生物系研究室における遺伝情報解析のための　UNIX環境の構築

--- MacOS XとLinuxを用いて ---

埼玉大学理学部
佐藤　直樹

はじめに

MacOS Xは，Darwinと呼ばれるBSD系のUNIXのシステムの上に，Aquaと呼ばれるMac風のGUI（graphical user interface，アイコンやウィンドウをマウスなどで操作する環境のこと）を載せたオペレーティングシステムであり，従来のMacOSとは全く異なっている。UNIXと呼ばれるオペレーティングシステムは，これまで，ワークステーションや大形コンピュータで使われてきており，情報科学や技術計算ではふつうに使われてきているものであるが，生物分野での利用はあまり一般的ではないようにも思われる。しかし，近年のIT革命により，いわゆるインターネットが整備されてくると，WEBブラウザを介して大形コンピュータの様々なサービスを利用することが可能になり，その一環として，遺伝子配列の簡単な処理はもとより，相同性検索や系統解析が容易に行えるようになり，間接的ではあるが，今やほとんどの生物系研究室でもUNIXマシンを利用していることになる。ところが，実際にUNIXマシンを導入している生物系研究室はまだ極めて稀である。相同性検索や系統解析も，従来のMacやWindowsマシンに移植されたプログラムを使って計算していることが多いようである。本稿では，生物系の研究室において比較的容易に導入できると思われる，MacOS Xを利用したUNIX環境の構築について，簡単に解説する。なお，このOS自体，発売されてから日が浅いため，必ずしも全てのことがうまくいくわけではなく，また，頻繁にアップグレードが行われる可能性があるため，記述の一部は近いうちに古くなるおそれもあることをお断りしておく。この原稿執筆時点では，バージョン10.1.2を利用している。

１．UNIX環境の有用性

いわゆるパソコンの利点としては，マウスを使ってメニューを選びボタンをクリックし，必要に応じて要求された場所に必要な情報を入力するという，対話形式で処理が可能であることである。このため，特定のソフトの使い方のトレーニングを特に受けていなくても，パソコンの一般的な操作に慣れたものであれば誰でも最低限の操作が可能であるということになる。しかし，良く考えてみるとこれは，人間が全く受動的にパソコンを利用しているということである。入試センター試験のようなもので，あらかじめ準備されたメニューなり選択肢から選ぶ作業しかしていないことになる。多くの場合，プログラムをエキスパートとして利用するためには，

しっかりとマニュアルを読み，的確なコマンドを選択し，オプションを正しく選ぶことが要求されており，操作の中身に精通していなければ目的通りの処理を行うことができないはずである。遠心機でもクロマト装置でも，詳しい説明を受けなければ正しい使い方ができないのと同じなのであるが，パソコンというだけで，詳しい使用説明も読まないで，ただクリックしている学生が多い。しかしパソコンのもっとも重大な欠点は，処理速度が遅いことと，システムクラッシュに弱いことである。クラッシュ（フリーズ）に泣かされた経験は誰でもあるはずで，パソコンの場合には，ある作業をしていてシステムが動かなくなると，装置を再起動しなければならない。これは，一見マルチタスク環境の備わったMacOS9やWindows98 (Me)などでもおこる問題である。Windows 2000 (NT) では，この問題は軽減されている。また，処理速度に関してはプロセッサやバスクロックの数字だけを信じている向きもあるが，同じマシンでもWindowsで動かす時と後述のPC-UNIXで動かす時とでは，処理速度が格段に異なる。単なる通常の計算であれば，２倍以上の速度の差があることも稀ではない。こうした，初心者でも使いやすいようにGUIを備えたパソコンOSは，いつでもユーザーからの様々な要求に応えられるように，様々な作業をユーザーの目につかないところで常時行い，また，そのためにメモリも大量に消費されている。最近のOSではメモリの消費量は30 MB以上が当たり前になっているが，ふだんの使用で本当に必要な部分はごくわずかである。自宅の娯楽用のパソコンで，テレビを見ながら動画の編集をし，CDを聞いたり3Dゲームをするというような利用法ならば，確かにこういうシステムも意味があるかも知れないし，研究室でも，3D分子グラフィックスを操作して分子モデルを作るとか，自前で動画からDVDやプレゼンテーション資料を作るというような場合には十分に有効であろう。しかし，研究室での遺伝子情報の様々な処理，特に，大量の相同性検索や大きな系統解析となると，こうしたGUIのためのOS機能はほとんど必要がなく，むしろ，パソコン資源の無駄遣いでしかない。こうした目的のためには，UNIXを導入するのが最適である。

UNIXの利点としては，以下のような点が挙げられる。（１）マルチユーザー・マルチタスクであるため，複数の処理を一つのマシン上で行うことができる。しかも，それぞれの処理が独立しているので，一つの処理がクラッシュしても他の処理には影響を与えない。（２）プログラムを自分で作って実行することができる。パソコンでも簡単なスクリプトを利用することはできるが，UNIXの場合には，大きな一連の処理を指示する命令群を，「スクリプト」として編集・実行できる。この場合，相当複雑な判断分岐を加えた処理ができるばかりでなく，同じスクリプトを異なった機種のマシンでもほとんど変更なしに利用できることが特徴である。また，いくつかのプログラミング言語を利用して自分でプログラムを作ることもできる。適当なソフトを買ってくれば，パソコン上でもプログラムを作ることはできるが，プログラム開発段階では頻繁につまらない原因でクラッシュが起きるうえに，パソコンの場合には，システムを壊す恐れも大きいというリスクがあり，あまり実際的ではない。（３）ネットワーク環境が整っており，学内LANやインターネットなどのネットワークに接続されたマシン間で，相互に乗り入れて処理ができる。Macや

Windowsでも，同じローカルネットワーク上にあるマシン間で，ファイルの共有・プログラム共有やプリンタ共有が可能になっているが，UNIXマシンの場合には，別のマシンにログインして，その中でプログラムを動かすことができる。MacやWindowsでもリモートマシン上のファイルをダブルクリックして開いたりプログラムを動かすことができるが，実行はこちらのマシンで行われている。逆に，机上にUNIXマシンが一台あれば，リモートのUNIXマシンを何台も同時にあやつることができる。現実に，私は，デスク上のノートパソコンにインストールしたPC-UNIXを使って，別の階にあるコンピュータ室においた数台のワークステーションや情報処理センターの大形コンピュータを操作している。（４）多くのPC-UNIXは基本的に無料である。売っているものもあるが，これは，FTPで公開されているものに，多少の有料ソフトを付加して，さらにサポートを加えたもので，大学の環境であれば，すべてをFTP経由で入手して無償で済ますこともできる。サポートについても様々なニュースグループやWEBサイトに情報がたくさんあって，たいていの問題は解決できる。

反面，PC-UNIXを導入する上での問題点もある。（１）ある程度の知識がないと使えない。これは上にも書いたことで，研究室の装置ならばなんでも同じことがあてはまるはずだが，パソコンとなるとどうも素人がいきなり使えないといけないような誤解がある。（２）インストールを自分でやる必要がある。初めての時には相当な覚悟が必要だが，いくつかやれば，留意点が分かってくるので，手間がかかることを別にすればあまり問題はない。しかも最近のPC-UNIXのインストーラーはとてもよくできており，比較的簡単に操作できる。（３）必ずしも日本語環境が整っているとは言い難い。これはおそらく普通の研究室ではたいした問題ではないと思うが，それでも時として不便を感ずることはある。

２．PC-UNIXの種類と選択

現在世の中にあるUNIX OSの種類はどのくらいあるのか見当もつかないが，導入や操作が比較的簡単で，しかも周辺機器もある程度利用できることを考えると，やはりLinuxが一番ふつうに使えるOSである。Linuxの導入に関しては，大きな問題はないと思われるが，Windowsとのdual boot（２つのOSを両方とも一台のコンピュータにインストールし，起動時に選択するもの）にして使う場合には，同時に両方のOSを使うことはできないため，いちいち再起動するのに時間がかかり，あまり便利ではない。ふつうに研究室の中で使う場合には，Linux専用マシンとWindowsマシンをLANで接続してデータの交換を行いながら利用するのがよいと思う。別の解決としては，VMwareを利用するという方法もある。これは，Linux上でWindowsを仮想的に動かしたり，その逆をするというものであるが，あまり安いとはいえないのが難点である。

これに対して，２００１年３月に発売されたMac OS Xは，一台のマシンの上で，

基本はBSD系のDarwinというOSが動いていながら，その上で，従来のMac風のデスクトップと操作性が実現している。これまでUNIXとMacやWindowsを接続してデータ交換をする場合の問題点としては，ハードディスクのフォーマットが異なるために相互に簡単にマウントできない（特別のソフトを必要とする）こと，テキストファイルの行末コードが異なること，漢字コードが異なることなどであった。Mac OS Xの場合には，通常のMac OS拡張フォーマットのハードディスクを使っていて，Mac OS 9からもファイル共有ができる。また，テキストファイルの行末コードも自動変換され，ユーザーは特に気にする必要がない。一方で，Terminalと呼ばれるコマンド入力画面で，コマンドによるUNIX環境での操作をしながら，まだあまり数は多くないが，Mac環境のソフトも同時に利用できる。Sambaというソフトが入っているため，Windowsとのファイル共有も容易である。こうした観点から，研究室で使うパソコンとしては，もっとも便利な操作性を持ったものといえると思う。従来のMacで使えたショートカットキーの多くが通用するので，この点もMacに慣れたユーザーには非常に使いやすい。以下の解説では，MacOS Xでの環境構築について一通り述べるが，始めの方のMac特有の部分を除けば，Linuxにも共通している。

３．MacOS Xのインストール

MacOS Xは，G3または G4プロセッサを搭載したパワーMacなら，たいていはインストール可能であるとされている。しかし，研究室での利用を考えると，iMacなどよりは，PowerMac G4の方が，拡張性が高く，また，CPUの速度の速いものが利用できる。メモリも1 GB以上搭載可能である。MacOSXではGUIのために非常に多くのメモリが使われるので，メモリはできるだけ多く搭載するのが良い。私自身が使っているのは，500 MHz と700 MHzのPowerMac G4であるが，どちらも1 GBのメモリと60 GBのHDを２台内蔵している。高速のethernet接続端子がはじめからあるため，この点も研究室で使うのには最適である。プリンタとしては，MacOS Xでは，原則としてpostscriptプリンタしか使えない。一部のプリンタについてはドライバが供給されているものもある。ふつうのプリンタが使えないのが残念であるが，この点はいずれ改善されることを期待したい。

MacOS Xは，Apple Storeまたはパソコンショップで入手できる。インストールは，基本的にはマニュアルやインストーラの指示通りにやれば良く，ここでは詳しく述べない。私自身は，一台目のHDを２つのパーティションに分割し，第一パーティションにMacOS9.1を，第二パーティションにMacOS Xをインストールしている。OS自体のインストールに加えてさらに，別のインストールCDから，開発者用環境のインストールを行い，コンパイラを利用できるようにする。また，将来インストールする様々なプログラムのために，管理者のパスワードを登録する必要がある。これは絶対的に重要なことであるので，一番最初のこの段階で必ず済ませておくこと。

一番簡単な方法は，インストール用CDから立ち上げて，最初の段階で，プルダウンメニューからルートパスワードの設定を選ぶというやり方である。これを行うと，以後，システムを書き換える際には，ユーザー名rootでログインし（あるいは，通常のユーザーからsuコマンドでrootに変更し），このパスワードを入力することになる。すべてをインストールした上で，ネットワーク環境の設定をし，Internet Explorer 5が使えることを確認する。この先は全てこのIE5を用いて必要なファイルをftpにより入手する。なお，いろいろな本が出ているので，それらを参考にしながら，さまざまな設定をする。特に，NetInfo Managerはネットワークを利用する上で重要な外部ホストの設定などをまとめて管理するもので，極めて重要だが，設定を誤ると重大な障害が発生する危険があり，注意が必要である。外部から接続して利用できるようにするには，ネットワークの他，System Preferencesの中の共有を設定する。

４．Macの裏で動く UNIX環境の体験

MacOS Xを使うと，従来のMac風のウィンドウを使った環境が利用できるようになる。この環境では，多くのプロセスがバックグラウンドで動いていて，ユーザーの仕事のために使えるのは，CPUの能力の80%程度である。これを95%程度まで上げるには，Mac環境を止めて，その下で動いているUNIX環境を使う必要がある。このためには，ログインウィンドウでユーザー名として，`'>console'` と入力する。なお，MacOSX version 10.1からは，デフォルトのログイン画面がユーザーのアイコンになっているが，これは，システム環境設定のログインを書き換えることにより，一般のUNIXと同様のユーザー名を打ち込むタイプのログイン画面に変えることができる。`'>console'` と入力すると，直ちに真っ黒な画面にかわり，UNIX風のログインプロンプトが出現する。驚かずに，ユーザー名とパスワードを入力してログインする。ここで，`'ls'` というコ マンドを入力すると，現在のディレクトリ（この場合，ユーザー someone のホームディレクトリである/Users/someone ）の中にあるファイルとディレクトリの名前が表示される。`'ls -al'` を使えば，ファイルの日付やサイズも表示される。基本的にはMacのウィンドウの表示で，リストを選んだ時のような情報が表示される。この状態では，アイコンなどの表示はなく，マウスも利かない。すべて，コマンドの入力だけで作業する。これでも，遺伝子解析用のソフトなどのインストールは可能であるが，画面のスクロールができないなど，やはりウィンドウが開かないとやりにくい面もあるので，まず，X環境をインストールすることにしたい。なお，元の画面に戻るには，`'logout'`コマンドを実行する。

５．X環境と基本的ソフトのインストール

(1) Xウィンドウシステム

これは，UNIX上でウィンドウを開くためのシステムで，MacやWindowsで使われているのと類似のウィンドウが使える。UNIXでは多くのフリーソフトがあり，それら

の多くはXウィンドウを利用して操作できるようになっている。また，ブラウザとして，UNIXではNetscapeが使われているが，Darwin上で動くXウィンドウで使えるブラウザは今のところない。ブラウザを使う場合には，MacOS Xの環境でIE 5かNetscape（プレリリース版）を使うしかない。Xウィンドウとして利用できるものにも何種類かあるが，フリーのソフトでは，XFree86のものが利用でき，OSX GNUからMacOSX対応版が公開されている（末尾のURL集を参照）。バージョン1.0.4が現在の最新版で，MacOS X 10.1に対応しており，さらに，Quartzアプリケーション(Xquartz.tgzが必要)としても開くことが出来て，「ルートレス」というモードを使うと，MacOSXの画面の上でウィンドウを切り替えながら作業することが可能になった。詳細はダウンロードサイトにあるドキュメントをみていただきたいが，慣れていないと多少とまどうかもしれない。ダウンロードのあと，インストールにはTerminalを開き，ファイルのあるディレクトリに移動してから(‘cd <directory>’ コマンドを使う)，’sh Xinstall.sh’というようにコマンドを入力する。いくつかの質問に答えるのだが，基本的にはデフォルトでよく，リターンキーを押していくだけである。インストールのあと，アプリケーションフォルダにある大きな赤い字で’X’と書かれたXDarwinのアイコンをダブルクリックする（図１）。私は，このアイコンをドックにドラッグして，いつでも使えるようにしている。上のメニューバーからXDarwin --> 環境設定を選び，いくつかの設定を行う。以前のバージョンでは，ここで日本語キーボードを選択していたが，今のバージョンにはこれはない。現状ではXDarwinで日本語の入力・表示ができるわけではないので，注意する必要がある。また，このままでは，正しく入力できないキーが生じることがあるので，Xを実際に動かす前に，’.xmodmaprc’ファイルを準備する必要がある。これは，私のHPからダウンロードできる。ただしこれもXDarwinのバージョンによって状況が変化しているので，最終的には，ユーザーの手で微調整する必要があるかもしれない。UNIX Xウィンドウシステムの解説書を見ながら，keycodeまたはkeysymを設定していく。

Xを起動する。フルスクリーン（全画面表示）を選ぶと，これまでのMacの画面とは全く異なるUNIXの画面になる。仮想画面ができていて，本来の画面サイズより大きな面積を利用できる。この状態では，twmというウィンドウマネージャーが動いているが，後述するLessTifを導入すれば，もう少し使いやすいmwmを使うことができる。また，細かい設定も自由に変えられるのだが，それは，X ウィンドウシステムの解説書を参考にしていただきたい。なお，Xの起動の際にうまくいかないことが時々あるが，もう一度やり直せば問題なく動くことが多く，この理由はよくわからない。このXウィンドウを使うと， GUIを使った様々なUNIXプログラムが動かせる。もとのMac画面に戻るには，コマンドキーとオプションキーとAを同時に押す。両画面をこのようにして往き来できる。

Xの起動時に「ルートレス」を選ぶと，twmのウィンドウがMacOSXの画面に重な

ってあらわれ，切り替えも簡単にできる。しかも仮想画面が使える。マウスカーソルが画面の端にくると，隣の画面に入るので驚くことが多いが，この場合，左上のPagerをクリックすることで，仮想画面間を移動する。

もう一つのXの起動の仕方は，上に述べた'>console'からログインし，その後で，'startx'というコマンドを実行する。これでもXウィンドウが動くが，この場合は，Macの様々なソフトは動いていないので，マシンの能力を最大限に発揮させたい場合はこのやり方がよい。ふつうのログイン画面に戻るには，すべてのウィンドウから'exit'で抜け，何も表示されない画面で，ポインターだけがくるくる回っている状態になったら，'logout'とタイプする。この場合何も画面表示が出ないので，注意して入力する。X関係のファイルはすべて，/usr/X11R6および/Applications/XDarwinに入っているので，失敗したりバージョンアップする場合には，これら２つのディレクトリだけ消去するか，名称を変更して退避すればよい。なお，/usrなどのUNIXディレクトリは，MacOSXのFinderからは見えないように設定されているが，Terminalからは操作でき，また，Finderでの不可視ファイルの設定も変更は可能である。

この段階で，一般的なLinuxのインストール時とほぼ同じ状態になっている。Linuxを使った遺伝情報解析環境の構築の場合には，ここから始めればよい。なお既に，LessTifまたはopenmotifがインストールされている場合には，(3)から始める。当面の目標として，系統解析でアラインメントを作る際に必要な，Clustal Xを導入するのだが，そのためには，まず，Motif (LessTif)と NCBI toolbox (blast検索ソフトなどを含む)をインストールする必要がある。

(2) LessTifの導入

フリーのMotifには，openmotifとlesstifがあるが，openmotifは現在，昨年に発表されたバージョン2.1.30だけしかないので，MacOSXではうまくコンパイルすることができなかった。また，lesstifでもバージョンにより，コンパイルできないことがあった。少し前の時点では，バージョン0.92.6はコンパイルでき， 0.92.26はコンパイルできなかった。現在の最新版である0.93.18は問題なくDarwin OSを認識し，コンパイルもできた。

Terminalを開いて，取得したtar.gzで終わる圧縮ファイル（圧縮アーカイブという）をおいたディレクトリに移動し，まず，gzipコマンド(gzip 「スペース」 -d 「スペース」「圧縮アーカイブ名」)，次に，tarコマンド(tar「スペース」xvf 「スペース」「生成したtarファイル名」)を使ってアーカイブを展開すると，ソースファイル含むディレクトリが生成される。UNIXのコマンドは，すべて，「コマンド」「オプション」「ファイル名」のような形で入力し，個々の項目の間は，「スペース」で区切ることになっている。この「スペース」は，日本語ではなく，ローマ字のスペースである。以後は，特に「スペース」を明示することはしないが，ローマ字のスペースを入力することに注意していただきたい。また，本稿では，入力コマンドはすべて等幅フォントで

あるCourierで表示し，基本的に引用符’ ‘で囲むことにしておく。基本的なコマンドは，末尾のコマンド一覧に示した。上のようにgzipとtarを使って二段階でファイルを展開する代わりに， MacOS Xに入っているtarでは，これを一段階で済ませることができるので，'tar zxvf 「圧縮ファイル名」'としても良い。UNIXでは，同名のコマンドでも，システムによって働きが微妙に異なるので，このやり方が常に使えるというわけではない。次にこの新しいディレクトリに移動し(‘cd 「ディレクトリ名」'), './configure', 'make', 'make install'と，順次コマンドを実行していけば，必要なファイルが通常/usr/local/LessTif以下のディレクトリにインストールされる。この作業は全部で１時間程度かかり，また，200 Mb近くの多量のHD領域を必要とする。まずconfigureであるが，このconfigureスクリプトがシステムを認識してくれることになっている。少し前のバージョンではDarwinを認識できないため，config.subとconfig.guessの２つのファイルを変更する必要がある。それには，/usr/libexec/ディレクトリにある同じ名前のファイルを持ってきて置き換える。すなわち，'cp /usr/libexec/config.sub .' などとする。cpの後と，最後のピリオドの前にはスペースがある。なお，このやり方は多くのGNUソフトウェアについても当てはまる。さて，'./configure'コマンドであるが，このはじめの「ピリオド　スラッシュ」は，現在のディレクトリを示す。これがないと，現在のディレクトリにちゃんと見えているのに，configureというスクリプトを実行することができないということで，慣れていないユーザーは悩むことになる。またこの際のオプションとして，スタティックなライブラリも必要であれば，'–-enable-static'も加える。Motif2.0または1.2互換ライブラリも作る場合には，'–enable-build-20' または'–enable-build-12'をさらに加える。オプションの一覧は，'./configure –help'で表示されるので，最初に確認しておくと良い。一度configureすると，'make clean'だけでは初期状態に戻せない。完全に初期状態に戻してやり直すには，'make distclean'とタイプする。

次に'make'を行い，エラーなしで終了すれば，'make install'を行う。これにより，/usr/local/LessTifディレクトリ内に様々なファイルがコピーされ，さらに，/usr/X11R6にもライブラリなどが置かれる。無事に作業が終われば，元のソースファイルのディレクトリは消しても構わない。さらに，ライブラリとヘッダーファイルを利用しやすくするため，それぞれ，/usr/X11R6/libディレクトリと/usr/X11R6/includeにリンクを張っておく方がよい。これには，'su' コマンドを入力した後，パスワードを入力することにより管理者またはrootになり，

```
‘ln –s /usr/local/LessTif/Motif2.0/lib/libXm.a /usr/X11R6/lib’
‘ln –s /usr/local/LessTif/Motif2.0/include/Xm /usr/X11R6/include/’
```

などとする。リンクは必須ではないが，後で他のソフトのコンパイルをする場合に，設定ファイルに正しく書き込む必要がある。rootをやめるときは，‘exit'と入力して，元のユーザーに戻る。

(3) NCBI toolboxのインストール

次にNCBI toolboxをインストールする。GenomeNet またはNCBIのFTPサイトからncbi.tar.gzをダウンロードする。これはいち早くMacOS Xに対応しており，全く問題なく作業できる。/usr/local/以下に上述のようにソースコードを展開することにする。これはルートになって行う必要がある。あるいは，ユーザーsomeoneのホームディレクトリ/Users/someoneの中で，作業をしても良い。ただ，複数のユーザーが利用できるようにするという意味では，/usr/localを使う方がよい。そこでncbiというディレクトリができたら，いくつかのファイルを修正する。ncbiディレクトリの下のmakeディレクトリを見ると，makeall.unxというファイルがある。これをTextEditなどで書き換える。VIBLIBS= およびVIBFLAGS= という行を探し，それぞれ，以下のようにする。

```
  VIBLIBS= -L/usr/X11R6/lib –lXm –lXpm –lXmu –lXp –lXt –lX11 –lXext
  VIBFLAGS= -I/usr/X11R6/include –I/usr/local/LessTif/Motif2.0/include -
DWIN_MOTIF
```

この２つの項目はそれぞれ一行で入力する。なお，これらの定数の定義は，同じファイルの下の行に再度別の定義があると書き換えられてしまうので，再定義されていないか念のため確認する。

次に，ncbiディレクトリの下のplatformディレクトリに入り，darwin.ncbi.mkというファイルを書き換える。このファイルが見つからない場合は, 古いバージョンのNCBI toolboxであると思われるので，再度最新版を入手する。各種定数の定義が書かれているが，その中で，行頭が#で始まっている行はコメントになっていて機能しない。本当のコメント行は別として，'NCBI云々='という行については，すべて#を取り去り，正しく設定する。すなわち，

```
  NCBI_BIN_MASTER = /usr/local/ncbi/bin
  NCBI_BIN_COPY = /usr/local/ncbi/bin
  NCBI_INCDIR = /usr/local/ncbi/include
  NCBI_LIBDIR = /usr/local/ncbi/lib
  NCBI_ALTLIB = /usr/local/ncbi/altlib
```

次にncbiディレクトリの上に出て，ncbiディレクトリが存在するディレクトリで，以下のコマンドを実行する。

```
  ‘ncbi/make/makedis.csh |& tee out.makedis.csh’
```

これは，ncbiディレクトリの下にあるmakeディレクトリの中にあるmakedis.cshというCシェルのスクリプトを実行し，その結果出てくるいろいろな画面表示をout.makedis.cshというファイルにも保存する，ということである。わかりにくいかもしれないが，'|&'というのをひとまとめにして，その前後はスペースがある。また，'tee'の後もスペースである。これにより，自動的にコンパイルが行われる。これも１～２時間の作業である。その後，ncbiの中のbuildというディレクトリを見ると，たくさんのファイルが作られている。'ls blastall', ‘ls formatdb'などというコマンドにより，それぞれの実行ファイルの存在を確認する。もしもこれらが作られなかった場合は，何らかのエラーがあるので，もう一度設定ファイルの記述な

どを点検する。再コンパイルの際には, buildディレクトリの中身はすべて消し（'rm –f *'をbuildディレクトリの中で実行する），改めて，ncbiディレクトリの上に出て，上述のcshスクリプトを実行する。

コンパイルに成功した場合には, ソースコードのコピーはいらないので, 'rm –f *.c *.h make*'と入力して消去する。-fおよび*.hの前後にはスペースがある。ここで，*は任意長の文字列の代理として機能する。残るのは，実行ファイルとライブラリである。ライブラリもncbi/libにコピーされているので，消して構わない。そのためには，'rm –f *.a'とする。-fの前後にはスペースがある。実行ファイルの中で，blastallなどBLASTサーチのコマンドとformatdbというデータベース処理ソフトがあれば，相同性検索ができる。これらを'mv blast* formatdb ../bin'として，ncbi/binディレクトリに移しておく（ここで，２個のピリオドは，一つ上の階層のディレクトリを指している）。実際にこれらのコマンドを利用できるようにするためには，ncbi/binディレクトリにパスを通すか，または，/usr/local/binディレクトリにリンクをつくる。ncbiのあるディレクトリを/usr/localとすると，「パスを通す」ためには，ホームディレクトリにある.tcshrcというファイル（なければテキストエディタで作る）に，'set path = ($path /usr/local/ncbi/bin)' という一行を加える。「リンクを作る」ためには，'ln –s /usr/local/ncbi/bin/blastall /usr/local/bin/blastall'などとする。または，そのまま/usr/local/binにコピーしても良い。つまり，'cp /usr/local/ncbi/bin/blastall /usr/local/bin'。ライブラリの存在場所は，この場合，/usr/local/ncbi/libとなる。環境変数NCBIを正しく設定することが重要で，これは，パッケージの中のドキュメントファイルに書かれているので，確認していただきたい。私は，'.ncbirc'ファイル（私のホームページにある）を/usr/local/ncbiに置いていて，環境変数NCBIがこれを指すようにしている。すなわち，ホームディレクトリの'.tcshrc'ファイル（上述）の中に，

```
'setenv NCBI /usr/local/ncbi'
```

という一行を加えておく。これを正しく設定していないと，プログラムが全く利用できない。

(4) Clustal Xのインストール

つぎにClustal Xのインストールを行う。あらかじめ，適当な名前で作ったディレクトリ(/usr/local/clustalとする)の中で，ソースファイルを展開する。取得するソースファイルはLinux用のものなどどれでも良い。入っているソースファイルに違いはない。実際には，ダウンロードしたファイルを展開すると，clustalx1.81のような名前のディレクトリがつくられるので，適当に名前を読み替えていただきたい。Terminalを開いてclustalディレクトリに入り，makefileを'vi'コマンドにより修正する。上のLessTif, ncbiのインストール場所により内容が少し変わることもあるはずだが，以下のようにする。

```
CC = cc     （コンパイラがMacOS Xではcc）
```

```
  NCBI_INC = /usr/local/ncbi/include
  NCBI_LIB = /usr/local/ncbi/lib
  LXFLAGS = -L$(NCBI_LIB) –lvibrant –lncbi -L/usr/X11R6/lib –lXm –lXmu –lXt
–lX11 -lSM –lICE
```

（この行はかなり複雑に書き換えるので注意）

ここで'-l'はライブラリをリンクすることを表し，/usr/X11R6/libディレクトリにあるlibXm.aというライブラリをリンクする場合に，'-lXm'と表記することになっている。また，'-L'はリンクするライブラリのあるディレクトリの指定である。MacOS Xの場合，数学ライブラリである'-lm'は元々Cライブラリに含まれているらしく，特に加える必要はないようである。その上で，makeとすればコンパイルが行われる。もともと存在していたclustalxなどのコマンドは，他のシステム用の実行ファイルなので，MacOSXでは使えない。'file clustalx'などとすると，どのシステム用の実行ファイルであるかが表示される。コンパイルが成功したら，できたclustalwとclustalxを/usr/local/binにコピーする。もしも失敗したときは，makefileの内容をもう一度確認する。

ここまでの作業に関しては，基本的にルートとして作業することが必要になるが，コンパイルはユーザーのディレクトリで行い，できたファイルだけを/usr/local/binなどにコピーするという場合には，最後のシステムへのコピーを除き，ユーザーとしてコンパイル作業をすることが可能である。詳細は，UNIXの入門書を見ていただきたい。また，パソコンと違い，インストールした後でマシンを再起動する必要はない。ただ，コマンドの検索パスをリフレッシュするため，'rehash'というコマンドを実行するか，一度シェルからexitして，再度シェルに入り直す必要がある。Xウィンドウで，clustalxと入力してみよう。パソコン上で見慣れたClustal Xのウィンドウ（に似たもの）が出現すれば成功である。だめなときはインストール作業をもう一度点検する。なお，MacOS XのTerminalからでは，グラフィックスシステムが異なるため，実行できない。

さらに，系統樹を表示するため，njplotというツールがある。これもclustalxとほとんど同様にmakefileファイルを修正すれば，コンパイルできる。インストールも同様である。Linuxなどで実行してみると，メッセージ画面で少し文字化けすることがあり，これは，もともとフランスで作られたソフトであることが原因であるらしいが，XDarwinでは問題ないようである。

６．その他の一般的 UNIXツールのインストール

基本的なツールとして，Perl, Tcl/Tkなどがあるが，MacOS Xの場合，Perlは最初からインストールされているので，ごく普通に使うことができる。Tclも入っているのだが，TkのウィンドウであるwishがMacのGUIと互換性がないため，MacOS Xにはwishが入っていない。Tkも上述のX環境では問題なく使える。なお，Aquaの上で動くTk

も作られているが，まだ試作段階のようである。Tcl/Tkを利用するには，それぞれのソースコードをダウンロードし，指示に従ってコンパイルすればよい。その際，まずTclをコンパイルし，その状態のままで，Tkのソースコードを展開して，コンパイルを行う。これは，TkがTclを利用するためで，全部ができあがったら，それぞれを/usr/local/などにインストールする。

７．様々な遺伝子解析ソフトのインストール

さらに私たちの研究室では，遺伝子解析ソフトパッケージとしては，SISEQ, Phylip, fastDNAml, Molphy，FASTA, HMMなどを使っているが，これらは，コマンドラインプログラムなので，Xウィンドウがなくても特に大きな問題はなく，MacOS XのTerminalからも利用できる。ほとんどのものは，ダウンロードしたパッケージを，makeコマンドでコンパイルし，できたバイナリを適当なところ(/usr/local/bin)に置くか，または，バイナリのあるところにパスを通すことで，利用可能になる。SISEQは，私が作った遺伝子データベースのファイル変換用ソフトで，MacやWindowsでも利用できるように作ってあるが，基本的にはUNIXで利用する。また，GUIも用意してあるので，Tkをインストールしてあれば，必要事項を埋めていくだけで利用できる。遺伝子データベースの管理用ソフトとしてSRSがある（URL集参照）。これを用いると，研究室レベルでデータベースのキーワード検索とダウンロードサイトが作れるため便利なこともあるが，反面，すぐにデータが古くなり，メンテナンスが大変なので，覚悟してかかる必要がある。また，日本のGenome NetではKEGGがあり，これもシステム全体をダウンロードできるが，自前で維持していくのはかなり大変で，解析システムとはこういうものかということを勉強するのには良いかもしれない。

８．遺伝子データベースの取得と解析環境の整備

遺伝子データベースとしては，DNAではGenBank, EMBLが，タンパク質ではSwissProt, PIRなどがある。ただ，GenBankとEMBLはほとんど重なっていて，しかも最近ではほとんどフォーマットが同じなので，両方をそろえておく必要はない。また，SwissProtとPIRもかなり重複するが，たいして大きなデータベースではないので，両方持っていても良いだろう。というわけで，私は，GenBankのフルセット，SwissProt, PIR, Pfam, Prodomを手元に持って使っている。これらは，元ファイルをSRSに組み込んで，キーワード検索ができる形にしてあるが，時間と手間を考えなければ，SISEQを用いてデータベースファイルを直接読み，検索処理することも可能で，これならばバッチ処理が可能になる。なお，SRSなどでもバッチ処理のコマンドがある。通常，研究室でデータベースを維持する利点は，多数の配列の検索を一度にやってしまうことができることと，自前のデータベースの利用である。一般の検索サイトでは，一度に一個の配列の相同性検索しかできない。しかも，ずっとパソコンの前にいて，カット・ペーストやマウスクリックを繰り返す必要がある。最近で

は，研究室で生産された新規の大量の配列の相同性検索をまとめてやりたいことが多い。また，既に公開されているデータベースから自分の目的にあった配列を選び出し，それを元に相同性検索を行うこともある。さらに，ゲノム全体のコード領域配列すべてを別のゲノム配列全体との間で相同性検索したいこともある。こうした大量検索は，自前のサイトでコマンドラインから行う。その場合，一番普通に検索をするための元になるデータベースファイルとしては，GenBankのnt, nr (non-redundant nucleotide, protein)，またタンパク配列としてswissprot，すべてのORFを集めたgenpeptなどを使っている。その他，ESTとして，ヒトならest_human，植物などはest_othersなどを使う。これらはすべて，NCBIのダウンロードサイトから取得できるが，日本国内では，GenomeNetのFTPサイトを利用すればよい。平日は回線が混んでいるので，土曜日か日曜日にダウンロードするとよい。複数のファイルをまとめてダウンロードできる'mget'というftpのコマンドを使えば，操作は楽になる。すべてのファイルは，Zまたはgzという拡張子がついた圧縮ファイルである。バイナリモードでダウンロード後，前者は'uncompress'，後者は'gzip –d 'というコマンドを使って展開する。数倍の大きさのファイルができあがるので，あらかじめ十分にHDの容量があることを確かめてから展開を始める必要がある。ファイルを取得後，NCBI toolboxをインストールしてあれば，その中の'formatdb'というコマンドを使って，Blast検索用のインデックス作製を行う。使い方は，このコマンドを引数なしでタイプすると表示される。必ず入力ファイル名を，-i オプションの後に記述する。DNA配列の場合，'-pF'というオプションが必要である。うまくできると，入力ファイル名の後に拡張子のついた新しいファイルが３個できる。また処理状況を記したformatdb.logというファイルができる。普通のファイルなら問題ないが，自前の配列ファイルやゲノムデータのプレリリース版の場合，正確に決まらない配列部分にハイフンなどの特殊記号が入っていて，うまく処理できないことがある。この場合は，問題の文字を，DNAならNに，アミノ酸ならXなどに変えてから処理する。

以上の準備ができれば，いよいよBLAST検索開始である。検索の前に，用いるデータベースの存在するディレクトリ名を環境変数BLASTDBにセットする。このためには，多くのLinuxやMacOS Xで使われているCシェルやTCシェルならば，'setenv BLASTDB /usr/local/db'などとする。普通の検索では，blastallというコマンドを用いる。オプション等を説明したヘルプ情報は，引数なしでこのコマンドを実行すると表示される。必ず確認したい。最低でもオプションを４個加える。-i の後に「入力ファイル」，-d の後に「データベース名」，-p の後に「プログラム名」，-o の後に「出力ファイル名」を記述する。このうち，プログラム名については，上述の様にヘルプを参照のこと。データベース名としては，上述のformatdbの際に –i の後に書いたものと同じものを使う。その他のオプションとしては，出力のカットオフ値，出力配列数その他多数ある。実行例としては，

```
blastall –i test.seq –d nr –p blastp –o test.results
```

などと一行で入力する。この場合，test.seqはアミノ酸配列を含むFASTA形式のファ

イルでなければならない。出力ファイル名は任意であるが，存在するファイル名を使うと上書きされる。また，UNIXの場合，書き込み許可のあるディレクトリにファイルを作らせるようにする必要がある。

９．現状の問題点

MacOS Xを用いた遺伝情報解析システムの問題点を，簡単にまとめておく。まず日本語が使えないことが大きな問題である。学術研究では日本語は必要でないとはいうものの，ファイルを交換したりする際には，結構重要なことである。ファイル名は，MacOS XのFinderであれば日本語でもよいが，Terminalからファイル処理をする場合にはうまくいかない。そもそもTerminalからは日本語が全く利用できない。'vi'などのコマンドでファイルを編集する場合にも，日本語はだめである。日本語を含むファイルの編集は，TextEditを使うしかないが，ファイルの出力形式がrtfだけであることが問題である。これはいずれ改善されることと思うが，当面の方策として，/Developerにインストールされているテストプログラムに含まれるSimple Textを使うと，テキスト文書の編集ができる。

Linux等と比べた場合，MacOS Xのみでうまく使えない機能がいくつかある。この理由としては，DarwinというOS自体はBSD系のUNIXなのだが，MacOS X特有のさまざまな改変がされていることが挙げられる。たとえば，GNUのフリーソフトのうちでもgccなど簡単にコンパイルできないものがある。また，ネットワーク関連の設定も，ふつうのUNIXのように/etcにある設定ファイルを使うのではなく，NetInfo Magagerを使うなど，かなり特殊な部分がある。だんだんMacOS Xの解説書も増えているので，それらを参考にしてやっていくほかない。

それでもMacとの透過的な利用は非常に便利で，DarwinでつくったファイルをそのままAppleshareで他のMacに移して使うこともできるし，同じマシン上で，XとMacを切り替えながら使うのも便利である。推奨されるG4を使っている限りは，スピードも特に問題なく，リモートからログインして，BLAST検索などをやってみると，かなり速いことが実感できる。同じクロック数のWindowsパソコンを使ったLinuxよりもだいぶ速いように思うが，メモリその他条件が同じでないので，正確な比較は難しい。今後は，MacOS Xをきっかけにして，UNIXベースで仕事をするユーザーが増えてくることは確実である。既に外国ではかなりブームになっているようであるが，日本の分子生物学関係の研究室でもこれを機に，MacOS Xで遺伝子解析をやっていくことが流行りになること疑いなしである。

参考文献

1. Thompson, J.D., Higgins, D.G. and Gibson, T.J.: Nucl. Acids Res. **22**, 4673-4680 (1994) – Clustal W**の論文**。
2. Felsenstein, J.: Annu. Rev. Genet. **22**, 521-565 (1988) – **系統解析の解説,** Phylip**パッケージもこれによっている**。
3. Sato, N.: Bioinformatics **16**, 180-181 (2000) – **著者によるデータベース処理ソフト**。
4. Baxevanis, A. D.: Nucleic Acids Res. 29, 1-10 (2001) – **データベースの紹介**。
5. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J.: Nucleic Acids Res. 25, 3389-3482 (1997) -- Gapped BLAST**について書かれている**。
6. Zhang, Z., Schäffer, A. A., Miller, W., Madden, T. L., Lipman, D. J., Koonin, E. V. and Altschul, S. F.: Nucleic Acids Res. 26, 3986-3990 (1998) – PSI BLAST**について書かれている**。

関連URL一覧

1. Apple　　http://www.apple.co.jp/macosx/
2. MacOS X GNU　　http://www.osxgnu.org/
3. Linux　　http://www.linux.or.jp/
4. XFree86　　http://www.xfree86.org/
5. Sato　　http://www.molbiol.saitama-u.ac.jp/~naoki/
6. LessTif　　http://www.lesstif.org/
7. NCBI toolbox
   ftp://ftp.genome.ad.jp/pub/db/ncbi/toolbox/ncbi_tools/ncbi.tar.gz
8. Clustal X　　ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalX/
9. Njplot　　ftp://pbil.univ-lyon1.fr/software/njplot.html
10. Phylip　　http://evolution.genetics.washington.edu/phylip/getme.html
11. FASTA　　ftp://ftp.virginia.edu/pub/fasta/
12. List of free software in molecular biology
    http://iubio.bio.indiana.edu/soft/molbio/Listings.html
13. Molecular Linux　　http://bioinformatics.org/mol_linux/
14. GenBank　　ftp://ftp.genome.ad.jp/pub/db/ncbi/genbank/および
    ftp://ftp.genome.ad.jp/pub/db/ncbi/blast/db/ (nr, nt, swissprotなど検索用ファイル)
15. EMBL　　ftp://ftp.genome.ad.jp/pub/db/ebi/embl/release/
16. SwissProt　　ftp://ftp.genome.ad.jp/pub/db/ebi/swissprot/release/
17. Pfam　　ftp://ftp.genome.ad.jp/pub/db/ebi/Pfam/
18. SRS　　http://www.lionbio.co.uk/
19. Mozilla　　http://www.mozilla.org/

設定ファイル一覧

ファイル名がピリオドで始まるファイルは，そのままではlsコマンドで表示されないので，ピリオドをとった名前に変えて保存してある。

私のHPにおいてある。http://www.molbiol.saitama-u.ac.jp/~naoki/macosx/

1. .xmodmaprc (Xfree86ホームディレクトリ用)
2. .tcshrc (ホームディレクトリ用)
3. darwin.ncbi.mk (/usr/local/ncbi/platform/)
4. makeall.unx (/usr/local/ncbi/make/)
5. .ncbirc (/usr/local/ncbi/)
6. makefile (ClustalX)

基本コマンド集

ls	現在のディレクトリにあるファイル名を表示する。
ls -al	同上だが，日付やサイズも含めて表示する。
cd	ディレクトリを移動する。行き先ディレクトリ名を書かないと，ホームディレクトリに移る。
pwd	現在のディレクトリを表示する。時々わからなくなることがあるため。
cp	ファイルのコピー。ファイル名とコピー先ディレクトリ名を書く。
rm	ファイルを削除する。パソコンと違い，一度消したら元には戻せない。
mv	ファイルの移動。または，ファイル名の変更。昔のDOSのrenameはUNIXにはない。
gzip -d	gzでファイル名が終わる圧縮ファイルの展開。
tar xvf	tarでファイル名が終わるアーカイブの展開。
make	プログラムパッケージでコンパイルするときに使う。
configure	プログラムパッケージで，システムの確認などをするために使うコマンド。すべてのパッケージについているわけではないので，それぞれのREADMEまたはINSTALLなどのファイルを読んでやり方を確認する。

パスの説明

.	現在のディレクトリ。現在のディレクトリにあるコマンド（commandというなまえとする）を実行する際には，'./command'とする。DOSと違い，カレントディレクトリにはパスが通っていない。
..	一つ上の階層のディレクトリ。隣のディレクトリtonariに移るためには，'cd ../tonari'とタイプする。