域・コドン位置ごとに選ばれたモデルを比例モデルとして連結配列に当てはめて樹形探索を行う Treefinder Language スクリプト)
とか
`whole_AIC_codonproportional_singlesearch.tl`

(AIC をモデル選択規準としてコドン位置ごとに選ばれたモデルを比例モデルとして連結配列に当てはめて樹形探索を行う Treefinder Language スクリプト)
とか
`whole_AIC_nonpartitioned_singlesearch.tl`
(AIC をモデル選択規準として選ばれた非区分モデルを配列全体に当てはめて樹形探索を行う Treefinder Language スクリプト)
といったファイルが出力されているはずです。タンパクコード領域配列では
`whole_AIC_nonpartitioned_singlesearch.tl`
は
`whole_AIC_codonnonpartitioned_singlesearch.tl`
という名前で出力されています。ただし、タンパクコード領域において全コドン位置に共通なモデルの検討を行わない設定で Kakusan4 によるモデル選択を行った場合はこのファイルは出力されません。

上述のことから分かるように、同じパーティションのデータに対して、多くのモデルの当てはめ方があり得ます。上記の例では比例モデルしか挙げていませんが、分離モデルも当然利用可能です。どの当てはめ方が最も適しているかはデータによって異なります。どこモデルを当てはめればよいかは p55 の第 3.2.2 節をご覧下さい。

以上のファイルの解析を Treefinder の `Kernel` メニューにある `Load TL Script ...` で表示されるダイアログで指定することで、各ファイル内に記述されたコマンドが実行されます。コマンドプロンプトやターミナルでこれらのファイルのあるフォルダに移動して、以下のようにコマンドを実行することでも同様の処理が可能です (`*.tl` は適当なファイルに読み替えて下さい)。

```
tf *.tl
```

解析が終わると、分子進化モデルパラメータの最尤推定値を記録したファイル (`_optimum.model` および `_optimum.rates` のこと) と最尤系統樹 (`_optimum.nwk`)、そしてそれらを含む全ての情報が記述された `.log` ファイル (TL Report 形式) が出力されます。

`_optimum.nwk` や `.log` ファイルを `File` メニューの `Open Image ...` から開くことで内容を Treefinder に描画させることができます。系統樹が表示されている状態では、`View` メニューの `Redraw ...` で表示されるダイアログにて根の付く場所の変更や枝交換、特定の枝の削除も行えます。この表示内容は `File` メニューの `Save` から PostScript 形式の画像ファイルとして保存することができます。対応する画像編集ソフトで開けば、好きなように加工することができます。

なお、ここでの樹形探索は 1 本の初期系統樹を近隣結合法で作成して行っています。Treefinder は複雑な樹形改変を行わないためあまり広範囲の探索を行っているとは言えません。この樹形探索を実行後、
`partition_criterion_xxx_shotgunsearch.tl`

を用いることで、それまでに得られた最尤系統樹から無作為に (OTU 数 – 3) / 2 回の最近隣交換 (nearest neighbor interchange) を行って OTU 数 – 3 本の系統樹を生成し、それらを初期系統樹としたショットガン的樹形探索を行うことができます。この探索を繰り返すことで系統樹を改善することが可能ですが、次節で説明する likelihood ratchet の方が尤度の島を発見しやすいはずなのでおすすめです。

4.3　Treefinder・Phylogears2 による並列 likelihood ratchet

Ratchet 法 (Nixon, 1999; Vos, 2003) は、無作為に重み付けしたデータを用いて樹形探索して得た系統樹を初期系統樹として用いることで、尤度の島を発見しやすくするというものです。とだけ説明しても分かりにくいのでもう少し詳しく解説しましょう。

まず、同じ高さの尤度の島がいくつもある状況を想像して下さい。無作為にデータを重み付けすると、特定の尤度の島だけが相対的に高くなります (実際にはデータの「加重」では島が低くなることはあっても高くなることはありません。それぞれの島の「沈降」がばらつくことで、特定の島が「相対的に」高くなるということです)。そのようなデータを用いて樹形探索を行うと、高くなった島の頂上が推定結果として得られるでしょう。その推定結果を初期系統樹として元の重み付けしていないデータで樹形探索を行えば、頂上の位置が多少ずれるかもしれませんが、同じ島の頂上を推定結果として得ることになるでしょう。これを何度も繰り返してやれば、全ての尤度の島の頂上を見つけられる可能性は、通常の樹形探索よりも高くなるはずです。

つまり、以下のような解析を繰り返すのが ratchet 法です。

1. データを無作為に重複を許して加重する
2. 重み付けデータで樹形探索を行う
3. その結果を初期系統樹として元のデータで樹形探索を行う

実際には全く同じ高さの尤度の島がたくさんある状況はまれですが、非常に近い高さの尤度の島がたくさんある状況は大規模データではよくあるでしょう。また、樹形空間内で最尤系統樹が初期系統樹 (近隣結合系統樹がよく用いられる) から非常に遠く離れており、しかもその間には尤度の島や谷、さらには「海溝」があったりすることもあるかもしれません。そのような状況では、問題を完全に解決できるとは言い切れませんが、ratchet 法の適用によって問題を軽減できるでしょう。重み付けデータにおける樹形探索は大変なので、最節約規準 (parsimony criterion) における無作為配列付加 (random sequence addition [Swofford and Begle, 1993])、または近隣結合法による系統樹の生成のみに留め、繰り返し数を増やすという方法もあります。非常に大規模なデータではそういう選択をせざるを得ないでしょう。likelihood ratchet を提案した Vos (2003) では近隣結合法を用いています。最節約規準に基づく無作為配列付加による系統樹の生成は PAUP*・POY4・TNT のいずれかによって行うことができます。

では、実際に ratchet 法による最尤系統推定を行ってみましょう。まずはコマンドプロンプトやターミナルを起動して Kakusan4・Aminosan の出力フォルダ内にある Treefinder フォルダに移動します。その上で、以下のように Phylogears2 に含まれる pgtfratchet コマンドを実行します。

```
pgtfratchet
```

すると、以下のような表示が出ます。

```
pgtfratchet 2.0.2010.11.07
============================================================================

Official web site of this script is
http://www.fifthdimension.jp/products/phylogears/ .
To know script details, see above URL.

Copyright (C) 2008-2009  Akifumi S. Tanabe

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Model and TF files were found. Entering interactive mode...

Which do you want to analyze? (name/number)
  1: CYTBnuc_P
  2: ND5nuc_P
  3: whole
```

これは、フォルダ内にある配列データを見つけたので、どれを使った解析を行うかをユーザーに質問しています。解析に使いたいデータを選んで番号か名前を入力し、Enter キーを押して下さい。すると、以下のように適用できるモデルの選択肢が表示され、どれにするのかを質問されます。当然ですが、適用可能なモデルしか表示されませんので選択肢はもっと少ないかもしれません。

```
Which model do you want to apply to the data? (name/number)
  1: proportional_codonproportional
  2: separate_codonproportional
  3: separate_codonseparate
  4: proportional_codonnonpartitioned
  5: separate_codonnonpartitioned
  6: nonpartitioned
```

適用したいモデルを選択して番号か名前を入力し、Enter キーを押して下さい。すると、次に以下の質問が出てきます。

```
Which criterion do you want to use? (name/number)
  1: AIC
  2: AICc1
  3: AICc2
  4: AICc3
  5: AICc4
  6: AICc5
  7: AICc6
  8: BIC1
  9: BIC2
  10: BIC3
  11: BIC4
  12: BIC5
  13: BIC6
```

これはどのモデル選択規準で選択されたモデルを各領域に適用するかという質問です。規準を選択して番号か名前を入力し、Enter キーを押して下さい。すると、以下の質問が表示されます。

```
Which do you want to use the program for generation of starting trees? (paup/poy
/tf/tnt)
(default: poy)
```

ここでは PAUP*・POY4・TNT のいずれかを用いた無作為配列付加によって初期系統樹を生成するか、Treefinder による浅い樹形探索によって初期系統樹を生成するかを選択します。大きなデータでは無作為配列付加を用いる方が良いでしょう。速度は TNT ≥ PAUP* >> POY4 >> Treefinder です。TNT は OTU 名が 31 文字までという制限がありますので注意して下さい。次は無作為重み付け量に関する質問です。

```
How many percentages of sites do you want to upweight? (integer)
(default: 25)
```

ここでは、配列長の何 % 分の重み付けを行うかを入力して下さい。1,000bp のデータで 25% とすれば 250 回の重複を許した加重がなされます。一般に 20~25% が良いとされています。次は反復数に関する質問です。

```
How many replicates do you want to run? (integer)
(default: 100)
```

これは、生成する初期樹形の数に当たります。ただし、もしも同一の樹形が複数生成された場合は pgtfratchet は重複を除去するため、実際に行われる樹形探索の反復数は減少します。つまりこれは正確には反復の最大値ということです。もしも反復数が足りなくても、同じ解析を走らせたときに前回の解析ログファイルが残っていると、pgtfratchet は前回のファイルを置き換えるか、それとも追加するかを訊いてきます。ここで追加すると答えれば、前回の解析を無駄にせずに反復数を増加させることができます。次は樹形制約に関する質問です。

```
If you want to give topological constraint, specify an input file name.
Otherwise, just press enter.
```

ここで制約を記したファイル名を入力することで制約付き樹形探索を行うことができます。樹形制約ファイルの作成法は p91 の第 7.1 節を参照して下さい。制約が必要無ければ空欄のまま Enter キーを押して下さい。最後に同時に走らせるプロセス数に関する質問です。ここで 2 以上の値を指定することで 2 つ以上の CPU(コア) を利用することが可能です。マシンが搭載する CPU(コア) 数と同じ値にすることで最も高速に解析が行われます。

```
How many processes do you want to run simultaneously? (integer)
(default: 1)
```

この質問が終わると以下のように表示が出ます。

```
All configurations have been completed.
Just press enter to run!
```

心の準備ができたら Enter キーを押して解析を始めて下さい。

この pgtfratchet は、以下のような処理を行っています。

1. 重み付けデータの作成
2. 重み付けデータにおける無作為配列付加 or 樹形探索による初期系統樹群の作成
3. 初期系統樹群からの重複樹形の除去
4. 初期系統樹群を初期系統樹とする樹形探索の並列実行
5. 実行ログファイルからの最尤系統樹とパラメータ値の取り出し
6. 探索密度評価に用いる指標の計算

解析が終わると、分子進化モデルパラメータの最尤推定値を記録したファイル (_optimum.model および_optimum.rates のこと) と最尤系統樹 (_optimum.nwk)、そしてそれらを含む全ての情報が記述された.log ファイル (TL Report 形式) が出力されます。なお、樹形制約を課した場合はそれぞれのファイル名末尾に制約系統樹ファイルのファイル名が付いた名前で出力されます。

4.3.1 Likelihood ratchet の探索密度の評価

Likelihood ratchet における「その初期系統樹生成方法での」樹形探索の密度は、各反復の最尤系統樹を尤度で並び替えたとき、1 位の樹形と同一の樹形が上位何位まで占めているかを数えることで計ることができます。これは、100 反復で全て同じ樹形を得てしまうなら、もう同じ初期系統樹生成法では何度やっても同じ結果を得る可能性が高いし、100 反復のいずれも異なる樹形を支持し、1 位の樹形は単独 1 位であるならば、もっとやればもっと良い樹形が見つかるかもしれないという考え方に基づくものです。従って、この方法で評価できるのはあくまで探索範囲内の探索密度であり、探索範囲が十分かどうかは分からないことに注意して下さい。

この探索密度評価は、pgtfratchet 終了時に_checkcoverage.txt として出力されています。もしも初期系統樹がそもそも 1 樹形分しか作成されなかった場合にはこのファイルは出力されていません。また、指標を計算できないため pgtfratchet がエラーを吐いて終了しますが気になさらなくて結構です。さて、このファイルをテキストエディタなどで開くと以下のようになっています。

ファイルの内容 4.1　_checkcoverage.txt の内容

```
# 0: same topology
# 1: different topology

source  input   same or not
1       2       0
1       3       0
1       4       0
1       5       0
1       6       1
1       7       1
以下略
```

ここで、source は比較元の樹形番号、input は比較対象樹形の番号、same or not は樹形が同じ (0) か異なる (1) かを示しています。1 位と同じ樹形が多ければ多いほど密度は高いことになります。どの程度あれば十分かは分かりませんが、筆者はとりあえずできるだけ 20 位以降まで同一樹形になるようにしています。それが無理な場合も上位何本が同一だったかを書いておけばよいでしょう。初期系統樹に重複が含まれていて最大反復数より少ない回数の樹形探索しか行われなかった場合は、もっとずっと少なくても密度は非常に高いと見なすことができるでしょう。また、そもそも OTU が多いのにデータ量が少ないなど、ブートストラップ解析をしたときに多数決合意樹に支持率の低い枝が沢山現れてしまうようなデータでは、どれだけ回数を増やしても 1 位と同じ樹形が全く得られないこともあります。そのようなデータでは適当なところで切り上げるしかありません。

4.4　Treefinder によるブートストラップ解析

樹形の信頼性 (credibility) を検討するために、ブートストラップリサンプリング (bootstrap resampling) したデータを用いて系統推定を繰り返すことで、各内分枝 (internal/interior branch) の再現率を得ます。これが系統推定におけるブートストラップ解析です (Felsenstein, 1985)。

Kakusan4 でモデル選択を行った場合は、出力フォルダにある
partition_criterion_xxx_bootstrap.tl
を Treefinder の Kernel メニューにある Load TL Script ... から指定して実行することで解析を行うことができます。このファイルでは反復数は 100 に設定されていますので、変更したい場合はファイルをテキストエディタで開いて編集して下さい。また、このファイルでは分子進化モデルのパラメータを元データでの系統推定で得られた値に、初期系統樹を最尤系統樹に固定して解析を行うため、事前に固定パラメータのモデル設定ファイル (元データでの系統推定で作成される_optimum.model および_optimum.rates のこと) と最尤系統樹 (同じく_optimum.nwk) を書き出しておく必要があります。前節までの内容の通りに最尤系統推定を行っていれば、こ

れらのファイルは既に作成されているはずです。

解析が終わると、_bootstrap.log というログファイル、_bootstrap.nwk という各反復の樹形探索結果が書かれた Newick 形式系統樹ファイル、_consensus.nwk という多数決合意樹ファイルが出力されています。適当なソフトで開いて内容を確認して下さい。

4.4.1 Treefinder と Phylogears2 による並列ブートストラップ解析

コマンドプロンプトやターミナルを起動して Kakusan4・Aminosan の出力フォルダ内にある Treefinder フォルダに移動します。その上で、以下のように Phylogears2 に含まれる pgtfboot コマンドを実行します。

```
pgtfboot
```

すると、以下のようなメッセージが表示されます。

```
pgtfboot 2.0.2010.11.07
========================================================================

Official web site of this script is
http://www.fifthdimension.jp/products/phylogears/ .
To know script details, see above URL.

Copyright (C) 2008-2009  Akifumi S. Tanabe

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Model and TF files were found. Entering interactive mode...

Which do you want to analyze? (name/number)
  1: CYTBnuc_P
  2: ND5nuc_P
  3: whole
```

pgtfratchet と同様に、まずはどのデータセットを解析するのかを選んで答えます。数字でも名前でもどちらでも構いません。すると、次に適用するモデルに関する質問が表示されます。

```
Which model do you want to apply to the data? (name/number)
  1: proportional_codonproportional
  2: separate_codonproportional
  3: separate_codonseparate
```

```
  4: proportional_codonnonpartitioned
  5: separate_codonnonpartitioned
  6: nonpartitioned
```

ここではデータセットに適用するモデルを選択します。次に、各領域に当てはめるモデルを選択する規準に関する質問が出ます。

```
Which criterion do you want to use? (name/number)
  1: AIC
  2: AICc1
  3: AICc2
  4: AICc3
  5: AICc4
  6: AICc5
  7: AICc6
  8: BIC1
  9: BIC2
  10: BIC3
  11: BIC4
  12: BIC5
  13: BIC6
```

この質問に答えると、以下のように反復数に関する質問がなされます。

```
How many replicates do you want to run? (integer)
(default: 100)
```

pgtfratchet と同様に、後から解析を足すことができますので少なめで行っておくと良いでしょう。次は樹形制約に関する質問です。

```
If you want to give topological constraint, specify an input file name.
Otherwise, just press enter.
```

制約付きブートストラップ解析がしたければここで制約を記したファイルを指定します。樹形制約ファイルの作成法は p91 の第 7.1 節を参照して下さい。必要無ければ空欄のまま Enter キーを押して下さい。次は各反復の樹形探索で用いる初期系統樹に関する質問です。

```
Which do you want to use as starting tree? (RAWML/NJ/FILENAME)
(default: RAWML)
```

標準の RAWML では、元データにおける樹形探索で得られた最尤系統樹 (_optimum.nwk ファイルに書かれている) を用います。元データでの樹形探索を行っていなければファイルが作成されていませんのでエラーになります。NJ は各反復で近隣結合法により初期系統樹を作成します。樹形ファイルを指定することで、初期系統樹を任意のものにすることもできます。次の質問は、分子進化モデルのパラメータ値に関するものです。

```
Which value do you want to use to the parameters? (RAWML/OPTIMIZE/FILENAME)
(default: RAWML)
```

RAWML は先程と同様に元データにおける樹形探索で得られた最尤系統樹での値 (_optimum.model および_optimum.rates ファイルに書かれている) に固定します。OPTIMIZE は各反復でデータから推定するものです。また、ファイルを指定することで任意の値を適用することもできます。最後に最大並列数に関する質問です。

```
How many processes do you want to run simultaneously? (integer)
(default: 1)
```

ここで 2 以上の値を指定することで 2 つ以上の CPU(コア) を利用することが可能です。マシンが搭載する CPU(コア) 数と同じ値にすることで最も高速に解析が行われます。この質問が終わると以下のように表示が出ます。

```
All configurations have been completed.
Just press enter to run!
```

心の準備ができたら Enter キーを押して解析を始めて下さい。

解析が終わると、_bootstrap.log というログファイル、_bootstrap.nwk という各反復の樹形探索結果が書かれた Newick 形式系統樹ファイル、_consensus.nwk という多数決合意樹ファイル、_optimum_with_supportvalues.nwk という元データの最尤系統樹に支持率をマッピングした Newick 形式系統樹ファイル、_allhypotheses.nwk という出現した全ての系統仮説とその出現頻度が記録された Newick 形式系統樹ファイルが出力されています。FigTree やテキストエディタなどの適当なソフトで開いて内容を確認して下さい。

第 5 章

ベイジアン系統推定

マルコフ連鎖モンテカルロ法 (Markov chain Monte Carlo 略して MCMC) を用いたベイジアン系統推定 (Bayesian phylogenetic inference) は、近年普及してきていますが、まだパラメータ設定や収束 (convergence) 判定には一定の知識が必要です。ここでは MCMC について簡単に説明した上で、MrBayes (Ronquist and Huelsenbeck, 2003) の改造版である MrBayes5D による系統推定と、Tracer を用いた収束判定について説明します。

5.1 メトロポリス・ヘイスティングス法

MrBayes (MrBayes5D) が行う MCMC は、メトロポリス・ヘイスティングス法 (Metropolis-Hastings algorithm [Metropolis *et al*., 1953; Hastings, 1970]) と呼ばれる MCMC です。この方法は、以下のような手順でパラメータを最適化していくものです。

1. 全てのパラメータの初期値を適当に決定する
2. パラメータを適当に選択する
3. 選択されたパラメータを事前分布から導かれる提案分布に従って変更する、ことを提案する
4. パラメータ変更後の尤度を計算する
5. 尤度が変更前より良くなっていれば提案を 100% 受理し、良くなっていなくても一定のルールで受理する
6. 2 へ戻る
7. 以上の処理を継続しつつ、一定の間隔でモデルをサンプリングする

この処理がある程度進むと定常状態 (steady state) に入ります。定常状態に入る前のサンプルを捨て (burn-in)、残ったサンプルを事後分布 (posterior distribution) からのサンプルと見なして事後確率 (posterior probability) を得ます。

5.2　MrBayes5D による系統推定

MrBayes5D は現在最もよく利用されているベイジアン系統推定用ソフトウェア MrBayes を拡張し、より多くのアミノ酸置換モデルを使えるようにしたものです。塩基配列データの解析にはオリジナルとの違いはありません。計算も高速で多くのモデルに対応しており、MPI による並列化にも対応しています。以下の説明のほとんどはオリジナルの MrBayes にも適用できます。

Kakusan4・Aminosan で分子進化モデルの選択を行った場合、出力フォルダの `MrBayes` フォルダ内にある NEXUS ファイルを読み込むことで容易に選択されたモデルを適用した解析が可能です。`MrBayes` フォルダには
`partition_criterion_xxx.nex`
というファイルが作成されているはずです。`partition` はパーティション名、`criterion` はモデル選択規準、`xxx` は非区分・比例・分離モデルの適用状況を示しています。全領域連結配列は `whole` という名前のパーティションとなっています。作成されるファイルは入力されたデータが複数遺伝子座データか、タンパクコード領域データかによって異なりますが、例えば、
`whole_BIC4_proportional_codonproportional.nex`
(配列長 [座位数] をサンプルサイズとした BIC をモデル選択規準として領域・コドン位置ごとに選ばれたモデルを比例モデルとして連結配列に当てはめる設定を適用する NEXUS ファイル)
とか
`whole_BIC4_codonproportional.nex`
(配列長 [座位数] をサンプルサイズとした BIC をモデル選択規準としてコドン位置ごとに選ばれたモデルを比例モデルとして連結配列に当てはめる設定を適用する NEXUS ファイル)
とか
`whole_BIC4_nonpartitioned.nex`
(配列長 [座位数] をサンプルサイズとした BIC をモデル選択規準として選ばれたモデルを連結配列に当てはめる設定を適用する NEXUS ファイル)
といったファイルが出力されているはずです。タンパクコード領域配列では
`whole_BIC4_nonpartitioned.nex`
は
`whole_BIC4_codonnonpartitioned.nex`
という名前で出力されています。ただし、タンパクコード領域において全コドン位置に共通なモデルの検討を行わない設定で Kakusan4 によるモデル選択を行った場合はこのファイルは出力されません。

このように、同じパーティションのデータに対して、多くのモデルの当てはめ方があり得ます。どの当てはめ方が最も適しているかはデータによって異なります。Kakusan4・Aminosan による非区分・比例・分離モデルの

比較結果を参考にしてどのモデルを適用するか＝どのファイルを用いるかを適宜選択して下さい。ただし、尤度計算に用いている Treefinder が対応していないために検討していないモデルもあるので、この結果を過度に信用しない方が良いかもしれません。MrBayes5D は分離モデルにも対応しているため、分離モデルを適用する NEXUS ファイルも出力されていますが、MrBayes5D はあまり分離モデルを適用した解析を得意としていません (樹形探索範囲が狭くなる)。分離モデルが選択されてしまった場合には、RAxML による最尤系統推定を推奨します。

MrBayes5D で以上のファイルを用いて MCMC を実行するには、以下のようにコマンドを実行します。

```
mrbayes5d -i partition_criterion_xxx.nex
MrBayes > MCMC
```

これで MCMC は走り始めます (NGen オプションで指定しない限り 1,000,000 ステップ) が、どれだけ MCMC を走らせ続けるかが問題です。こちらに関しては p75 の第 5.3 節をご覧下さい。

5.3 Tracer による収束判定と有効サンプルサイズの推定

MCMC で難しいのは、「収束しているか」と「収束後のサンプル数は十分か」の判断です。これを補助してくれるのが Tracer です。MrBayes5D も ASDSF という収束判断の参考になる値を出力してくれますが、あまり当てにならないので気にしなくていいでしょう。ASDSF は標準設定では 1,000 ステップごとに計算されますが、この計算が案外重いので、MCMC コマンドのオプションに DiagnFreq=10000 (10,000 ステップごとに ASDSF を計算する) などと付けることでこの計算の頻度を変更してやると良いかもしれません。もしくは、MCMCDiagn=No をオプションとして与えることで最初から ASDSF の計算をしないようにしてもいいでしょう。NRuns=1 として同時に走らせる MCMC を 1 つに制限した場合も ASDSF は計算されません。

まず、MrBayes5D で MCMC を走らせて、以下のメッセージがでたところで、MrBayes5D はそのままにして Tracer を起動します。MCMC 実行時に NGen オプションで指定しない限り 1,000,000 ステップの時点で表示されるはずです。

```
MrBayes > MCMC
  中略
  Continue with analysis? (yes/no):
```

Tracer が起動したら、File メニューの Import Trace File... から MrBayes5D に読み込ませている NEXUS ファイルのあるフォルダにある NEXUS **ファイル名**.run1.p を指定して読み込ませます。同様に NEXUS **ファイル名**.run2.p も読み込ませます。現在のバージョンでは左側ペインへのファイルのドラッグアンドドロップによってファイルを読み込ませることも可能になっています。2 つのファイルの読み込みが終わったら、左上の Trace Files ペインで 2 つのファイル名を選択して反転表示状態にします。複数ファイルを選択するに

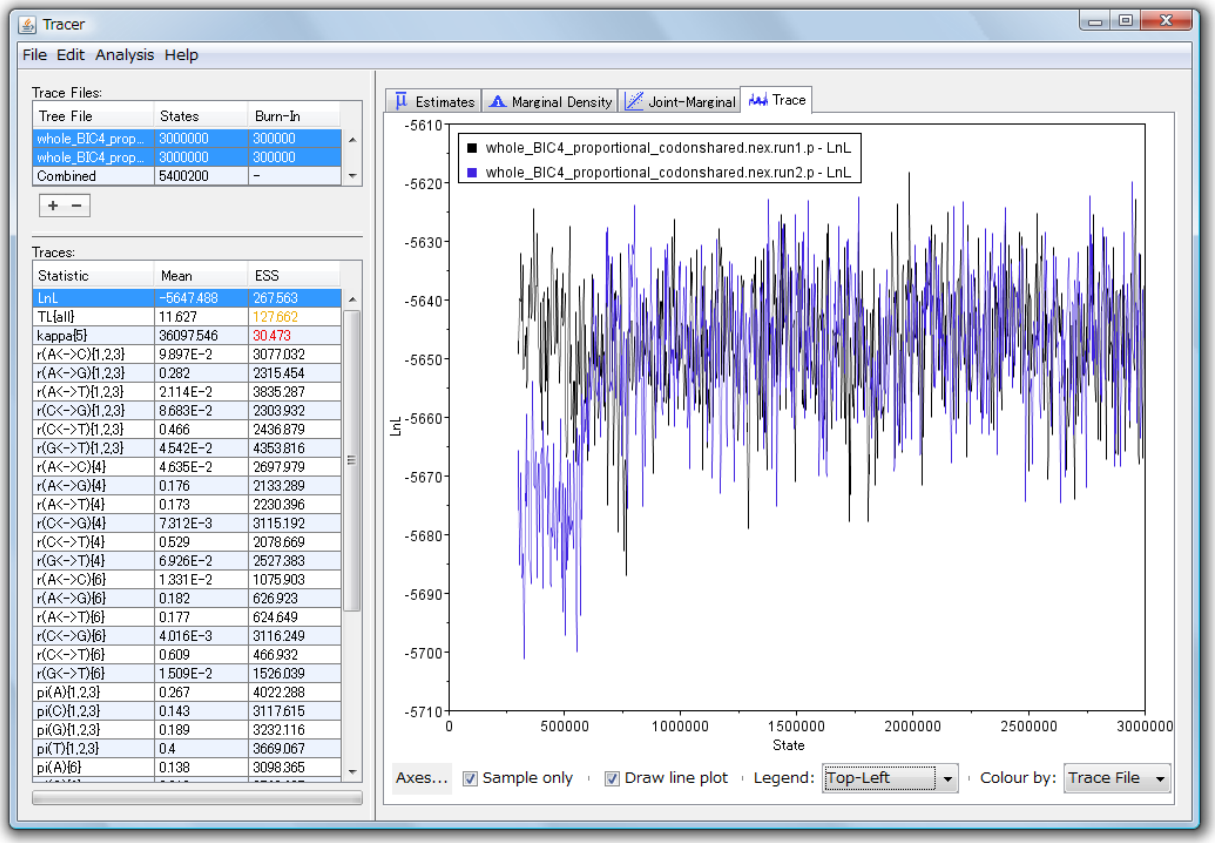
は Ctrl か Shift キーを押しながらファイル名を左クリックして下さい。

ここで読み込ませた 2 つのファイルは、MrBayes5D が同時に 2 つ (標準設定の場合) 走らせている MCMC のそれぞれのモデルのパラメータ値などが保存されているログファイルです。この 2 つの MCMC で、パラメータが定常状態に入っているか、近い値に収束しているかを Tracer で図示することで判断しようということです。

さて、この状態で、右側ペインのタブを Trace にして折れ線グラフを表示させます。そして、右下の Colour by を Trace File に、Legend を None 以外にして下さい。すると、2 つの MCMC の折れ線グラフが色分け表示されます (図 5.1)。左下の Traces ペインで反転表示させるパラメータを変更していくと、右ペインの折れ線グラフもそれに応じて変化していきます。このプロットを見て各パラメータが定常状態 (steady state) に入っているかを検討して下さい。もし定常状態に入っていそうもないようであれば、MCMC を継続して下さい。MCMC が中断したら、再度ファイルの読み込みをし直して定常状態に入るまで繰り返して下さい。

定常状態には入っても、2 つの MCMC が異なる局所最適解に収束してしまっている場合、両方の MCMC がより尤度の高い方へと収束するまで解析を続ける必要があります。しかし、あまりに尤度の谷が深いといつまで

図 5.1　Tracer による収束判定



右側のプロットでは対数尤度のステップごとの変化を表示しています。この例では 70 万ステップ付近で 2 つの MCMC がパラメータ空間内で同程度の尤度の場所に収束していると考えられます。その後の波形の乱れも一定していることから、定常状態に入っていると考えてよいでしょう。

経っても同じところへ収束しないことがあります。そのような場合、とりあえず尤度の高い方だけが最適解付近に収束していると見なしておき、サンプル数が十分量 (後述) の半分程度になるまで解析を続けます。その上で、各種出力ファイルの名前を変更してから何度も同じ解析を実行してやります。そして、2 つ以上の MCMC が同じ値に収束しているものの中で、最も尤度の高いものを本当に収束しているものとして結果に採用します。

パラメータが定常状態に入っていると確認できたら、収束後のサンプル数が十分かどうかを検討しましょう。まず、左上の `Trace Files` ペインの `Burn-In` (収束前で捨てるサンプル数) を適切な値に設定して下さい。この値は解析開始から定常状態に入るまでのステップ数で指定しますが、ログファイルには第 1 ステップの結果が入っているため、最初の 1,000,000 ステップを burn-in するには、この値を 1,000,100 に設定します。ただし、これは標準の 100 ステップに 1 回のサンプリング頻度に設定している (`SampleFreq=100`) 場合で、1,000 ステップに 1 回に設定している場合には 1,001,000 にします。つまり、burn-in したいステップ数に `SampleFreq` の値を加えた値にするということです。ここで、MrBayes5D の機能上の制約により、解析結果の要約 (summarize) するには全ファイルの `Burn-In` を等しくしておく必要があります。ただし、MrBayes5D の要約機能を利用せずに p80・第 5.4 節の方法で要約するのならばその必要はありません。

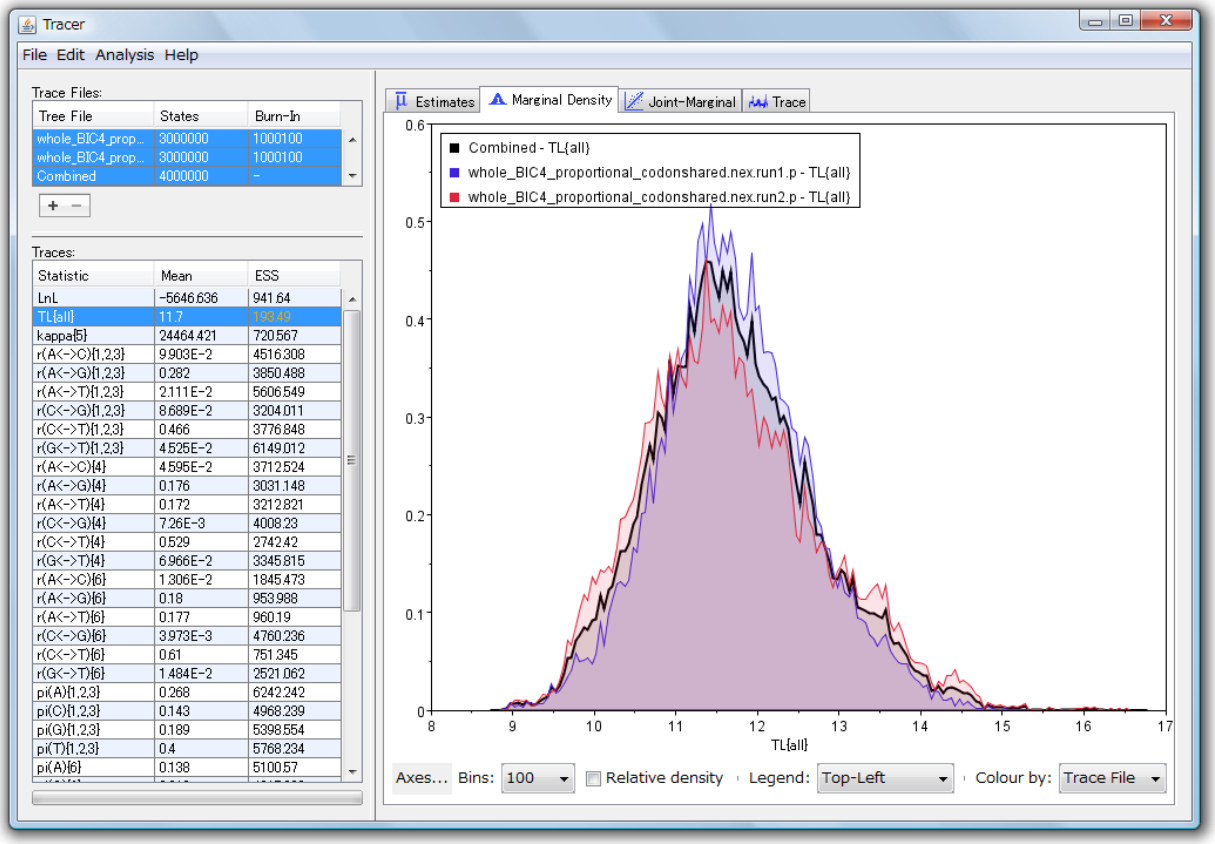
全ファイルの `Burn-In` を適切に設定できたら、左上 `Trace Files` ペインで `Combined` を含む全ての項目が反転表示された状態にして下さい。また、右側ペインのタブを `Marginal Density` に設定して事後確率密度を表示させます。そして、右下の `Colour by` を `Trace File` に、`Legend` を `None` 以外にして下さい。すると、各 MCMC と全 MCMC から得られたパラメータの事後確率密度が色分け表示されます (図 5.2)。左下の `Traces` ペインで反転表示させるパラメータを変更していくと、右ペインの密度曲線もそれに応じて変化していきます。このプロットを見て MCMC 間で同様の密度曲線となっていることを確認して下さい。さらに、左下 `Traces` ペインの ESS (effective sample size, 有効サンプルサイズ, [Kass *et al*., 1998]) の値を見ます。これが全て 100、できれば 200 を超えるようにして下さい。100 を下回るようなら、MCMC をさらに継続してサンプル数を増やす必要があります。

各パラメータの要約統計量を見るには、右側ペインのタブを `Estimates` にした状態で、左下 `Traces` ペインのパラメータをクリックして下さい。右側のペイン上部に表示されます。

5.3.1 収束しやすくする・有効サンプルサイズを大きくする方法

MCMC では、間隔を空けてモデルをサンプルすることで、各サンプルは独立したものとしてみなしています。しかし、独立性が低いと ESS の値は小さくなります。ESS は、実際のサンプル数ではなく、サンプル間の独立性を考慮した「実質的なサンプル数」を示しています。提案 (proposal) の受理率 (acceptance rate) が低かったり、状態交換 (state exchange) の成立が少ないと、サンプル間の独立性が下がり、ESS が小さくなります。そのような MCMC では、尤度の改善も進みにくいために収束に時間がかかるようになってしまいます。

図 5.2　Tracer による有効サンプルサイズの推定



右側のプロットは樹長の密度曲線を示しています。各 MCMC の密度曲線に極端なずれがないことを確認します。また、左下ペインの ESS が全て 100 以上になるまで MCMC を継続します。

これを解決するには、2 つのアプローチがあります。それは、サンプル間の独立性が低かろうがなんだろうがとにかく長く MCMC を続けて ESS を十分な数にするという方法と、サンプル間の独立性を上げてステップ数当たりの ESS を大きくする方法です。ESS の不足が少しだけであれば、解析を続けるだけで済む前者の方法を採るのが良いでしょう。しかし、絶望的なまでにステップ数当たりの ESS が小さくとても ESS が十分になるまではやっていられないということであれば、設定を変えて解析をやり直すしかありません。

受理率は、MCMC を停止したときに表示されますのでその値を見てどの提案の受理率が低いのかを確認します。以下のように表示されているはずです。

```
Acceptance rates for the moves in the "cold" chain:
  With prob. Chain accepted changes to
    1.23 % param. 1 (state frequencies) with Dirichlet proposal
    以下略
```

この値が低く、ESS がとても確保できないものをメモしておき、Props コマンドを使って設定を変更します。MCMC の最中でも、Tracer でパラメータ値の変遷を表示させることでパラメータの最適化の進行とかき乱れの良さを確認できます。横軸をステップ数、縦軸をパラメータ値とするプロットにおいて、上下に激しく乱れてお

らず矩形波になっているものがあれば、そのパラメータの最適化が進んでいないか、かき乱れが良くないと考えられます。受理率が高くても、提案頻度が低すぎて矩形波になることもあります。Tracer による確認方法ではそれを発見可能ですので、こちらの方法の方がおすすめです。Props コマンドは、以下のように用います。

```
MrBayes > Props
  中略
  Select a parameter to change (1 - 36; 0 to exit; 37 to zero all proposal rates): 26
  (変更するパラメータを選択)
  Proposal 26: Change (rate multiplier) with Dirichlet proposal
    New proposal rate (<return> to keep old = 1.000):
    (提案頻度は変更しないときは空欄のままEnter)
    New Dirichlet parameter (<return> to keep old = 500.000): 50000
    (提案の過激さを変更する)
  中略
  Select a parameter to change (1 - 36; 0 to exit; 37 to zero all proposal rates): 0
  (設定変更を終了する)
```

proposal rate はそのパラメータの変更が提案される相対頻度で、値を大きくするとより変更の提案される頻度が高くなります。提案の受理率が高くても提案頻度が低い場合はこの値を変更します。提案の受理率が低すぎるのであれば、提案頻度を上げるよりももう一方の値 (提案の過激さを決定する) を変更した方が良いでしょう。この値は、大きいほど過激な提案がされたり、逆に小さいほど過激な提案がされたりとパラメータによって意味が変わりますので、MrBayes のマニュアルを見て大きくするか小さくするかを考えて下さい。多くの場合、提案が過激すぎて十中八九受理されないパラメータ値が提案されてしまっていることが多いでしょうから、提案を穏当にする方向へ値を変更すると良いでしょう。設定後に MCMC コマンドで MCMC を走らせ始めることで設定の適用された MCMC を走らせることができます。MCMC の途中で変更することはできません。

複数領域データやタンパクコード領域データでは、比例・分離モデルやコドン位置ごとに異なる置換速度を当てはめるモデルが適用されていることが多いと思います。しかし、MrBayes5D では比例モデルを適用しているときにパーティションごとの置換速度パラメータ (rate multiplier) を提案する Dirichlet proposal の受理率が異常に低くなることがしばしばあります。デフォルトでは Dirichlet parameter (提案の過激さを示す。小さいほど過激な提案がなされる) は 1000 (純正の MrBayes では 500) に設定されていますが、もっと大きな値にして提案を穏当にしてやることで改善できることがあります。比例関係にあるパーティション数が多い時にこの問題が起きやすいようです。また逆に、デフォルト値では提案が穏当すぎて最適化が進まず、収束に時間がかかってしまうこともあり得ます。

MrBayes5D は、同時に 2 つの MCMC を走らせていると書きましたが、その 2 つの MCMC のそれぞれはさらに 4 つの MCMC を同時に走らせています。4 つの中には乱数の乱れ (temperature) が大きい＝より過激な提案がなされる高温系列 (heated chain) が 3 つ (temperature は異なる) と、乱数の乱れが最も小さい低温系列 (cold chain) が 1 つあり、MCMC からのサンプリングはこの低温系列から行われています。各系列間ではモデル状態の交換が一定の頻度で試行されます。これを Metropolis-coupled MCMC、略して MC^3 と言います。パラレル・テンパリング法と呼ぶこともあります。こうすることで、より早く収束し、かき乱れが良くなるため、少ないステップ数で大きな ESS を得られます。状態交換 (state exchange) の試行が成立するかどうかは Metropolis *et al.*

(1953) および Hastings (1970) のルールに従って決定されます。これは前述のパラメータ変更に関しても同じです。

MCMC を停止すると表示されるメッセージの中に以下のようなものがあります。

```
Chain swap information for run 1:

           1     2     3     4
     -------------------------
   1 |        0.07  0.01  0.01
   2 | 10293        0.04  0.03
   3 |  9928 10392        0.05
   4 | 10394  9827  9919

中略

Upper diagonal: Proportion of successful state exchanges between chains
Lower diagonal: Number of attempted state exchanges between chains
```

これが状態交換試行の回数と交換の成立率です。1 が低温系列で、2~4 は順に温度が高くなっていく高温系列を示しています。温度の隣接した系列間の交換成立率が上記のように低い場合、温度の間隔 (標準では 0.2) を狭くしてやることで交換成立をしやすくすることで改善できる可能性があります。これは以下のようなコマンドで設定可能です。

```
MrBayes > MCMCP Temp=0.15
```

この設定後に MCMC コマンドで MCMC を走らせ始めると、上記の設定が適用された MCMC になります。

5.4　解析結果の要約

MCMC を停止したら、そのままでは何らかの意味を見出すのは難しいので、その結果を要約する必要があります。まず初めに、burn-in (収束前で捨てるサンプル数) を決めます。前述した Tracer とは違い、ここでの burn-in は解析開始からのステップ数ではなく、サンプル数です。つまり、100 ステップに 1 回サンプルする設定 (標準設定) で最初の 1,000,000 ステップを捨てるには、burn-in は 10,001 にします (MrBayes5D は初期状態＝第 1 ステップを保存するため 1 多くなる)。最初の何ステップを捨てるべきかを判断する方法は p75 の第 p5.3 節で説明しています。次に、MrBayes5D が生成する `.t` ファイルから要約を行います。MrBayes5D の SumT コマンドを用いる方法と、Phylogears2 を用いる方法があります。後者は複数の MCMC で burn-in の値が異なる場合にも対応できます。

SumT コマンドを用いて要約を行う場合、MrBayes5D に NEXUS データファイルを読み込ませた後、以下のようにコマンドを実行します。`integer` には burn-in するサンプル数を入力します。

```
MrBayes > SumT BurnIn=integer
```

これで.con ファイルと.parts ファイルが作成されます。.con は MCMC からのサンプル系統樹群から生成された多数決合意樹で、枝長は互換性のある系統樹群での平均値です。内分枝 (internal/interior branch) の出現頻度はこのファイルにも書かれていますが、.parts をテキストエディタで開くと、対立する内分枝も含めた支持率が書かれています。

Phylogears2 を用いる場合、まずは Phylogears2 の pgsplicetree で必要な系統樹だけを取り出します。以下のようにコマンドを実行します。

```
pgsplicetree from-to input_file output_file
```

from-to には取り出す系統樹の番号を入力します。10002-. などと指定します。これは、10,002 本目の系統樹から最後の系統樹までを出力ファイルに取り出すという意味です。これで、最初の 10,001 本の系統樹は burn-in されることになります。-500-. と指定すれば、最後から 500 本目の系統樹から最後の系統樹までを出力ファイルに取り出すことができます。複数の.t ファイルがある場合 (標準設定では 2 つできます)、以上の処理を全ての.t ファイルに対して行った後、pgjointree で出力したファイルを結合します。以下のようにコマンドを実行して下さい。

```
pgjointree input_file1 input_file2 output_file
```

入力ファイル名は 3 つ以上指定することも可能です。このファイルを pgsumtree に与えて出力を得ます。pgsumtree の使い方は p88 の第 6.4 節をご覧下さい。

その他の各種パラメータの要約は p75 の第 5.3 節をご参照下さい。

5.5 MrBayes5D MPI 版による並列計算

インストール方法のところで述べたように、MrBayes5D は MPI による並列化版 (Altekar *et al*., 2004) があり、これを用いることで大規模な解析を高速に行うことができます。~/に mrbayes5d-mpi として実行ファイルがあるとすると、起動するには以下のようにコマンドを実行します。

```
mpirun -np 利用するCPU数 ~/mrbayes5d-mpi -i NEXUSファイル名
```

なお、MPI フレームワークとして LAM/MPI をインストールした場合は mpirun で起動する前に lamboot -v を実行しておく必要があります。解析後は lamhalt を実行しておきます。起動後は通常版と同様に扱うことが

できますが、Props コマンドによる提案に関する各種パラメータの変更を正常に行うことができません。そのため、これらのパラメータを変更したい場合は、ソースコードに書かれているパラメータを直接書き換え、そのソースから作成した実行ファイルを用いる必要があります。当該箇所は mcmc.c の SetUpMoveTypes 関数にあります。

MrBayes5D では、標準では 4 系列 (NChains)×2 セット (NRuns) で合計 8 系列の MCMC が実行されます。この状態では最大で 8 つまでしか CPU を用いることができません。1 つの系列に複数の CPU を割り当てることができないためです。大量の CPU があっても、1 つの系列当たりの解析を高速化することはできません。系列数を増加させて温度間隔を狭くすることで系列間の状態交換試行が成立しやすくすることはできますが、劇的に高速化したりはしません。逆に 1 ステップ当たりの状態交換試行数 (NSwaps) を増やさないと、交換の絶対数が減ってしまって系列間の混合具合が悪くなってしまいます。NRuns を増やしても、必要なサンプル数を確保するためのステップ数を小さくすることはできますが、計算そのものを高速化はできません。大量の CPU を用いた高速化は将来のバージョンや他のソフトに期待して下さい。

第 6 章

系統樹の編集・統計と可視化

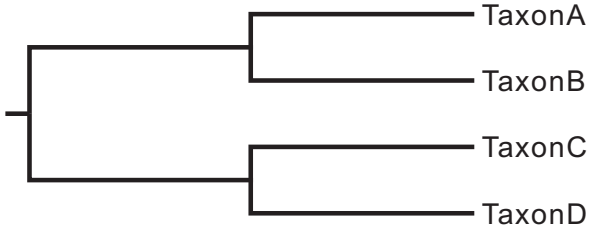
6.1　クレード・単系統・側系統・多系統・祖先的・派生的

ここではよく使う用語の定義を説明しておきます。

まず、クレード (clade) についてです。クレードとは、系統樹上で複数の OTU が所属する部分系統樹のことです。ただし、有根系統樹と無根系統樹ではやや意味が異なります。無根系統樹では、ある内分枝 (internal/interior branch) の一方の端点に接続されている部分系統樹をクレードと言いますが、有根系統樹では内分枝の根から遠い側の端点に接続されている部分系統樹を指します。つまり、有根系統樹上のクレードが根点を含むことはありません。

次に、単系統 (monophyly)・側系統 (paraphyly)・多系統 (polyphyly) です。有根系統樹上で、クレードを形成する分類群を単系統群 (monophyletic group) と呼びます。これに対して、メンバーを全て含んでいる最小のクレード内にメンバーでない単系統群を含み、かつ全メンバーに共通の祖先とそこから各 OTU までの枝に当たる生物がその分類群に分類されるような分類群を側系統群 (paraphyletic group) と言います。例えば魚類や両生類、爬虫類は側系統群です。メンバーを全て含んでいる最小のクレード内にメンバーでない単系統群を含み、かつ全メンバーに共通の祖先とそこから各 OTU までの枝に当たる生物のいずれかがその分類群に分類されないのが多系統群です。この定義では、共通祖先と共通祖先から各 OTU までの枝に当たる生物の形質状態が問題になり、特定できない状況ではこれらの言葉は使わないようにすべきだと思います。

図 6.1　単系統・側系統・多系統の例



念のため図 6.1 のような有根系統樹の場合を考えましょう。この系統樹では、(TaxonA, TaxonB)・(TaxonC,

TaxonD)・(TaxonA, TaxonB, TaxonC, TaxonD) は単系統群です。(TaxonA, TaxonB, TaxonC)・(TaxonA, TaxonB, TaxonD)・(TaxonA, TaxonC, TaxonD)・(TaxonB, TaxonC, TaxonD) は、根点に当たる共通祖先と根点から各 OTU までの枝が同一分類群に分類されるなら側系統群です。(TaxonA, TaxonC)・(TaxonA, TaxonD)・(TaxonB, TaxonC)・(TaxonB, TaxonD) は、共通祖先 (根点) か根点から各 OTU までの枝のどこかが同一分類群でないなら多系統で、同一分類群であると言えるなら側系統群です。

最後に、祖先的 (ancestral/plesiomorphic)・派生的 (derived/apomorphic) という言葉に関する注意点です。この言葉は二つの意味で使われています。一つは特定の形質 (やそれを有する OTU・単系統群) を指して実際により古くからあるものを祖先的、新しいものを派生的と言っている場合です。もう一つは、単に有根系統樹上でより根に近い (間にある分岐数が少ない) ものを祖先的、遠いものを派生的と言っている場合があります。この二つを混同しないように注意が必要です。というのも、根に近い単系統群の形質が祖先的であるとは限らないからです。

6.2　系統樹ファイルの形式と相互変換

系統樹のファイル形式は主に PHYLIP/Newick 形式と NEXUS 形式があります。PHYLIP/Newick 形式は以下のようなものです。

ファイルの内容 6.1　PHYLIP/Newick 形式系統樹

```
3
(TaxonA:0.1,TaxonB:0.1,(TaxonC:0.1,TaxonD:0.1):0.1);
(TaxonA:0.1,TaxonC:0.1,(TaxonB:0.1,TaxonD:0.1):0.1);
(TaxonA:0.1,TaxonD:0.1,(TaxonB:0.1,TaxonC:0.1):0.1);
```

最初の行はファイル中の系統樹の本数を示していますが、これは省略されていることもあります。コロン (:) の後ろの数字は枝長を示しています。PHYLIP 形式は、OTU 名に使用できる文字数が 10 文字までである点が Newick 形式との違いです。これに対して、NEXUS 形式は以下のようになっています。

ファイルの内容 6.2　NEXUS 形式系統樹

```
#NEXUS

Begin Trees;
  tree tree_1 = [&U] (TaxonA:0.1,TaxonB:0.1,(TaxonC:0.1,TaxonD:0.1):0.1);
  tree tree_2 = [&U] (TaxonA:0.1,TaxonC:0.1,(TaxonB:0.1,TaxonD:0.1):0.1);
  tree tree_3 = [&U] (TaxonA:0.1,TaxonD:0.1,(TaxonB:0.1,TaxonC:0.1):0.1);
End;
```

系統樹部分の体裁はほとんど同じですが、Trees ブロック内に書かれています。[&U] は、系統樹が無根系統樹であることを示しています。有根系統樹では [&R] になります。この記述は省略可能です。また、下記のように Translate コマンドを用いて系統樹内の OTU 名を数字に置き換えているものもあります。

ファイルの内容 6.3 Translate コマンド使用 NEXUS 形式系統樹

```
#NEXUS

Begin Trees;
  Translate
    1 TaxonA,
    2 TaxonB,
    3 TaxonC,
    4 TaxonD
    ;
  tree tree_1 = [&U] (1:0.1,2:0.1,(3:0.1,4:0.1):0.1);
  tree tree_2 = [&U] (1:0.1,3:0.1,(2:0.1,4:0.1):0.1);
  tree tree_3 = [&U] (1:0.1,4:0.1,(2:0.1,3:0.1):0.1);
End;
```

大量の系統樹を 1 ファイルに保存するときにはこちらの形式の方が容量は小さくなるでしょう。

6.2.1 Phylogears2 による変換

Phylogears2 には、系統樹ファイル形式を変換することができる `pgconvtree` コマンドが含まれています。PHYLIP/Newick・NEXUS に加えて Treefinder の TL Report 形式を読み込み、Newick/PHYLIP か NEXUS 形式へ書き出すことができます。使い方は下記のようになります。

```
pgconvtree --output=Newick input_file output_file
pgconvtree --output=NEXUS input_file output_file
```

Translate コマンドを使用している NEXUS 形式を読み込むことはできますが、書き出すことはできませんので注意して下さい。

6.2.2 Treefinder による変換

Treefinder は、独自形式に加えて PHYLIP/Newick・NEXUS 形式の読み書きができます。ただし、Phylogears2 と同様に Translate コマンドを使用している NEXUS 形式を読み込むことはできますが、書き出すことはできません。

グラフィカルインターフェイスの場合、`File` メニュー下にある `Open Image ...` から系統樹ファイルを指定して開いて下さい。系統樹が表示されますので、`File` メニュー下の `Export Tree ...` を選択します。`Save As` に出力ファイル名を指定して適宜出力ファイル形式を選択し、`OK` を押せばファイルが出力されます。

コマンドラインから操作する場合には `tf` コマンドを用います。何も指定せずに `tf` を実行すると対話型インターフェイスが起動します。このインターフェイス上でファイルを変換して出力するには以下のように入力します。

```
TL> SaveTreeList[LoadTreeList["input_file"],"output_file",Format->"NEWICK"]
```

以下のように入力しても同じ結果になります。

```
TL> "input_file",LoadTreeList,"output_file",Format->"NEWICK",SaveTreeList
```

対話型インターフェイスを終了するには、Quit を実行します。

上述の入力コマンドをテキストファイルに保存した上で、以下のように実行してやれば、テキストファイル内のコマンドが実行されるため、同様に入力ファイルの内容が変換されて出力ファイルに保存されます。

```
tf command_file
```

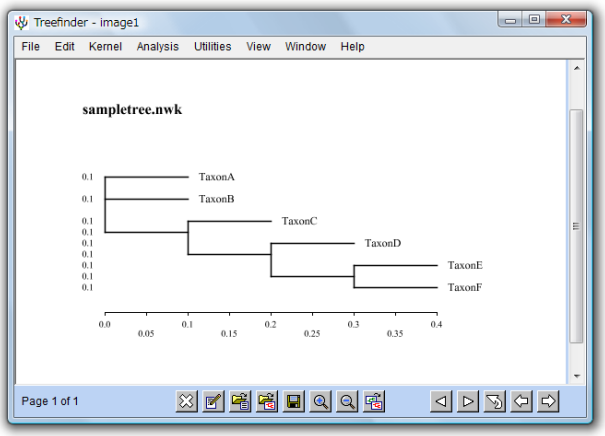
6.3　系統樹の有根化と樹形の変形

6.3.1　Treefinder による有根化と樹形改変

Treefinder では、柔軟な系統樹の編集が可能です。グラフィカルインターフェイスでもコマンドラインからでも操作できます。1 万本の系統樹を全て一発で有根化するような編集が可能なソフトは他に Mesquite くらいしかありませんが、コマンドラインから操作すれば Mesquite よりはるかに高速です。

グラフィカルインターフェイスの場合、まず File メニュー下にある Open Image ... から系統樹ファイルを指定して開いて下さい。系統樹が表示されます (図 6.2) ので、View メニュー下の Redraw ... を選択します。

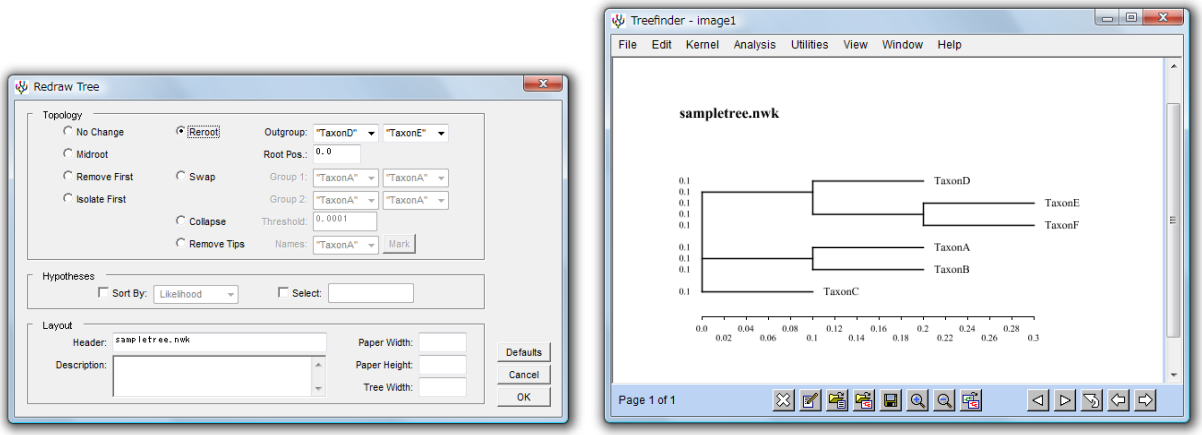
図 6.2　Treefinder による系統樹表示



すると、図 6.3 左のようなダイアログが出ますので、有根化したい場合は Reroot のラジオボタンを選択して

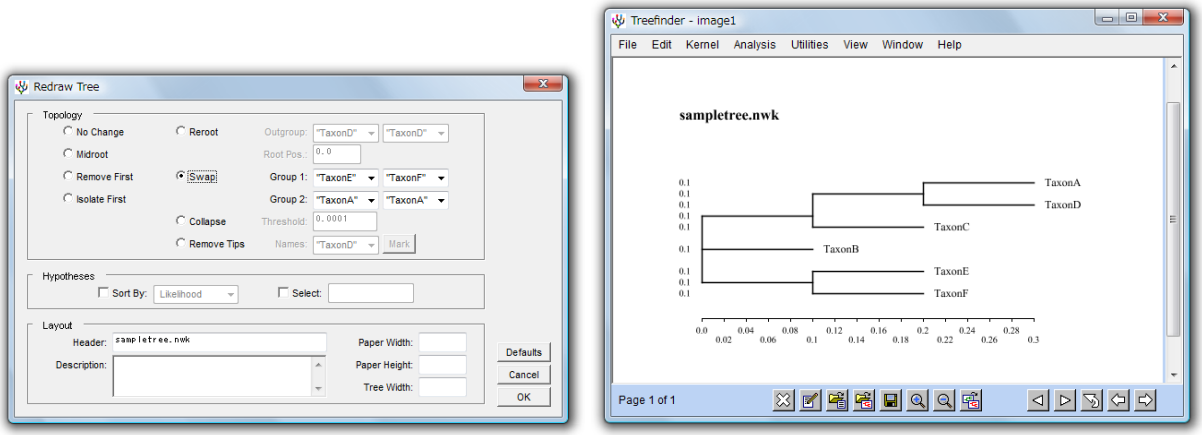
から外群に所属する OTU を選択します。2 つの OTU を指定すれば、その 2 つの OTU が共に所属する最小のクレードが外群になります。図 6.3 左のように `TaxonD` と `TaxonE` を指定して `OK` を押すと、図 6.3 右のように有根化がなされます。最初に読み込んだファイルに 100 本の系統樹が含まれていれば、同じ処理が全ての系統樹に対して行われます。

図 6.3　Treefinder による有根化



樹形を変更したい場合は、同じダイアログで `Swap` のラジオボタンを選択して、`Group 1` と `Group 2` に入れ替えたい OTU を指定します。2 つの OTU を指定すれば、その 2 つの OTU が共に所属する最小のクレードが入れ替え対象になります。図 6.4 左のように、`Group 1` に `TaxonE` と `TaxonF` を、`Group 2` に `TaxonA` を指定して `OK` を押すと、図 6.4 右の結果が得られます。

図 6.4　Treefinder による樹形改変



他にも、OTU・クレードを除去 (`Remove Tips`) したり、閾値以下の長さの枝を 0 にしたり (`Collapse`) することもできます。

コマンドラインから操作する場合には `tf` コマンドを用います。何も指定せずに `tf` を実行すると対話型インターフェイスが起動します。このインターフェイス上で系統樹を有根化して出力するには以下のように入力しま

す (改行は入れないで下さい)。

```
TL> SaveTreeList[RedrawTree[LoadTreeList["input_file"],Outgroup->{"TaxonA","TaxonB"}],
        "output_file",Format->"NEWICK"]
```

外群が 1OTU の場合は {} で囲む必要はありません。また、以下のように入力しても同じ結果になります。

```
TL> "input_file",LoadTreeList,Outgroup->{"TaxonA","TaxonB"},RedrawTree,"output_file",
        Format->"NEWICK",SaveTreeList
```

OTU やクレードを入れ替えて樹形を変更したい場合は、Outgroup->{"TaxonA","TaxonB"} の代わりに GroupsToSwap->{{"TaxonA","TaxonB"},{"TaxonC","TaxonD"}} を使います。OTU やクレードを削除するには、TipsToRemove->{"TaxonA","TaxonB"} とします。

対話型インターフェイスを終了するには、Quit を実行します。

上述の入力コマンドをテキストファイルに保存した上で、以下のように実行してやれば、テキストファイル内のコマンドが実行されるため、同様に入力ファイルの内容が変換されて出力ファイルに保存されます。

```
tf command_file
```

6.4　内分枝出現頻度の分析

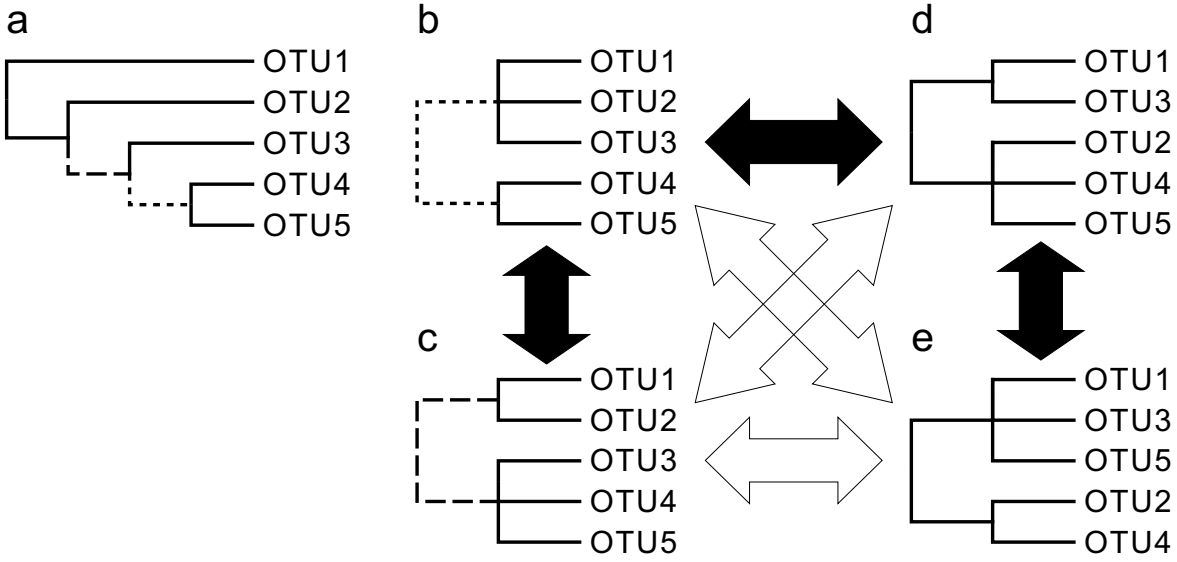
そもそも系統樹は複数の系統仮説の集合体です。たとえば、図 6.5a の最尤系統樹には図 6.5b, c のような系統仮説が含まれています。つまり図 6.5b, c の系統仮説は互いに矛盾せず同時に成立し得る＝互換性がある、と言えます。また、系統樹そのものもまた多数の互換性のある系統仮説が同時に成立するという系統仮説です。系統仮説の実体は系統樹上に現れる内分枝 (他の枝とのみ接している枝) なので、図 6.5b-e のように系統仮説もまた系統樹として表現することができます。

多数決合意樹を見れば、最も多く出現する系統仮説＝内分枝は分かりますが、それらと矛盾する内分枝の再現率も分かりません。そこで、Phylogears2 に含まれている pgsumtree を用いることで、ブートストラップ解析や MCMC で現れた全ての内分枝の出現頻度を得ることができます。pgtfboot コマンドを用いてブートストラップ解析を行った場合には以下の処理は自動的に実行されます。

まず、コマンドプロンプトやターミナルを起動して、ブートストラップ解析の各反復から得られた系統樹 (p68 の第 4.4 節に従って解析した場合は_bootstrap.nwk というファイル) が保存してあるフォルダに移動します。そして、以下のようにコマンドを実行します。

```
pgsumtree --mode=ALL *_bootstrap.nwk output_file
```

図 6.5 系統樹と系統仮説



a の系統樹を分解すると b, c の系統仮説になります。a で点線・破線の内分枝は b, c の同じ線の内分枝に対応しています。b-e の系統仮説間の矢印は黒塗りが互換性あり、白抜きが互換性無しということを意味します。

解析結果は入力ファイルと同じ形式の系統樹ファイルとなっています。仮に Newick 形式のファイルを入力ファイルとして与えて開いたとすると、下記のようになっているはずです。この例は 16OTU のデータで 100 反復のブートストラップ解析結果を pgsumtree で解析したものです。

ファイルの内容 6.4 内分枝出現頻度分析結果

```
1  [majorhypothesis_1] ((TaxonA,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonG,TaxonH,TaxonI,
       TaxonJ,TaxonK,TaxonL,TaxonM,TaxonN)100.0,(TaxonO,TaxonP));
2  [majorhypothesis_2] ((TaxonA,TaxonO,TaxonP,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonG,
       TaxonH,TaxonI,TaxonJ,TaxonM,TaxonN)100.0,(TaxonK,TaxonL));
3  [majorhypothesis_3] ((TaxonA,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonH,TaxonI,TaxonJ,
       TaxonK,TaxonL,TaxonM)100.0,(TaxonO,TaxonP,TaxonG,TaxonN));
4  [majorhypothesis_4] ((TaxonA,TaxonO,TaxonP,TaxonB,TaxonE,TaxonF,TaxonG,TaxonH,TaxonI,
       TaxonJ,TaxonK,TaxonL,TaxonM,TaxonN)100.0,(TaxonC,TaxonD));
5  [majorhypothesis_5] ((TaxonA,TaxonO,TaxonP,TaxonC,TaxonD,TaxonF,TaxonG,TaxonH,TaxonI,
       TaxonJ,TaxonK,TaxonL,TaxonM,TaxonN)98.0,(TaxonB,TaxonE));
6  [majorhypothesis_6] ((TaxonA,TaxonO,TaxonP,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonH,
       TaxonI,TaxonJ,TaxonK,TaxonL,TaxonM)85.0,(TaxonG,TaxonN));
7  略
8  [minorhypothesis_1] ((TaxonA,TaxonO,TaxonP,TaxonB,TaxonE,TaxonF,TaxonG,TaxonH,TaxonJ,
       TaxonK,TaxonL,TaxonM,TaxonN)25.0,(TaxonC,TaxonD,TaxonI));
9  [minorhypothesis_2] ((TaxonA,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonH,TaxonI,TaxonJ,
       TaxonK,TaxonL)21.0,(TaxonO,TaxonP,TaxonG,TaxonM,TaxonN));
10 [minorhypothesis_3] ((TaxonA,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonH,TaxonI,TaxonK,
       TaxonL,TaxonM)17.0,(TaxonO,TaxonP,TaxonG,TaxonJ,TaxonN));
11 [minorhypothesis_4] ((TaxonA,TaxonH,TaxonJ)15.0,(TaxonO,TaxonP,TaxonB,TaxonC,TaxonD,
       TaxonE,TaxonF,TaxonG,TaxonI,TaxonK,TaxonL,TaxonM,TaxonN));
12 [minorhypothesis_5] ((TaxonA,TaxonO,TaxonP,TaxonB,TaxonE,TaxonF,TaxonG,TaxonH,TaxonJ,
       TaxonK,TaxonL,TaxonN)14.0,(TaxonC,TaxonD,TaxonI,TaxonM));
13 [minorhypothesis_6] ((TaxonA,TaxonC,TaxonD,TaxonM)12.0,(TaxonO,TaxonP,TaxonB,TaxonE,
       TaxonF,TaxonG,TaxonH,TaxonI,TaxonJ,TaxonK,TaxonL,TaxonN));
14 略
```

majorhypothesis は多数決合意樹に出力された内分枝を表す系統樹で、全て互いに互換性があります。minorhypothesis は多数決合意樹とは矛盾する内分枝＝非互換な仮説を表す系統樹で、majorhypothesis のいずれか 1 つ以上の系統仮説と非互換な仮説群です。minorhypothesis の仮説間は互換性があるもの

も無いものも混じっています。いずれも系統樹にも出現頻度が含まれています。85% の確率で出現した majorhypothesis_6 という系統仮説は、TaxonG と TaxonN からなるクレードと、それ以外の OTU からなるクレードとを隔てる内分枝であることを表しています。これと非互換な系統仮説を探すには、minorhypothesis の中から探せばいいわけです。ただ、目視で探すのは面倒なので、それよりは多少楽で確実な方法を用意してあります。まずは pgsplicetree コマンドを用いて majorhypothesis_6 だけを別ファイル (仮に majorhypothesis_6.nwk とする) に取り出します。

```
pgsplicetree 6 input_file majorhypothesis_6.nwk
```

その上で、以下のようにしてこの出力ファイル内の系統樹と非互換な系統仮説をブートストラップ解析や MCMC の結果から探し出します。

```
pgsumtree --mode=ALLi --treefile=majorhypothesis_6.nwk *_bootstrap.nwk output_file
```

出力結果をテキストエディタで開くと以下のようになっています。

ファイルの内容 6.5　内分枝出現頻度分析結果

```
1 [majorincompatible_1_of_tree_1] ((TaxonA,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonH,
      TaxonI,TaxonJ,TaxonK,TaxonL,TaxonM,TaxonN)8.0,(TaxonO,TaxonP,TaxonG));
2 [minorincompatible_1_of_tree_1] ((TaxonA,TaxonB,TaxonC,TaxonD,TaxonE,TaxonF,TaxonG,
      TaxonH,TaxonI,TaxonJ,TaxonK,TaxonL,TaxonM)7.0,(TaxonO,TaxonP,TaxonN));
```

majorincompatible_N_of_tree_K は、入力ファイル内で見られる系統仮説の中で、--treefile オプションで指定した系統樹ファイルの K 番目の系統樹と非互換なもので、かつ N 番目に出現頻度の高いものです。N が 2 以上のものもあるかもしれませんが、これは N=1 の系統仮説と互換性があるということです。minorincompatible は majorincompatible のどれか 1 つ以上の仮説と非互換な仮説であることを表しています。majorincompatible の仮説間では互換性がありますが、minorincompatible の仮説間では互換性があったり無かったりします。majorincompatible_1 は非互換な仮説の中で出現頻度最大なので第 2 位の仮説と言えるでしょう。minorincompatible_1 は第 3 位の仮説と考えられます。第 4 位以下の仮説を探すには、1 位から 3 位までの全ての仮説のいずれとも非互換な仮説を探さなくてはなりませんが、まだその方法は用意していません。出現頻度はブートストラップ解析の反復数・MCMC のサンプル数が小さいとかなり変動しますので、第 2 位の仮説が本当に第 2 位かどうかはよく検討する必要があります。

第 7 章

仮説検定

p88 の第 6.4 節で述べたようにして非互換な系統仮説を探すことができます。また、過去の論文から非互換な系統仮説を得られることもあるでしょう。いずれかの仮説を厳密に棄却できるかどうかを検討するには、各系統仮説を制約として課した系統推定の結果を比較します。ここでは Treefinder による制約付き最尤系統推定と MrBayes5D を用いた制約付きベイジアン系統推定の方法と、その結果に基づいた KH・SH・AU 検定、パラメトリックブートストラップ検定、Bayes factor による仮説比較について説明します。

7.1 Treefinder による樹形制約付き最尤系統推定

Treefinder で樹形制約 (topological constraint) を課した系統推定を行うには、まず制約となる系統樹を作成する必要があります。例えば、TaxonA~TaxonE の 5 OTU のデータで TaxonA と TaxonB の単系統性 (monophyly) を制約として課す場合、以下のような系統樹ファイルを用意します。

ファイルの内容 7.1　樹形制約を指定する系統樹ファイル

```
1 ((TaxonA,TaxonB),TaxonC,TaxonD,TaxonE);
```

以下のようにしても無根系統樹として見れば意味は同じです。

ファイルの内容 7.2　樹形制約を指定する系統樹ファイル

```
1 ((TaxonA,TaxonB),(TaxonC,TaxonD,TaxonE));
```

TaxonA と TaxonB の単系統性だけでなく、さらに TaxonA と TaxonB と TaxonC の単系統性も課すには、以下のようなファイルにします。

ファイルの内容 7.3　樹形制約を指定する系統樹ファイル

```
1 (((TaxonA,TaxonB),TaxonC),TaxonD,TaxonE);
```

このように、「特定の系統仮説を満たす」樹形制約を正の制約 (positive constraint) と言います。正の制約下の系統推定では、その系統仮説と互換性のある系統樹の中でベストな系統樹を探索することになります。「特定の系統仮説を満たさない」という制約もあり、これを負の制約 (negative constraint) と呼びます。負の制約下の系統推定では、その系統仮説と互換性の無い系統樹の中でベストな系統樹を探索することになります。Treefinder は負の制約に対応していないため、単系統「でない」という制約を課すことができません。しかし、ブートストラップ解析結果から得られる内分枝出現頻度を見れば、その負の制約下で最も尤度の高い樹形を含む正の制約＝第 2 位の系統仮説が推定できますので、それを課した樹形探索を行うことで対処することができます。つまり、ある仮説とは矛盾する仮説 (= 負の制約) の中で最も高頻度に再現される仮説へと、負の制約を正の制約へ読み替えてしまえばよいということです。

制約を表す系統樹ファイルが用意できたら、

partition_criterion_xxx_singlesearch.tl

をテキストエディタで開いて以下のような内容に編集します。念のため別名で保存しましょう。

ファイルの内容 7.4　変更後のファイル

```
report:=ReconstructPhylogeny[
    "partition_xxx.tf",
    SubstitutionModel->Load[
        "partition_criterion_xxx.model"
    ],
    PartitionRates->Load[
        "partition_criterion_xxx.rates"
    ],
    Tree->"constraint.nwk",
    ResolveMultifurcations->True,
    WithEdgeSupport->False,
    SearchDepth->2,
    AcceptFlatness->True
],
Oprec[
    20,
    SaveReport[
        AsReport[
            report
        ],
        "partition_criterion_xxx_treesearch_constraint.log"
    ],
    Save[
        report|1|SubstitutionModel,
        "partition_criterion_xxx_optimum_constraint.model"
    ],
    Save[
        report|1|PartitionRates,
        "partition_criterion_xxx_optimum_constraint.rates"
    ],
    SaveTree[
        AsTree[
            report|1|Phylogeny
        ],
        "partition_criterion_xxx_optimum_constraint.nwk",
        Format->"NEWICK"
    ]
]
```

変更点は以下の通りです。

- Tree->"constraint.nwk", の行を追加する

- `ResolveMultifurcations->True,` の行を追加する
- 出力ファイル名を適宜変更する

ただし、樹形制約を課した樹形探索ではなく、樹形を完全に固定しての尤度最大化を行う場合は 2 つ目は必要ありません。ファイルが用意できたら、Treefinder の `Kernel` メニューにある `Load TL Script ...` からこのファイルを開くことで内容が実行されます。

制約付き likelihood ratchet を行うには、`pgtfratchet` の下記の質問の際に制約系統樹ファイルの名前を与えてやればいいだけです。

```
If you want to give topological constraint, specify an input file name.
Otherwise, just press enter.
```

`pgtfratchet` で初期系統樹作成を TNT にさせる場合、ファイル 7.3 のような制約を課そうとするとうまくいかないことがあります。これは、TNT では必ず外群を指定する必要があり、外群を正の制約内に含めることができないという制限があるためです。`pgtfratchet` は制約樹形の先頭にある OTU を外群とするため TaxonA が外群になりますが、TaxonA を含む正の制約を課そうとしているのでうまくいかないわけです。制約樹形の先頭にある OTU を外群とするのですから、以下のように記述すれば問題無く動作するはずです。

ファイルの内容 7.5　樹形制約を指定する系統樹ファイル

```
1 (TaxonD,TaxonE,((TaxonA,TaxonB),TaxonC));
```

これでも制約の内容は全く同じですが、TaxonD が外群として使われるため問題が起きません。ファイル 7.2 の制約は TaxonA と TaxonB の単系統性制約と考えるとうまくいかないように思えますが、TaxonC と TaxonD と TaxonE の単系統性制約として自動的に変換されるので問題無いはずです。ただし、この変換は外群となる先頭 OTU を含む制約を消すことで実現しているため、注意しないと意図していない制約へと変えられてしまう可能性があるので十分注意して下さい。

7.2 Treefinder による仮説検定

複数の系統仮説を比較したいとき、それぞれの仮説を制約として課した最尤系統推定によって得られた制約下の最尤系統樹を比較してやることで、どの仮説が他の仮説より良いか、それは有意な違いかを調べることができます。また、特定の単系統性を検証したいときは、その単系統性の制約下の最尤系統樹と、その単系統性とは矛盾する仮説、即ち負の制約下の最尤系統樹とを比較してやればよいでしょう。ここではブートストラップリサンプリングを応用した検定法と、モンテカルロシミュレーションを応用したパラメトリックブートストラップ検定の実行方法について説明します。

7.2.1　KH・SH・AU 検定

ブートストラップリサンプリングによって、複数の系統樹間で尤度の差が有意と言えるのか否かを調べる方法が Kishino-Hasegawa 検定 (KH test) です (Kishino and Hasegawa, 1989)。しかし、この方法では 3 つ以上の系統樹を比較する場合に多重検定となってしまい第 1 種の過誤 (有意な差は無いのに誤検出する) が増大してしまうため、その抑制を行う補正を加えたのが Shimodaira-Hasegawa 検定 (SH test) です (Shimodaira and Hasegawa, 1999)。ただし、この方法では逆に第 2 種の過誤 (有意な差があるのに検出できない) が増大してしまいます。そこで、近似的に不偏な検定 (approximately unbiased [AU] test) は、マルチスケールブートストラップ法を用いてさらに高度な補正を行うことでこれをある程度解決しています (Shimodaira, 2002)。

Treefinder でこれらの検定を行うには、あらかじめ比較する系統樹・尤度・分子進化モデルを最尤法で推定しておきます。これまでの説明の通りに解析を行っていれば、`.log` というファイルができているはずです。それぞれの系統樹の`.log` ファイルが用意できているなら、それを結合します。

`.log` ファイル (Treefinder では Report File と呼んでいる) を結合するには、Treefinder の `Utilities` メニューにある `Join Report Files ...` を用います。結合元のファイルと結合後の出力ファイルを 1 つ 1 つ指定して `OK` を押せば結合した Report File が作成されます。また、Phylogears2 にも `pgtfjoinlog` というコマンドがあり、これを用いて同様のことができます。コマンドプロンプトやターミナルで以下のように用います。入力ファイルは 3 つ以上指定することもできます。

```
pgtfjoinlog input_file1 input_file2 output_file
```

結合した Report File の用意ができたら、`Analysis` メニューの `Test Hypotheses ...` からダイアログを呼び出します。`Sequence File` には Report File の系統推定に用いたデータファイル (ここまでの説明通りなら `partition_xxx.tf`) を指定し、`Hypothesis File` に先ほど作成した Report File を指定します。`# Replicates` (ブートストラップリサンプリングの反復数) を適当な値 (AU 検定を使うなら 10 万以上) に、`Criterion` を `Likelihood` に設定して `OK` を押すことで計算が開始されます。しばらく待って計算が終わると、結果が画面に表示されます。この結果は、`File` メニューの `Export Tree ...` から Report File として、`Save` から PostScript 画像として保存できます。結果は、Report File にある系統樹ごとに検定結果が表示されます。KH や SH 検定の p 値が 1 になっているのが最尤系統樹で、最尤系統樹の p 値には意味がありません。画面下部の ▹ のボタンを押すと別の系統樹の結果が出ますので、最尤でない系統樹の結果を見て下さい。

7.2.2 パラメトリックブートストラップ検定

パラメトリックブートストラップ検定 (parametric bootstrap test) とは、帰無仮説に基づいたモンテカルロシミュレーションによって生成したデータに帰無仮説と対立仮説を当てはめて尤度最大化し、その尤度比の分布に対して元データにおける尤度比がどうなるかを検討することで帰無仮説を棄却する (または棄却しない) 検定法です。つまり、「帰無仮説が正しい」という仮定の下で生成したデータでの尤度比よりも元データでの尤度比が有意に大きいなら、それは「帰無仮説が正しい」という仮定が間違っていると考えられるので、帰無仮説を棄却できるということです。

この方法の特長は、系統推定法そのものに内在する問題を検出可能であるという点にあります。本当は帰無仮説とされた方が正しいにもかかわらず、系統推定法に問題があるせいで対立仮説がより尤度が高くなってしまうのなら、帰無仮説の下で生成したデータにおける尤度比は元データにおける尤度比と大きく変わらないはずです。そのため、系統推定法に問題があるのなら帰無仮説を棄却できなくなります。前述の KH・SH・AU 検定ではこのような能力は**全く**ありません。ただし、帰無仮説が棄却できたからといって、推定方法に問題は無いと言い切れるわけではありません。

Treefinder でパラメトリックブートストラップ検定を行うには、前述の検定と同様に Report File が必要です。ただし、結合する必要はありません。Report File が用意できたら、`Analysis` メニューの `Parametric Bootstrap Test ...` からダイアログを呼び出します。表示されるダイアログで、`H0` に帰無仮説となる = データ生成に用いられる Report File を、`H1` に対立仮説の Report File を指定します。`# Replicates` (反復数) を適当な値に、`Criterion` を `Likelihood` に設定して `OK` を押します。しばらくして計算が終わると、結果が表示されます。計算にはかなりの時間を要します。Treefinder の現行バージョンでは、帰無仮説が対立仮説より有意に劣るのは p 値が 0.95 以上のときです。帰無仮説が棄却できないなら、系統推定の方法に問題があるか、2 つの系統仮説間に有意な差があるとは言えないと考えられます。

推定方法に問題が無ければ (あってもこの検定で検出できるような問題でなければ)、この検定では 2 つの仮説がよほど僅差でない限り、ほとんどの場合で帰無仮説は棄却されます。KH・SH・AU 検定の結果とこの検定の結果が一致する場合には何も悩む必要はありませんが、矛盾する場合には少し解釈を考える必要があります。KH・SH・AU 検定で棄却された仮説がこの検定で棄却できない場合は、2 つの仮説間に有意な差があるものの系統推定法に問題があるため確実とは言えない、と考えるべきでしょう。KH・SH・AU 検定で棄却できなかった仮説がこの検定で棄却された場合の解釈方法に関してはおそらく幅広いコンセンサスが得られていないと思いますが、筆者個人としては、系統推定法に問題は (おそらく) 無いが仮説間の尤度の差は有意ではない、とするのが安全だろうと考えています。

7.3　MrBayes5D による樹形制約付きベイジアン系統推定

Treefinder と同様に、TaxonA~TaxonE の 5 OTU のデータで TaxonA と TaxonB の単系統性 (monophyly) を制約として課す場合を考えましょう。その場合、以下のようなコマンドを NEXUS データファイル読み込み後に実行することで樹形探索に制約が課されるようになります。コマンドを NEXUS ファイルの MrBayes ブロックに記述しても結構です (行末にはセミコロンを付加する必要があります)。

```
MrBayes > Constraint monophyly1 100=TaxonA TaxonB
MrBayes > PrSet TopologyPr=Constraints(monophyly1)
```

さらに TaxonA と TaxonB と TaxonC の単系統性も強制する場合は以下のようにします。

```
MrBayes > Constraint monophyly1 100=TaxonA TaxonB
MrBayes > Constraint monophyly2 100=TaxonA TaxonB TaxonC
MrBayes > PrSet TopologyPr=Constraints(monophyly1,monophyly2)
```

MrBayes5D も Treefinder と同様に負の制約には対応していません。負の制約を課したい場合には内分枝出現頻度から負の制約を正の制約へ読み替えることで対処する必要があります。

7.4　Bayes factor に基づく仮説比較

複数の系統仮説を比較したいとき、それぞれの仮説を制約として課した解析結果を比較してやることでどちらの仮説が正しいかを検証することができます。ベイズ統計学では、そのような目的に Bayes factor (Kass and Raftery, 1995) というものを用います。これは周辺尤度の比に当たります。

多くの分子系統学の論文では、MCMC における周辺尤度の調和平均に基づいて Bayes factor を算出しますが、この方法では Bayes factor が安定せず、同じ解析を別々に実行してどちらも同じところへ収束していても、どちらか一方を支持してしまう結果を得てしまうことがよくあります。十分に安定した Bayes factor を得るには、非常に長い MCMC を走らせなくてはなりません。そこで Tracer には、ブートストラップリサンプリングを応用して少ないサンプルからでも高精度に Bayes factor を算出する機能が実装されています (Newton and Raftery, 1994)。この機能を用いることで、現実的な計算量で Bayes factor を利用した仮説選択が可能です。ここでは、樹形制約 1 を課した NEXUS ファイル `constraint1.nex` の解析結果と、樹形制約 2 を課した NEXUS ファイル `constraint2.nex` の解析結果を比較する場合を考えます。

MCMC が終わっていれば、`constraint1.nex.run1.p` と `constraint1.nex.run2.p`、`constraint2.nex.run1.p`、`constraint2.nex.run2.p` の 4 つのファイルができているはずです。それぞれの burn-in を決定し (ただ

しステップ数ではなくサンプル数)、Phylogears2 の pgmbburninparam コマンドで 2 つの burn-in 済のログファイルを作成します。それぞれの burn-in を 10001、20001、15001、15001、作成するファイルは constraint1_param.txt と constraint2_param.txt だとしておくと、コマンドプロンプトかターミナルで以下のようにします。

```
pgmbburninparam --burnin=10001 constraint1.nex.run1.p constraint1_param.txt
pgmbburninparam --burnin=20001 --append constraint1.nex.run2.p constraint1_param.txt
pgmbburninparam --burnin=15001 constraint2.nex.run1.p constraint2_param.txt
pgmbburninparam --burnin=15001 --append constraint2.nex.run2.p constraint2_param.txt
```

これで、それぞれの樹形制約を課した解析結果の burn-in 済ログファイルが作成できます。

次に、Tracer を起動し、File メニューの Import Trace File... から constraint1_param.txt と constraint2_param.txt を読み込ませます。そして、左上 Trace Files ペインで Burn-In を両方とも 0 にしてから、両ファイルを選択して反転表示状態にし、Analysis メニューの Calculate Bayes Factors... からダイアログを呼び出します。ダイアログでは、Likelihood trace を LnL に、Calculate harmonic mean only (no smoothing) のチェックを外し、Bootstrap replicates を 1000 以上に設定し、計算を実行します。計算が終わると表が示されるので、Show を ln Bayes Factors に設定します。Trace 列が対立仮説のファイル名、ln Bayes factor の値の列名が帰無仮説のファイル名となっています。ln Bayes factor の値から、表 7.1 の基準で仮説の優劣を判断します (Kass and Raftery, 1995)。

表 7.1 Bayes factor の値と仮説間の優劣

ln Bayes factor	帰無仮説に対して対立仮説が
1~3	より優れている
3~5	強く支持されている
5~	非常に強く支持されている

この方法にも多重比較の問題はあるはずですが、これまでのところそのための補正方法などは普及していません。

前述の通り、MrBayes5D は 2 つの MCMC を同時に走らせています。この 2 つの MCMC 間でも Bayes factor を算出することができます。もしその 2 つの MCMC がパラメータ空間上の同じ辺りに収束しているのなら、その Bayes factor によってどちらか一方が支持されることはないはずです。というわけで、「Bayes factor によってどちらか一方への支持が得られてしまう」か否かを収束判定に用いることもできるでしょう。ただ、この方法では「収束していない」ということは分かりますが、「収束している」ということは言えないので注意して下さい。

第 8 章

参考書籍

最後に、いくつか参考書籍を挙げておきます。

8.1 分子系統学

まず、分子系統学と分子系統解析に関する情報がまとまっている本としては以下の 3 冊が良いと思います。

分子進化と分子系統学

著者 根井正利, Sudhir Kumar

出版社 培風館

ISBN13 978-4563078010

分子進化学を黎明期から支えてこられた根井先生と Kumar 博士が書かれた本の邦訳です。日本語で分子系統学について幅広く説明されています。分子系統学を体系的に概観するには英語でもこれを上回る本はほとんど無いと思います。

分子系統学への統計的アプローチ－計算分子進化学

著者 Ziheng Yang

出版社 共立出版

ISBN13 978-4320056770

分子系統解析法開発の第一人者 Yang 博士が書かれた本の邦訳。最尤法・ベイズ法や、最先端のトピックスまで扱われた良書です。

Inferring Phylogenies

著者　Joseph Felsenstein

出版社　Sinauer Associates Inc.

ISBN13　978-0878931774

分子系統解析に最尤法やブートストラップ法を導入した Felsenstein 博士による系統推定法を網羅的に解説した決定版的書籍です。

The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing

編者　Philippe Lemey, Marco Salemi, Anne-Mieke Vandamme

出版社　Cambridge University Press

ISBN13　978-0521730716

タイトルから分かる通り英語です。とは言え、ソフトウェアの使用方法の解説部分は、使いながら見ればさほど難しいものではないと思います。旧版から大幅に改訂され最新のソフトウェアまでカバーしています。

8.2　統計学

分子系統学は、ある種の「超」応用統計学です。ですから、当然統計学の知識が役に立つ、というか必要になってきます。この本で触れている方法に関連する統計解析法について書かれている本を紹介します。

モデル選択－予測・検定・推定の交差点

著者　下平英寿, 伊藤 秀一, 久保川達也, 竹内啓

出版社　岩波書店

ISBN13　978-4000068437

AIC の導出過程や KH・SH・AU 検定までも説明されています。これらの検定法を使われる方は是非ご一読下さい。

ベイズ統計と統計物理

著者 伊庭幸人

出版社 岩波書店

ISBN13 978-4000111584

ベイジアン MCMC についておそらく最も易しく説明されている本です。MrBayes を使いながら読むとパラメータの意味が良く分かるだろうと思います。

計算統計 II －マルコフ連鎖モンテカルロ法とその周辺

著者 伊庭幸人, 種村正美, 大森裕浩, 和合肇, 佐藤整尚, 高橋明彦

出版社 岩波書店

ISBN13 978-4000068529

ベイジアン MCMC についてもっと深く知りたい方のための本です。

8.3 UNIX 入門

分子系統解析を行うソフトウェアは、UNIX の関連知識があると大変楽に使うことができます。以下では Windows 上で UNIX ライクな環境を構築できる Cygwin の入門書、Linux の中でも初心者でも比較的取っ付きやすい Ubuntu Linux の入門書、MacOS X を UNIX として使うための入門書、シェルの入門書を挙げます。CD や DVD が付属しているものもありますが、この世界は進歩が早いので、ソフトウェアは Web から最新版をダウンロードするようにしましょう。なお、以下の本は必ずしも私は読んではいません。

ちなみに、私が主に使っている UNIX は Gentoo Linux という、マイナーなものです。極限までカスタマイズ・チューニングができるのが特徴です。コンピュータの性能を限界まで引き出したい方は検討されてみるとよいでしょう。公式サイトのハンドブックが大変よくできていますのである程度の UNIX 利用経験があれば簡単に使えるようになると思います。

UNIX が使えるようになったら、SSH という遠隔操作するためのソフトウェアと、GNU screen または tmux というソフトを是非インストールしましょう。これらを組み合わせることで、遠隔地からインターネット経由で自宅や研究室の高速なコンピュータに接続して系統解析を行わせ、さらに行わせたままで接続を切ったり再接続したりすることができるようになります。使用方法は、検索すれば説明してくれている Web ページがすぐに見つかります。

Windows で使える UNIX 環境ー Cygwin 徹底入門

著者　小川淳一

出版社　ソーテック社

ISBN13　978-4881663622

Windows で UNIX を使う本ー Cygwin で UNIX 入門

著者　阿久津良和

出版社　毎日コミュニケーションズ

ISBN13　978-4839911959

はじめての Ubuntu ー超初心者向け Linux を使いこなす

著者　天野友道

出版社　工学社

ISBN13　978-4777513086

Ubuntu スタートアップバイブル

著者　佐々木宣文

出版社　毎日コミュニケーションズ

ISBN13　978-4839930691

MacOS X ユーザのための UNIX 入門ーターミナルから覗く UNIX の世界

著者　大津真

出版社　毎日コミュニケーションズ

ISBN13　978-4839909574

入門 Unix for Mac OS X

著者　Dave Taylor

出版社　オライリージャパン

ISBN13 978-4873112749

シェルの基本テクニック

著者 西村めぐみ

出版社 IDG ジャパン

ISBN13 978-4872802252

UNIX シェル入門ー bash の基本操作と UNIX の環境設定

著者 北浦訓行, 小島範幸

出版社 技術評論社

ISBN13 978-4774139203

引用文献

Ababneh, F., Jermiin, L. S., Ma, C., and Robinson, J., 2006, “Matched-pairs tests of homogeneity with applications to homologous nucleotide sequences”, *Bioinformatics*, **22**, 1225–1231.

Abascal, F., Posada, D., and Zardoya, R., 2007, “MtArt: a new model of amino acid replacement for Arthropoda”, *Molecular Biology and Evolution*, **24**, 1–5.

Adachi, J. and Hasegawa, M., 1996, “MOLPHY version 2.3: programs for molecular phylogenetics based in maximum likelihood”, *Computer Science Monographs*, **28**, 1–150.

Adachi, J., Waddell, P. J., Martin, W., and Hasegawa, M., 2000, “Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA”, *Journal of Molecular Evolution*, **50**, 348–358.

Akaike, H., 1974, “New look at statistical-model identification”, *IEEE Transactions on Automatic Control*, **19**, 716–723.

Altekar, G., Dwarkadas, S., Huelsenbeck, J. P., and Ronquist, F., 2004, “Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference”, *Bioinformatics*, **20**, 407–415.

Blanquart, S. and Lartillot, N., 2006, “A Bayesian compound stochastic process for modeling nonstationary and nonhomogeneous sequence evolution”, *Molecular Biology and Evolution*, **23**, No. 11, 2058–2071, Nov.

——— , 2008, “A site- and time-heterogeneous model of amino acid replacement”, *Molecular Biology and Evolution*, **25**, No. 5, 842–858, May.

Boussau, B. and Gouy, M., 2006, “Efficient likelihood computations with nonreversible models of evolution”, *Systematic Biology*, **55**, No. 5, 756–768, Oct.

Cao, Y., Janke, A., Waddell, P. J., Westerman, M., Takenaka, O., Murata, S., Okada, N., Pääbo, S., and Hasegawa, M., 1998, “Conflict among individual mitochondrial proteins in resolving the phylogeny of eutherian orders.”, *Journal of Molecular Evolution*, **47**, 307–322.

Capella-Gutiérrez, S., Silla-Martínez, J. M., and Gabaldón, T., 2009, “trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses”, *Bioinformatics*, **25**, No. 15, 1972–1973, Aug.

Castresana, J., 2000, “Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis”, *Molecular Biology and Evolution*, **17**, No. 4, 540–552, Apr.

Cochran, W. G., 1954, “Some methods for strengthening the common χ^2 tests”, *Biometrics*, **10**, 417–451.

Criscuolo, A. and Gribaldo, S., 2010, “BMGE (Block Mapping and Gathering with Entropy): a new software for

selection of phylogenetic informative regions from multiple sequence alignments", *BMC Evolutionary Biology*, **10**, 210.

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C., 1978, "A model of evolutionary change in proteins, Vol. 5, Suppl. 3", in Dayhoff, M. O. ed. *Atlas of Protein Sequence Structure*: National Biomedical Research Foundation, 345–352.

Dimmic, M. W., Rest, J. S., Mindell, D. P., and Goldstein, R. A., 2002, "rtREV: an amino acid substitution matrix for inference of retrovirus and reverse transcriptase phylogeny", *Journal of Molecular Evolution*, **55**, 65–73.

Edgar, R. C., 2004, "MUSCLE: multiple sequence alignment with high accuracy and high throughput", *Nucleic Acids Research*, **32**, No. 5, 1792–1797.

Felsenstein, J., 1981, "Evolutionary trees from DNA sequencies - a maximum-likelihood approach", *Journal of Molecular Evolution*, **17**, 368–376.

——— , 1985, "Confidence-limits on phylogenies - an approach using the bootstrap", *Evolution*, **39**, 783–791.

Fleissner, R., Metzler, D., and von Haeseler, A., 2005, "Simultaneous statistical multiple alignment and phylogeny reconstruction", *Systematic Biology*, **54**, 548–561.

Hastings, W. K., 1970, "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika*, **57**, 97–109.

Henikoff, S. and Henikoff, J. G., 1992, "Amino acid substitution matrices from protein blocks", *Proceedings of the National Academy of Sciences of the United States of America*, **89**, 10915–10919.

Hrdy, I., Hirt, R. P., Dolezal, P., Bardonová, L., Foster, P. G., Tachezy, J., and Embley, T. M., 2004, "Trichomonas hydrogenosomes contain the NADH dehydrogenase module of mitochondrial complex I", *Nature*, **432**, No. 7017, 618–622, Dec.

Jobb, G., 2008, "Treefinder version of April 2008", Software distributed by the author at http://www.treefinder.de/.

Jobb, G., von Haeseler, A., and Strimmer, K., 2004, "Treefinder: a powerful graphical analysis environment for molecular phylogenetics", *BMC Evolutionary Biology*, **4**, 18.

Jones, D. T., Taylor, W. R., and Thornton, J. M., 1992, "The rapid generation of mutation data matrices from protein sequences", *Computer Applications in the Biosciences*, **8**, 275–282.

Jukes, T. H. and Cantor, C. R., 1969, "Evolution of protein molecules", in Munro, H. N. ed. *Mammalian protein metabolism*, New York: Academic Press, 21–132.

Kass, R. E. and Raftery, A. E., 1995, "Bayes Factors", *Journal of the American Statistical Association*, **90**, 773–795.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R., 1998, "Markov chain Monte Carlo in practice: a roundtable discussion", *American Statistician*, **52**, 93–100.

Katoh, K., Kuma, K., Toh, H., and Miyata, T., 2005, "MAFFT version 5: improvement in accuracy of multiple sequence alignment", *Nucleic Acids Research*, **33**, 511–518.

Kimura, M., 1980, "A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences", *Journal of Molecular Evolution*, **16**, 111–120.

——— , 1983, *The neutral theory of molecular evolution*: Cambridge University Press.

Kishino, H. and Hasegawa, M., 1989, "Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea", *Journal of Molecular Evolution*, **29**, 170–179.

Kosiol, C. and Goldman, N., 2005, "Different versions of the Dayhoff rate matrix", *Molecular Biology and Evolution*, **22**, 193–199.

Larkin, M. A., Blackshields, G., Brown, N. P., Chenna, R., McGettigan, P. A., McWilliam, H., Valentin, F., Wallace, I. M., Wilm, A., Lopez, R., Thompson, J. D., Gibson, T. J., and Higgins, D. G., 2007, "Clustal W and Clustal X version 2.0", *Bioinformatics*, **23**, No. 21, 2947–2948, Nov.

Le, S. Q. and Gascuel, O., 2008, "An improved general amino acid replacement matrix", *Molecular Biology and Evolution*, **25**, 1307–1320.

Lunter, G., Miklós, I., Drummond, A., Jensen, J. L., and Hein, J., 2005, "Bayesian coestimation of phylogeny and sequence alignment", *BMC Bioinformatics*, **6**, 83.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., and Teller, A. H., 1953, "Equation of state calculations by fast computing machines", *Journal of Chemical Physics*, **21**, 1087–1092.

Müller, T. and Vingron, M., 2000, "Modeling amino acid replacement", *Journal of Computational Biology*, **7**, 761–776.

Newton, M. A. and Raftery, A. E., 1994, "Approximate Bayesian inference with the weighted likelihood bootstrap", *Journal of the Royal Statistical Society*, **56**, 3–48.

Nickle, D. C., Heath, L., Jensen, M. A., Gilbert, P. B., Mullins, J. I., and Pond, S. L. K., 2007, "HIV-specific probabilistic models of protein evolution", *PLoS ONE*, **2**, e503.

Nixon, K. C., 1999, "The parsimony ratchet : a new method for rapid parsimony analysis", *Cladistics*, **15**, 407–414.

Posada, D. and Crandall, K. A., 1998, "Modeltest: testing the model of DNA substitution", *Bioinformatics*, **14**, 817–818.

Redelings, B. D. and Suchard, M. A., 2005, "Joint Bayesian estimation of alignment and phylogeny", *Systematic Biology*, **54**, 401–418.

Ronquist, F. and Huelsenbeck, J. P., 2003, "MrBayes 3: Bayesian phylogenetic inference under mixed models", *Bioinformatics*, **19**, 1572–1574.

Ronquist, F., Huelsenbeck, J. P., and van der Mark, P., 2005, "MrBayes 3.1 Manual 5/26/2005", Distributed at http://mrbayes.csit.fsu.edu/manual.php.

Rota-Stabelli, O., Yang, Z., and Telford, M. J., 2009, "MtZoa: a general mitochondrial amino acid substitutions model for animal evolutionary studies", *Molecular Phylogenetics and Evolution*, **52**, No. 1, 268–272, Jul.

Saitou, N. and Nei, M., 1987, "The neighbor-joining method: a new method for reconstructing phylogenetics trees", *Molecular Biology and Evolution*, **4**, 406–425.

Schwarz, G., 1978, "Estimating the dimension of a model", *Annals of Statistics*, **6**, 461–464.

Shimodaira, H., 2002, “An approximately unbiased test of phylogenetic tree selection”, *Systematic Biology*, **51**, 492–508.

Shimodaira, H. and Hasegawa, M., 1999, “Multiple comparisons of log-likelihoods with applications to phylogenetic inference”, *Molecular Biology and Evolution*, **16**, 1114–1116.

Stamatakis, A., 2006, “RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models”, *Bioinformatics*, **22**, 2688–2690.

Sugiura, N., 1978, “Further analysis of the data by Akaike’s information criterion and the finite corrections”, *Communications in Statistics: Theory and Methods*, **A7**, 13–26.

Swofford, D. L., 2003, *PAUP*. Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4*, Sunderland, Massachusetts: Sinauer Associates.

Swofford, D. L. and Begle, D. P., 1993, *PAUP: Phylogenetic Analysis Using Parsimony , Ver.3.1. User’s Manual*: Laboratory of Molecular Systematics, Smithonian Institution.

Talavera, G. and Castresana, J., 2007, “Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments”, *Systematic Biology*, **56**, No. 4, 564–577, Aug.

Tanabe, A. S., 2007, “Kakusan: a computer program to automate the selection of a nucleotide substitution model and the configuration of a mixed model on multilocus data”, *Molecular Ecology Notes*, **7**, 962–964.

Tavaré, S., 1986, “Some probabilistic and statistical problems in the analysis of DNA sequences”, *Lectures on Mathematics in the Life Sciences*, **17**, 57–86.

Veerassamy, S., Smith, A., and Tillier, E. R. M., 2003, “A transition probability model for amino acid substitutions from blocks.”, *Journal of Computational Biology*, **10**, 997–1010.

Vos, R. A., 2003, “Accelerated likelihood surface exploration: the likelihood ratchet”, *Systematic Biology*, **52**, 368–373.

Whelan, S. and Goldman, N., 2001, “A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach”, *Molecular Biology and Evolution*, **18**, 691–699.

Woese, C. R., Achenbach, L., Rouviere, P., and Mandelco, L., 1991, “Archaeal phylogeny: reexamination of the phylogenetic position of Archaeoglobus fulgidus in light of certain composition-induced artifacts”, *Systematic and Applied Microbiology*, **14**, No. 4, 364–371.

Yang, Z., 1993, “Maximum likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites”, *Molecular Biology and Evolution*, **10**, 1396–1401.

——— , 1994, “Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods”, *Journal of Molecular Evolution*, **39**, 306–314.

——— , 1995, “A space-time process model for the evolution of DNA sequences”, *Genetics*, **139**, 993–1005.