Bioinformatics 用ソフトウエアのインストール

1　はじめに

３回生前期までの実習では、 BLAST 等のソフトウエアを、主にインターネット上に存在するサーバを利用して実行してきた。このやり方は、少ないデータを扱うには十分だが、ゲノムのアノテーションを行う場合など、大量のデータを処理するには不向きである。大量のデータを処理するには、類似の作業を繰り返して実行させるバッチ処理が非常に有効である。しかし、バッチ処理を実行するには、ソフトウエアが手元にあるコンピュータ（ローカル環境）において実行可能になっている必要がある。ここでは、Linux のローカル環境上で Bioinformatics 用ソフトウエアをインストールし、実行可能にするための手順を実習する。実際にインストールを行うのは、ゲノムアノテーションに実際に用いるソフトウエアである

- BLAST パッケージ（相同性検索ソフトウエア）
- Glimmer（原核生物遺伝子の予測ソフトウエア）
- ps_scan（prosite モチーフの検索ソフトウエア）

である。また、より詳細なアノテーションに有効である

- Rasmol（タンパク質立体構造表示ソフトウエア）
- ClustalW（マルチプルアラインメント作成ソフトウエア）
- NJplot（系統樹表示ソフトウエア）

についても、インストールを行う。

2　ソフトウエアをインストールするための準備

ホームディレクトリの下にある *works* ディレクトリにおいて以下のディレクトリを作成する。

- *softwares*　（ソフトウエアをインストールする際の作業用ディレクトリ）
- *db*　（データベースを格納するディレクトリ）
- *practice*（練習用ディレクトリ）
- *bin*　（実行するソフトウエア本体を格納するディレクトリ）

具体的には、次のコマンドを実行する。なお、コマンドの先頭にある“$”は shell のプロンプトなので入力しないこと。

```
$ cd ~/works ⏎
$ mkdir softwares db practice bin ⏎
```

以下、“*works* ディレクトリ”や“*softwares* ディレクトリ”、“*db* ディレクトリ”、“*practice* ディレクトリ”、“*bin* ディレクトリ”と表記した場合、特に断らない限りここで作成したディレクトリを指すこととする。

3 BLAST パッケージ

BLAST をローカル環境で実行するのに必要なものは、プログラム本体とデータベースである。まずは、BLAST プログラム本体のインストールについて解説する。Web ブラウザを起動し、以下のサイトにアクセスしてみよう。

ftp://ftp.ncbi.nih.gov/blast/executables/LATEST/

複数のファイルのリストが表示されるが、我々の実習環境で実行できるプログラムに関連するファイルは、

blast-2.2.17-ia32-linux.tar.gz

である。ファイル名の前半部分（blast-2.2.17）は、BLAST のバージョンを示す。また、後半部分（ia32-linux.tar.gz）は、Intel 32bit CPU 用 Linux で実行できる BLAST 関連プログラムが tar.gz 形式で圧縮されたファイルであることを意味する。

プログラム本体のインストール

あらかじめ NCBI のサーバから共有ディレクトリへダウンロードしてある BLAST パッケージの圧縮ファイル */common/softwares/blast-2.2.17-ia32-linux.tar.gz* を *softwares* ディレクトリにコピーし、解凍を行う。

```
$ cp /common/softwares/blast-2.2.17-ia32-linux.tar.gz ~/works/softwares ⏎
$ cd ~/works/softwares ⏎
$ tar zxf blast-2.2.17-ia32-linux.tar.gz -C ~/works/ ⏎
```

解凍によりできた *blast-2.2.17* ディレクトリの中身を確認しよう。

```
$ cd ~/works/blast-2.2.17 ⏎
$ ls -l ⏎
```

ディレクトリの中には、プログラムのバージョンが書かれた *VERSION* ファイル、プログラム本体が入った *bin* ディレクトリ、プログラムの実行に必要なデータが入った *data* ディレクトリ、プログラムの説明文等が入った *doc* ディレクトリがある。

次に、テキストエディタを起動し、

[NCBI]
Data=（ユーザ依存）/blast-2.2.17/data

と入力して、*.ncbirc* というファイル名で、ホームディレクトリに保存する。ここで、(ユーザ依存)のところは、以下のコマンドを実行して得られた文字列に置き換えてほしい。

$ cd ~/works; pwd ⏎

例えば、“Data=/mnt/bis/b105000/blast-2.2.17/data”のようになるはずである。なお、ここで作成した*.ncbirc* ファイルは、相同性検索に必要なパラメータがどのディレクトリにあるかを指定するものである。

最後に、テキストエディタで新しいファイルを作成し、

setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin

と入力して、*.tcshrc* というファイル名で、ホームディレクトリに保存する。*setenv* は、環境変数を設定するコマンドである。PATH という環境変数に *bin* ディレクトリおよび BLAST のプログラムがあるディレクトリを加えることで、それらのディレクトリに存在するプログラムは、プログラム名を入力するだけで実行できるようになる。また、端末を開くと最初に実行されるファイルである*.tcshrc* の中に *setenv* コマンドを書き込むことにより、端末が開くと BLAST へのパスが通されることになる。パスが通ったかを確認するために、新しい端末を開き、開いた端末上で

$ which blastall ⏎

と実行してみよう。プログラム名が絶対パスで表示されれば、ここまではうまくいっていることがわかる。もし、“コマンドが見つかりません”と返されたら、途中で入力ミスをしていないか確認しよう。特に、特殊記号（$, {}など）の入力間違いや、大文字・小文字も間違いなどに注意すること。

データベースの準備

Swiss-Prot や PDB などメジャーなデータベースについては、BLAST で検索できるようにフォーマットされたデータベースが NCBI のサーバに用意されている。Web ブラウザを起動し、以下のサイトにアクセスしてみよう。

ftp://ftp.ncbi.nih.gov/blast/db/

なお、NCBI のサーバで用意されているデータベースの詳細については、

ftp://ftp.ncbi.nih.gov/blast/db/blastdb.html

を参照して欲しい。また、フォーマットされる前の multi-FASTA 形式のファイルは、

ftp://ftp.ncbi.nih.gov/blast/db/FASTA/

に置かれている。

実際の作業では、フォーマット済みのデータを使うだけでなく、自前のデータを自分でフォーマットをする必要がでてくる。ここでは、練習のため、NCBI で公開されているデータ *pdbaa*（PDB に登録されているタンパク質のアミノ酸配列を集めたファイル）をフォーマットしてみよう。

BLAST 用データベースのフォーマットには、*formatdb* コマンドを用いる。共有ディレクトリにダウンロードしてある圧縮された *pdbaa*（*/common/2007/db/pdbaa.gz*）を *db* ディレクトリにコピーして解凍し、*formatdb* を実行しよう。具体的には以下のように入力する。

```
$ cp /common/2007/db/pdbaa.gz ~/works/db/ ⏎
$ cd ~/works/db/ ⏎
$ gzip -d pdbaa.gz ⏎
$ formatdb -i pdbaa -o T ⏎
```

ここで、*gzip -d* は、gz 形式の圧縮ファイルを解凍するコマンドである。*formatdb* は、フォーマットする multi-FASTA 形式のファイル名を-i の後に指定して実行させる。また、*-o T* オプションを追加すると、フォーマットしたデータから、*fastacmd* コマンド（後述）を使って FASTA 形式のデータを取得できるようになる。

なお、自前のデータが塩基配列の場合は、 基礎と実習バイオインフォマティクス p.66 2.2.3 「オリジナルのデータベースを作成し、BLAST 検索を実行する」を参照して欲しい。

BLAST の実行

ここまでの作業がうまくいっていれば、ローカル環境において BLAST を実行できるはずである。以下の手順に従って、BLAST を実行してみよう。

まずは、BLAST の query となるアミノ酸配列を取得する。３回生前期までの実習を思い出して、インターネット上から大腸菌の conserved protein（SwissProt Entry name:

YBHB_ECOLI)を FASTA 形式で取得しよう。取得したデータは、*YBHB_ECOLI.aa* というファイル名で、*practice*ディレクトリに保存すること。

次に、以下のように入力して PDB に対する BLAST を実行する。

```
$ cd ~/works/practice ⏎
$ blastall -p blastp -i YBHB_ECOLI.aa -d ~/works/db/pdbaa -o YBHB_ECOLI_pdb.
blp -e 0.01 ⏎
```

上記の *blastall* が BLAST のプログラム本体である。*-p* の後にどのアルゴリズムで相同性検索を行うか（今回の場合、タンパク質 vs タンパク質なので *blastp* を用いる、塩基配列 vs 塩基配列の場合は *blastn* となる）、*-i* の後には入力ファイル名、*-d* の後には検索するデータベース名、*-o* の後には検索結果を格納するファイル名を指定する。また、*-e* オプションを指定すると、*-e* の後の数字よりも小さい e-value をもつ検索結果だけが出力される。検索終了後に *YBHB_ECOLI_pdb.blp* をテキストエディタ等で開くと、以下のような検索結果を見ることができるはずである。

```
BLASTP 2.2.17 [Aug-26-2007]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Reference for composition-based statistics:
Schaffer, Alejandro A., L. Aravind, Thomas L. Madden,
Sergei Shavirin, John L. Spouge, Yuri I. Wolf,
Eugene V. Koonin, and Stephen F. Altschul (2001),
"Improving the accuracy of PSI-BLAST protein database searches with
composition-based statistics and other refinements",  Nucleic Acids Res. 29:2994-3005.

Query= P12994|YBHB_ECOLI UPF0098 protein ybhB - Escherichia coli
(strain K12).
         (158 letters)

Database: pdbaa
           31,946 sequences; 7,188,193 total letters

Searching.................................................done



                                                                 Score    E
Sequences producing significant alignments:                      (bits) Value

pdb|1VI3|A Chain A, Crystal Structure Of An Hypothetical Protein      302   3e-83
pdb|1FJJ|A Chain A, Crystal Structure Of E.Coli Ybhb Protein, A ...   301   6e-83
pdb|1FUX|A Chain A, Crystal Structure Of E.Coli Ybcl, A New Memb...   157   2e-39
pdb|2EVV|A Chain A, Crystal Structure Of The Pebp-Like Protein O...    45   1e-05

>pdb|1VI3|A Chain A, Crystal Structure Of An Hypothetical Protein
          Length = 170

 Score =  302 bits (773), Expect = 3e-83,   Method: Composition-based stats.
 Identities = 153/158 (96%), Positives = 154/158 (97%)

Query: 1   MKLISNDLRDGDKLPHRHVFNGMGYDGDNISPHLAWDDVPAGTKSFVVTCYDPDAPTGSG 60
           +KLISNDLRDGDKLPHRHVFNG GYDGDNISPHLAWDDVPAGTKSFVVTCYDPDAPTGSG
                                              :
                                              :
```

BLAST 用データベースからの配列の取得

“YBHB_ECOLI”のアミノ酸配列と類似性の見られた PDB エントリーのアミノ酸配列を、*pdbaa* から取得してみよう。*YBHB_ECOLI_pdb.blp* の中身を見ると、“pdb|1VI3|A”、“pdb|1FJJ|A”、“pdb|1FUX|A”、“pdb|2EVV|A”の４つのエントリーが得られたことがわかる。そこで、*practice*ディレクトリにおいて、以下のように入力しよう。

```
$ fastacmd -s "pdb|1VI3|A pdb|1FJJ|A pdb|1FUX|A pdb|2EVV|A" -d
~/works/db/pdbaa -o YBHB_ECOLI_homolog.aa ⏎
```

fastacmd は、BLAST 用のデータベースから指定した ID の配列を取得するプログラムである。*-s* の後に取得したい配列の ID（””で囲むこと）、*-d* の後に BLAST 用データベース、*-o* の後に取得した配列を格納するファイル名を指定する。実行後に作成された *YBHB_ECOLI_homolog.aa* の中身を見ると、指定した４つの配列データが格納されているはずである。

同じ結果は、以下のようにしても得ることができる。まず、取得したい配列の ID を列挙したファイルを用意する。例えば、エディタで新規書類を作成して、

```
pdb|1VI3|A
pdb|1FJJ|A
pdb|1FUX|B
pdb|2EVV|D
```

と書き込み、*YBHB_ECOLI_homolog.id* と言う名前で *practice* ディレクトリに保存する。次に、*practice*ディレクトリにおいて

```
$ fastacmd -i YBHB_ECOLI_homolog.id -d ~/works/db/pdbaa -o
YBHB_ECOLI_homolog.aa ⏎
```

と実行すれば、４つのエントリーのアミノ酸配列が得られる。*-i* は、ID が書かれたファイル名を指定するオプションである。大量の配列を取得したい場合は、*-i* オプションを使って ID の指定をした方が効率の良い場合が多い。

4 演習

1. */common/2007/db* に、Swiss-Prot に登録されているアミノ酸配列のデータが圧縮された *swissprot.gz* が置いてある。次の(ア)から(オ)までを実行しなさい。

   (ア) *pdbaa* を BLAST 用にフォーマットした手順に従って、*swissprot* を BLAST 用にフォーマットしなさい。

(イ) *pdbaa* から *fastacmd* を使って“pdb|1VI3|A”のアミノ酸配列を取得しなさい。

(ウ) フォーマットした *swissprot* に対して、“pdb|1VI3|A”のアミノ酸配列を query とした BLAST を実行しなさい。

(エ) BLAST の実行結果から query のアミノ酸配列と全長にわたって類似性が見られるエントリーの ID を取り出しなさい。

(オ) (エ)で得られた ID が示すアミノ酸配列を、*fastacmd* を用いて *swissprot* から取得しなさい。

5 立体構造観察

YBHB_ECOLI に最も配列類似性が高かった PDB エントリー“pdb|1VI3|A”は、どんな立体構造をしているだろうか。それを知るには、立体構造を観察するためのソフトウエアが必要である。３回生前期までの実習では、立体構造観察用ソフトウエアとして Windows 上で実行できる Rasmol および UCSF Chimera を用いてきた。これらのソフトウエアは、Linux 上でも実行できるバイナリが存在する。ここからは、Rasmol と UCSF Chimera を使えるようにセッティングを行う。

UCSF Chimera のセッティング

UCSF Chimera は、あらかじめ共有ディレクトリにインストールしてある（*/common/chimera-1.2422/bin*）。そのため、UCSF Chimera を起動するには、実行ファイルが存在するディレクトリにパスを通せば良い。具体的には、ホームディレクトリにある *.tcshrc* をエディタで開き、BLAST をインストールした時に書き込んだ

```
setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin
```

を、

```
setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin
:/common/chimera-1.2422/bin
```

と変更して保存する（紙面上は改行されているが、入力するときは改行しないこと）。新しい端末を開いて、開いた端末上で

```
$ chimera & ⏎
```

と入力すると、UCSF Chimera が起動するはずである。

Rasmol のインストール

ここでは、Rasmol をローカル環境にインストールする方法を示す。

まず、WWW ブラウザを使って以下の URL にアクセスしよう。

http://www.bernstein-plus-sons.com/software/rasmol/

表示された画面の中で、「Linux binaries」と書かれた列の「32」、および、「RasMol Help File」と書かれた列の「raw」と書かれたリンクをクリックすれば必要なファイルをダウンロードできる。

上記のサイトからダウンロードできるファイルが、共有ディレクトリ */common/softwares* に保存してある。以下のように入力してファイルを移動させよう。

```
$ mv /common/softwares/rasmol_32BIT ~/works/bin/rasmol ⏎
$ mv /common/softwares/rasmol.hlp ~/works/bin/ ⏎
$ chmod +x ~/works/bin/rasmol ⏎
```

最後にホームディレクトリの *.tcshrc* をエディタで開き、

```
setenv RASMOLPATH $HOME/works/bin
```

という行を追加する。新しい端末を開き、その端末上で

```
$ rasmol ⏎
```

と入力すると Rasmol が起動するはずである。

6 演習

1. Linux にインストールされた UCSF Chimera もしくは Rasmol を使って PDB エントリー “1VI3” と “1FUX” の立体構造を表示しなさい。この時、両方の立体構造が比較しやすい図を作成すること。

7　遺伝子領域予測

ここでは、原核生物のゲノム塩基配列から遺伝子領域を予測するために必要なソフトウエアの設定を行う。

まず、遺伝子領域の予測を行うソフトウエアである Glimmer のインストール方法を説明する。原核生物と真核生物では遺伝子構造の特徴が大きく違っており、そのために、遺伝子領域を予測するソフトウエアもそれぞれに専門化している。今回の実習で用いる Glimmer は原核生物用の遺伝子領域予測ソフトウエアで、TIGR や NCBI における原核生物ゲノムのアノテーションなどに使われているものである。後半では、tRNA 遺伝子を予測するソフトウエアである tRNAscan-SE のインストール方法について説明する。

Glimmer のコンパイル

以下の URL にアクセスすると、Glimmer のアルゴリズムの簡単な説明や予測精度の情報がある。またページの下部にはダウンロード用のリンクが見つかる。

http://cbcb.umd.edu/software/glimmer/

Glimmer はこれまでインストールしてきたソフトウエアと違い、“ソースコード”による配布である。BLAST パッケージや Rasmol などのソフトウエアは“バイナリファイル”による配布であるため、圧縮ファイルの解凍と簡単なパラメータ設定だけでプログラムを実行することができる。一方で、ソースコードで配布されているソフトウエアは、コンピュータが実行できる形式に変換（コンパイル）するという手順が加わる。ここでは、Glimmer のコンパイル方法を示す。なお、Glimmer のソースコードが圧縮されているファイルは、あらかじめ共有ディレクトリにダウンロードしてある（*/common/softwares/glimmer302.tar.gz*）。この圧縮ファイルを *softwares* ディレクトリにコピーし、コンパイルしてみよう。具体的には以下の様にコマンドを入力する。

```
$ cp /common/softwares/glimmer302.tar.gz ~/works/softwares/ ⏎
$ cd ~/works/softwares ⏎
$ tar zxf glimmer302.tar.gz -C ~/works/ ⏎
$ cd ~/works/glimmer3.02/src ⏎
$ make ⏎
```

最初の４行は、圧縮されたソースコードを共有ディレクトリからコピーして、解凍する操作である。次に *make* コマンドで、複数あるソースコードをまとめてコンパイルしている。*make* コマンドは、ソースコードで配布されているソフトウエアをコンパイルする際に、ほぼ必ず使われるコマンドである。Glimmer の場合は *make* を実行するだけでコンパイルが完了するが、これ以外のソフトウエアの場合は必ずしもそうはなっていない。それぞれ

のソフトウエアにはコンパイルの仕方が書かれたドキュメントは必ず付属しているので、そのドキュメントを良く読んでコンパイルを実行しよう。

コンパイルができたら、ホームディレクトリにある*.tcshrc* をエディタで開く。パスを設定している *setenv* の行

setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin:/common/chimera-1.2422/bin

に、以下の四角で囲んだ部分を追加する。

setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin:/common/chimera-1.2422/bin:$HOME/works/glimmer3.02/bin

追加後に新しい端末を開けば、Glimmer が使えるようになる。

Glimmer を使うための準備

Glimmer は１つのプログラムではなく、複数のプログラムを順番に実行していくことで遺伝子領域の予測を行う。これらの一連の作業を自動的に行うため、Glimmer にはいくつかの Shell Script が付属している。それらの Shell Script のうち、*g3-iterated.csh* を使えるようにしよう。この *g3-iterated.csh* を実行するには、少し準備が必要である。

まず、下記のページにある ELPH というソフトウエアを使えるようにする。これは、複数の塩基配列やアミノ酸配列からモチーフを探し出すソフトウエアであり、*g3-iterated.csh* の実行に必要である。

http://cbcb.umd.edu/software/ELPH/

ELPH もあらかじめ共有ディレクトリにダウンロードしてあるので、*softwares* ディレクトリにコピーし、ELPH のコンパイルをしよう。コンパイルできたら ELPH のバイナリファイルを *bin* ディレクトリにコピーしよう。

```
$ cp /common/softwares/ELPH-1.0.1.tar.gz ~/works/softwares/ ⏎
$ cd ~/works/softwares ⏎
$ tar zxf ELPH-1.0.1.tar.gz ⏎
$ cd ELPH/sources ⏎
$ make ⏎
$ cp elph ~/works/bin/ ⏎
```

次に、*g3-iterated.csh* を *bin* ディレクトリにコピーし、そのファイルをエディタで開く。

```
$ cp ~/works/glimmer3.02/scripts/g3-iterated.csh ~/works/bin/ ⏎
```

```
$ vi ~/works/bin/g3-iterated.csh ⏎
```

g3-iterated.csh では、実行に必要なプログラムがある場所を指定する必要があるが、その部分は自分の環境に合わせて書き換える必要がある。

```
set awkpath = /fs/szgenefinding/Glimmer3/scripts
set glimmerpath = /fs/szgenefinding/Glimmer3/bin
set elphbin = /nfshomes/adelcher/bin/elph
```

と書かれている行の四角で囲った部分を、以下のように書き換える。

```
set awkpath = （ユーザ依存）/glimmer3.02/scripts
set glimmerpath = （ユーザ依存）/glimmer3.02/bin
set elphbin = （ユーザ依存）/bin/elph
```

ここで、（ユーザ依存）のところは、以前にも出てきた以下のコマンドを実行して得られる文字列である。

```
$ cd ~/works; pwd ⏎
```

また、今回アノテーションを行うゲノムの特性に合った遺伝子予測をするために、「step1:」と書かれた行の付近にある

```
$glimmerpath/long-orfs -n -t 1.15 $genome $tag.longorfs
```

を、

```
$glimmerpath/long-orfs -n -t 1.08 $genome $tag.longorfs
```

と書き換える。これで Glimmer を使う準備が整った。

Glimmer による遺伝子領域予測

まず、NCBI Entrez から Locus ID が“ECOHU43”となっている塩基配列の FASTA 形式のデータを取得してほしい。この塩基配列は、大腸菌のゲノムの一部である。取得したデータが入ったファイルに *ECOHU43.fa* と名前を付け、*practice* ディレクトリに置こう。次に、新しい端末を開いて以下のように入力し、*g3-iterated.csh* を実行しよう。

```
$ cd ~/works/practice ⏎
$ g3-iterated.csh ECOHU43.fa ECOHU43⏎
```

これにより、ECOHU43 を共通の prefix に持つ複数のファイルが作られる。これらのファイルのうち、*ECOHU43.predict* が最終的な予測結果が書き込まれたファイルである。*ECHOHU43.predict* の中身は、

```
>gi|456556|gb|U00009.1|ECOHU43 sbcB region of E.coli K12 BHB2600
orf00001     1119      646  -1     7.88
orf00002     2374     1238  -2     8.37
orf00003     2541     4040  +3     9.32
orf00005     4310     4083  -3     6.02
orf00006     5373     4324  -1     8.63
orf00007     6925     5561  -2    11.32
orf00008      448     7101  -3     9.46
```

となっているはずである。このファイルの２行目以降の見方は、

- １カラム目　：　Glimmer によってつけられた、予測された遺伝子領域の ID
- ２カラム目　：　開始コドンの１塩基目の位置
- ３カラム目　：　終止コドンの最後の塩基の位置
- ４カラム目　：　リーディングフレーム
- ５カラム目　：　Glimmer でつけられたスコア

である。

アミノ酸配列への翻訳

遺伝子領域が予測できたら、次に必要なのは、その領域にコードされているアミノ酸配列である。Glimmer は遺伝子領域の予測結果を示すだけなので、予測された領域の具体的な塩基配列の取得やその配列のアミノ酸配列への翻訳は別のソフトウエアで行う必要がある。

まず、Glimmer の結果をもとに、予測された遺伝子領域に該当する塩基配列を抜き出そう。これは、Glimmer の一連のプログラムの１つである *extract* というプログラムにより実行できる。

```
$ extract -t ECOHU43.fa ECOHU43.predict > ECOHU43_orf.fa ⏎
```

これで、それぞれの ORF の塩基配列データが multi-FASTA 形式で得られる。

次に、得られた塩基配列データをアミノ酸配列に翻訳する。翻訳プログラムは、２回生前期の実習で作成した Perl 言語によるプログラムを改良しても良いが、今回は EMBOSS というパッケージに収録されている *transeq* を使用する。*transeq* は、multi-FASTA 形式の塩基配列を読み込んでそれぞれのエントリーをアミノ酸配列に翻訳してくれるプログラムである。*transeq* を実行できるようにするために、EMBOSS パッケージがインストールされているディレクトリにパスを通そう。

まず、ホームディレクトリにある*.tcshrc* をエディタで開く。パスを設定している *setenv* の行に EMBOSS パッケージがインストールされているディレクトリ（*/common/EMBOSS-5.0.0/bin*）を追加する。具体的には、

setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin:/common/chimera-1.2422/bin:$HOME/works/glimmer3.02/bin

を、

setenv PATH ${PATH}:$HOME/works/bin:$HOME/works/blast-2.2.17/bin:/common/chimera-1.2422/bin:$HOME/works/glimmer3.02/bin:/common/EMBOSS-5.0.0/bin

とする。パスを追加したら*.tcshrc* を保存する。新しい端末を開いて、その端末上で、

```
$ cd ~/works/practice ⏎
$ transeq ECOHU43_orf.fa ECOHU43_orf.aa ⏎
```

と入力すると、*ECOHU43_orf.aa* ファイルに multi-FASTA 形式のアミノ酸配列が納められる。

EMBOSS パッケージには、*transeq* の他にも、あると便利な配列解析用プログラムが多数（約 100 種類）収録されている。ここで詳しい紹介はしないが、必要なプログラムが EMBOSS パッケージに収録されているかどうかは、

http://emboss.sourceforge.net/

などを参照すれば確認できるだろう。

tRNA 遺伝子領域の予測

tRNA 遺伝子を予測するには、tRNA 遺伝子の２次構造体であるクローバーリーフ構造上に存在する、特定の位置に特徴的に出現する配列(コンセンサス配列)情報(図 1)を利用することが有効であると考えられる。実際に、コンセンサス配列に基づく予測プログラムがいくつか公開されており、その中でも代表的な tRNA 遺伝子予測プログラムとして tRNAscan-SE がある。予測精度としては、原核生物においては 99%と高精度な予測が可能となっていて、多くの原核生物のゲノムプロジェクトにおいて tRNA 遺伝子の予測に使われている。本実習では、tRNAscan-SE をローカルで実行できるように設定を行う。

なお、ここで注意しておいてほしいことは、tRNAscan-SE によって tRNA 遺伝子の大部分は予測可能であるが、全てを完璧に予測できているとは限らないということである。また、古細菌や真核生物においては、tRNA 遺伝子においてもイントロンが存在することが知られており、イントロンを持つ tRNA 遺伝子の予測においては、tRNAScan-SE においても予測精度が高いとはいえないのでより詳細な注意が必要である。

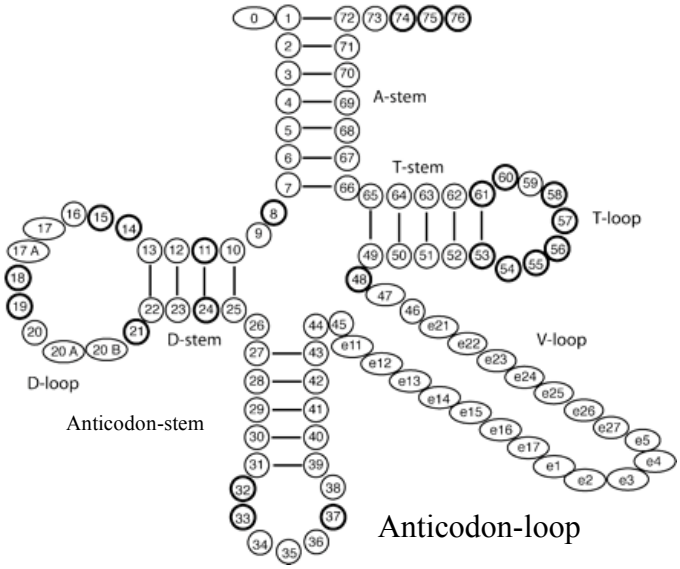

図 1　tRNA クローバーリーフ構造とクローバーリーフ構造上でのコンセンサス配列(図中黒丸）　Laslett, D. and Canback, B. (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. Nucleic Acids Research 32:11-16. の図を一部改変。

今回の授業においては詳細は説明しないが、そのほかの tRNA 遺伝子予測プログラムとしては、

- ARAGORN (http://130.235.46.10/ARAGORN1.1/HTML/)
- tDNAfinder (http://ei4web.yz.yamagata-u.ac.jp/~kinouchi/tRNAfinder/)
- SPLIT (http://splits.iab.keio.ac.jp/)

などがあり、イントロンを持つ tRNA の予測精度の向上や tmRNA の予測など改良が行われている。実際にゲノムアノテーションを行う場合は、これらの予測プログラムも組み合わせて tRNA 遺伝子領域の予測を行うとなお良い。

tRNAscan-SE のコンパイル

WEB ツール版の tRNAscan-SE は、以下の URL でアクセスできる。

http://lowelab.ucsc.edu/tRNAscan-SE/

また、同じサイトにおいて、tRNAscan-SE のソースコード（*tRNAscan-SE.tar.gz*）もダウンロードできる。なお、Web ツール版の tRNAscan-SE のバージョンは 1.21 と表示されているが、ダウンロード版の tRNAscan-SE のバージョンは 1.23 である。あらかじめ tRNAscan のソースコードを共有ディレクトリにダウンロードしてあるので（*/common/softwares/tRNAscan-SE.tar.gz*）、このファイルを *software* ディレクトリにコピーし、コンパイルしてみよう。

```
$ cp /common/softwares/tRNAscan-SE.tar.gz ~/works/softwares/ ⏎
$ cd ~/works/softwares/ ⏎
$ tar zxf tRNAscan-SE.tar.gz ⏎
$ cd tRNAscan-SE-1.23 ⏎
```

ここで、ソフトウエアをインストールする場所を指定するため、*Makefile* という名前のファイルを編集する。*Makefile* の中の

```
BINDIR  = $(HOME)/bin
LIBDIR  = $(HOME)/lib/tRNAscan-SE
MANDIR  = $(HOME)/man
```

という記述を探し、

```
BINDIR  = $(HOME)/works/bin
LIBDIR  = $(HOME)/works/lib/tRNAscan-SE
MANDIR  = $(HOME)/works/man
```

と書き換えよう。書き換えたら保存し、

```
$ make ⏎
$ make install ⏎
```

と実行するとコンパイルとインストールが完了する。これにより、*bin* ディレクトリ以下に tRNAscan-SE の実行ファイル一式が格納される。なお、csh 系の shell の場合、あらかじめパスが通してあるディレクトリ（今回の場合は bin ディレクトリ）に新規にプログラムを置いても、すぐには認識されない（つまり、プログラム名だけを入力して実行しようとしても“コマンドが見つかりません”と表示される）。これを解消するには、新しい端末を開いてその端末を使うか、以下のコマンドを実行する。

```
$ rehash ⏎
```

tRNAscan-SE の実行方法と結果の見方

実行の準備として、まず、NCBI Entrez から Locus ID が“X63976”である大腸菌由来の塩基配列を FASTA 形式で取得しよう。なお、tRNAscan-SE は、FASTA 形式の塩基配列データを読み込んで tRNA を予測する。取得したデータは“*X63976.fa*”という名前にし、*practice* ディレクトリに置いておこう。次に、以下のように入力すると、tRNAscan-SE が実行される。

```
$ cd ~/works/practice ⏎
$ tRNAscan-SE -B -o X63976.out1 -f X63976.out2 X63976.fa ⏎
```

tRNAscan-SE の実行により作成されるファイルは以下のような意味となる。

・ -o の後に指定したファイル（実行例では X63976.out1）

ゲノム上での位置情報と tRNA 遺伝子の種類についてスペース区切り形式のリストとして出力される。

```
Sequence                  tRNA    Bounds  tRNA    Anti    Intron Bounds   Cove
Name                     tRNA #  Begin    End     Type    Codon   Begin   End Score
--------                 ------  -----    ------  ----    -----   -----   ----    ------
gi|41594|emb|X63976.1|  1   4291    4366    Ala GGC 0   0   86.51
gi|41594|emb|X63976.1|  2   4406    4481    Ala GGC 0   0   86.51
gi|41594|emb|X63976.1|  3   1618    1543    Val TAC 0   0   94.34
                                                :
                                                :
```

各項目の意味は、左の列から、配列の ID、予測された tRNA の通し番号、tRNA の開始位置、tRNA の終了位置、アミノ酸、アンチコドン、イントロンの開始位置、イントロンの終了位置、予測スコアである。ここで、ストランドの向きがセンス側(+, 5'->3')に tRNA が予測された場合“開始位置 < 終了位置”となるが、アンチセンス側に予測された場合、“開始位置 > 終了位置”となるため、ストランドの向きに注意が必要である。

・ -f の後に指定したファイル（実行例では X63976.out2）

各予測 tRNA 遺伝子ごとにその配列と２次構造が記載されている。

```
gi|41594|emb|X63976.1|.trna1 (4291-4366)    Length: 76 bp
Type: Ala   Anticodon: GGC at 34-36 (4324-4326) Score: 86.51
        *    |    *    |    *    |    *    |    *    |    *    |    *    |    *
Seq: GGGGCTATAGCTCAGCTGGGAGAGCGCTTGCATGGCATGCAAGAGGtCAGCGGTTCGATCCCGCTTAGCTCCACCA
Str: >>>>>>>..>>>>........<<<<.>>>>>.......<<<<<.....>>>>>.......<<<<<<<<<<<....

gi|41594|emb|X63976.1|.trna2 (4406-4481)    Length: 76 bp
Type: Ala   Anticodon: GGC at 34-36 (4439-4441) Score: 86.51
        *    |    *    |    *    |    *    |    *    |    *    |    *    |    *
Seq: GGGGCTATAGCTCAGCTGGGAGAGCGCTTGCATGGCATGCAAGAGGtCAGCGGTTCGATCCCGCTTAGCTCCACCA
Str: >>>>>>>..>>>>........<<<<.>>>>>.......<<<<<.....>>>>>.......<<<<<<<<<<<....
                                                :
                                                :
```

str 行にて各 stem(図 1 を参照)でペアとなる塩基が表示されている。

8　モチーフ検索

タンパク質の機能を推定する場合、機能既知タンパク質に対する相同性検索は非常に有効な手段である。しかし、機能既知タンパク質の数は、全ゲノムに存在するタンパク質の数に比べて少なく、相同性検索だけでは機能推定できるタンパク質数に限界がある。これを補う方法はいろいろなものが考案されているが、中でもよく用いられるのがモチーフ検索である。ここでは、PROSITE データベースに登録されているモチーフを検索するソフトウエアである ps_scan が使えるよう設定をする。

ps_scan の入手とインストール

ps_scan は、PROSITE データベースを構築しているグループが公開しているソフトウエアである。Linux 環境で実行できる ps_scan を入手するには、WWW ブラウザで

ftp://ftp.expasy.org/databases/prosite/tools/ps_scan/

にアクセスし、

ps_scan_linux_x86_elf.tar.gz

をダウンロードすれば良い。

ダウンロードしたファイルは、ps_scan のバイナリファイルが圧縮されたものである。そのため、このファイルを解凍して、適切なディレクトリにプログラムを移動させるだけ ps_scan を実行できる。あらかじめ、共有ディレクトリに上記の圧縮ファイルをダウンロードしてきてあるので（*/common/softwares/ps_scan_linux_x86_elf.tar.gz*）、それを *softwares* ディレクトリにコピーし、解凍しよう。解凍できたら、パスが通してある *bin* ディレクトリに必要なプログラム（*pfscan*、*ps_scan.pl*、*psa2msa*）を移動する。具体的には、以下のように入力すれば良い。

```
$ cp /common/softwares/ps_scan_linux_x86_elf.tar.gz ~/works/softwares ⏎
$ cd ~/works/softwares ⏎
$ tar zxf ps_scan_linux_x86_elf.tar.gz ⏎
$ cd ps_scan ⏎
$ cp pfscan ps_scan.pl psa2msa ~/works/bin/ ⏎
$ rehash ⏎
```

これでインストールした ps_scan が実行できるはずである。

PROSITE モチーフのデータベース

ps_scan を使って PROSITE モチーフを検索するには、PROSITE モチーフのデータベースが必要となる。このデータベースは、以下の URL で配布されている。

ftp://ftp.expasy.org/databases/prosite/release_with_updates/

WWW ブラウザで上記の URL にアクセスすると、いくつかのファイルが表示されるはずである。これらのファイルのうち、

prosite.dat

が目的の PROSITE モチーフのデータベースである。このデータベースもあらかじめ共有ディレクトリにダウンロードしてある（*/common/2007/db/prosite.dat*）。このファイルを *db* ディレクトリにコピーしよう。

```
$ cp /common/2007/db/prosite.dat ~/works/db/ ⏎
```

次に、*.tcshrc* をエディタで開いて次の文字列を追加し、保存しよう。

```
setenv PROSITE $HOME/works/db
```

ps_scan は、環境変数 PROSITE に指定されているディレクトリに対して *prosite.dat* を探しにいく。その設定を *.tcshrc* に書き込むことで、いちいち *prosite.dat* がある場所をいちいち入力しなくとも済むようになる。

ps_scan の実行

PROSITE モチーフの検索をするために、検索対象となるアミノ酸配列を準備しよう。ここでは、練習のために機能既知のタンパク質 CheA のアミノ酸配列を用いる。このタンパク質の機能は、自己リン酸化を行い、そのリン酸をシグナルの下流タンパク質（CheB や CheY）に移すことで、大腸菌の走化性シグナルを伝えることである。Swiss-Prot から、LOCUS ID が“CHEA_ECOLI”となっているエントリーを FASTA 形式で取得し、*CHEA_ECOLI.aa* という名前で *practice* ディレクトリに保存しよう。これは、4. の演習が実行済みであれば、*db* ディレクトリにある *swissprot* をもとにして *fastacmd* を使って行うことができる。具体的には新規に端末を開いて次のように入力すれば良い。

```
$ cd ~/works/practice ⏎
$ fastacmd -s "CHEA_ECOLI" -d ~/works/db/swissprot -o CHEA_ECOLI.aa ⏎
```

次に、ps_scan を実行する。ps_scan は複数のプログラムの集合であり、Perl Script の *ps_scan.pl* がまとめて実行してくれる。*practice* ディレクトリにおいて次のように入力すれば、検索が始まる。

```
$ ps_scan.pl CHEA_ECOLI.aa ⏎
```

PROSITE モチーフが見つかった場合は、標準出力に結果が示される。*CHEA_ECOLI.aa* に対する結果を見ると、いくつかの PROSITE モチーフが見つかっているはずである。しかし、見つかったモチーフの中でも“PS00001 ASN_GLYCOSYLATION N-glycosylation site.”などは、本来その機能を持っていない場所でも偶然にモチーフパターンが一致してしまう確率の高いものである。このようなモチーフを除いた検索結果を得たい場合は、

```
$ ps_scan.pl -s CHEA_ECOLI.aa ⏎
```

として、*-s* オプションを付けて *ps_scan.pl* を実行する。そうすると、

- PS50851 CHEW CheW-like domain profile.
- PS50894 HPT Histidine-containing phosphotransfer (HPt) domain profile.
- PS00039 DEAD_ATP_HELICASE DEAD-box subfamily ATP-dependent helicases signature.
- PS50109 HIS_KIN Histidine kinase domain profile.

などの確実性の高いものが選ばれてくる。これらのモチーフは、(PS00039 を除いて) CheA の機能と直感的に一致するものである。

9 演習

1. NCBI Entrez から LOCUS ID が“X13462”となっている塩基配列のデータを取得して次のことを求め、わかりやすいようにまとめなさい。

   (ア) Glimmer により予測された遺伝子領域

   (イ) “X13462”のデータに既にアノテートされている遺伝子領域と、予測された遺伝子領域の比較

   (ウ) 予測された遺伝子領域にコードされるアミノ酸配列と、それに見つかる PROSITE モチーフ

10 マルチプルアラインメントと系統樹作成

マルチプルアラインメントは、機能部位の保存性などを見るのに役立ち、詳細な機能アノテーションに欠かせないものである。このマルチプルアラインメントを作成するソフトウエアでよく使われているものに ClustalW がある。ここでは、ClustalW をローカル環境にインストールし、それを使って系統樹も作成できるよう設定を行う。

ClustalW の入手

ClustalW は、以下の URL から入手できる。

ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalW/

いくつかファイルが表示されるが、Linux 環境で使える ClustalW の最新バージョンは

clustalw1.83.linux.tar.gz

である。これは共有ディレクトリ（*/common/softwares*）にあらかじめダウンロードしてある。

ClustalW のコンパイルとインストール

Linux 用の ClustalW はソースコードによる配布であり、コンパイルが必要となる。この手順は、Glimmer をコンパイルした時とほぼ同じとなっている。すなわち、ダウンロードされた圧縮ファイル（*/common/softwares/clustalw1.83.linux.tar.gz*）を *softwares* ディレクトリにコピーし、解凍する。*make* コマンドによりコンパイルすると、バイナリファイルが作られる。できたバイナリファイルをインストールするには、パスの通っている *bin* ディレクトリにそのバイナリファイルを移動させれば良い。具体的には以下のようなコマンドを実行する。

```
$ cp /common/softwares/clustalw1.83.linux.tar.gz ~/works/softwares ⏎
$ cd ~/works/softwares ⏎
$ tar zxf clustalw1.83.linux.tar.gz ⏎
$ cd clustalw1.83.linux ⏎
$ make -f makefile.linux ⏎
$ mv clustalw ~/works/bin/ ⏎
$ rehash ⏎
```

以上により、ClustalW を使えるようになっているはずである。

ClustalW の実行

ここでは、インストールした ClustalW を実際に使ってみる。ローカル環境にインストールされた ClustalW は、対話式でアラインメントを作成していく。アラインメントを作成する題材としては、BLAST をインストールした項目で作成した *YBHB_ECOLI_homolog.aa* を使おう。*practice* ディレクトリに移動し、*clustalw* を実行する。

```
$ cd ~/works/practice ⏎
$ clustalw ⏎
```

すると、以下のようなメインメニューが出力される。

```
*************************************************************
******** CLUSTAL W (1.83) Multiple Sequence Alignments  ********
*************************************************************
```

```
    1. Sequence Input From Disc
    2. Multiple Alignments
    3. Profile / Structure Alignments
    4. Phylogenetic trees

    S. Execute a system command
    H. HELP
    X. EXIT (leave program)


Your choice:
```

まず最初にアラインメントを作成する配列を読み込ませる必要がある。そこで“1”を入力すると（以下、四角で囲まれた文字はキーボードから入力した文字列とする）、

```
Your choice: 1

Sequences should all be in 1 file.

7 formats accepted:
NBRF/PIR, EMBL/SwissProt, Pearson (Fasta), GDE, Clustal, GCG/MSF, RSF.


Enter the name of the sequence file:
```

と表示される。読み込ませたいファイルの名前 *YBHB_ECOLI_homolog.aa* を入力すると、以下のように読み込ませたファイルに入っていた配列の情報が表示され、メインメニューに戻る。

```
Enter the name of the sequence file: YBHB_ECOLI_homolog.aa

Sequence format is Pearson
Sequences assumed to be PROTEIN


Sequence 1: gi|40889976|pdb|1VI3|A     170 aa
Sequence 2: gi|15826202|pdb|1FJJ|A     159 aa
Sequence 3: gi|15826205|pdb|1FUX|A     166 aa
Sequence 4: gi|85544553|pdb|2EVV|A     207 aa


 ************************************************************
 ******** CLUSTAL W (1.83) Multiple Sequence Alignments  ********
 ************************************************************


    1. Sequence Input From Disc
    2. Multiple Alignments
    3. Profile / Structure Alignments
    4. Phylogenetic trees

    S. Execute a system command
    H. HELP
    X. EXIT (leave program)


Your choice:
```

次は、マルチプルアラインメントの作成である。メインメニューの時に“2”を入力すると以下のようなマルチプルアラインメントメニューが出力がされる。

```
Your choice: 2

****** MULTIPLE ALIGNMENT MENU ******


   1.  Do complete multiple alignment now (Slow/Accurate)
   2.  Produce guide tree file only
   3.  Do alignment using old guide tree file
```

```
   4.  Toggle Slow/Fast pairwise alignments = SLOW

   5.  Pairwise alignment parameters
   6.  Multiple alignment parameters

   7.  Reset gaps before alignment? = OFF
   8.  Toggle screen display         = ON
   9.  Output format options

   S.  Execute a system command
   H.  HELP
   or press [RETURN] to go back to main menu


Your choice:
```

ここで“1”を入力すると、出力とガイドツリーを格納するファイル名を入力することを促される。これには enter キーを押せば自動的に[]で囲まれたファイル名にしてくれる。ファイル名が決定すると、マルチプルアラインメントの計算が始まる。

```
Your choice: 1

Enter a name for the CLUSTAL output file  [YBHB_ECOLI_homolog.aln]:

Enter name for new GUIDE TREE          file   [YBHB_ECOLI_homolog.dnd]:

Start of Pairwise alignments
Aligning...
Sequences (1:2) Aligned. Score:  98
Sequences (1:3) Aligned. Score:  47
Sequences (1:4) Aligned. Score:  15
Sequences (2:3) Aligned. Score:  49
Sequences (2:4) Aligned. Score:  16
Sequences (3:4) Aligned. Score:  19
Guide tree       file created:   [YBHB_ECOLI_homolog.dnd]
Start of Multiple Alignment
There are 3 groups
Aligning...
Group 1: Sequences:   2     Score:3433
Group 2: Sequences:   3     Score:2648
Group 3:                   Delayed
Sequence:4    Score:1367
Alignment Score 2303



Consensus length = 213
CLUSTAL-Alignment file created  [YBHB_ECOLI_homolog.aln]

CLUSTAL W (1.83) multiple sequence alignment


gi|40889976|pdb|1VI3|A     --------------------XSLKLISNDLRDGDKLPHRHVFNG------
gi|15826202|pdb|1FJJ|A     --------------------AXKLISNDLRDGDKLPHRHVFNG------
gi|15826205|pdb|1FUX|A     --------------------AEFQVTSNEIKTGEQLTTSHVFSG------
gi|85544553|pdb|2EVV|A     XGSSHHHHHHSSGRENLYFQGHXKTFEVXIQTDSKGYLDAKFGGNAPKAF
                                          :  .  :: ..:     *.*

gi|40889976|pdb|1VI3|A     XGYDGD-NISPHLAWDDVPAGTKSFVVTCYDPDAPTGSG--WWHWVVVNL
gi|15826202|pdb|1FJJ|A     XGYDGD-NISPHLAWDDVPAGTKSFVVTCYDPDAPTGSG--WWHWVVVNL
gi|15826205|pdb|1FUX|A     FGCEGG-NTSPSLTWSGVPEGTKSFAVTVYDPDAPTGSG--WWHWTVVNI
gi|85544553|pdb|2EVV|A     LNSNGLPTYSPKISWQKV-EGAQSYALELIDHDAQKVCGXPFVHWVVGNI
                         . :*  . ** ::*. *  *::*:.:   * ** . .*  : **.* *:

gi|40889976|pdb|1VI3|A     PADTRVLPQGFG-----SGLVAXPDGVLQT--------RTDFGKTGYDGA
gi|15826202|pdb|1FJJ|A     PADTRVLPQGFG-----SGLVAXPDGVLQT--------RTDFGKTGYDGA
gi|15826205|pdb|1FUX|A     PATVTYLPVDAGR----RDGTKLPTGAVQG--------RNDFGYAGFGGA
gi|85544553|pdb|2EVV|A     AHNVLEENASXXDKRIVQGVNSLTQGFIRSPLNESEKQRSNLNNSVYIGP
                        .  .     .        .    . * ::         *.::. : : *.

gi|40889976|pdb|1VI3|A     APPKGE-THRYIFTVHALDIERIDVDEGASGAXVGFNVHFHSLASASITA

Press [RETURN] to continue or  X  to stop:
```

以上で、*YBHB_ECOLI_homolog.aln* というファイルにマルチプルアラインメントが作成されるが、続いて系統樹も計算してみよう。上記の状態で“X”を入力するとマルチプルアラインメントのメニューに戻り、次に enter キーを押すとメインメニューに戻る。

```
Press [RETURN] to continue or  X  to stop: X


****** MULTIPLE ALIGNMENT MENU ******


   1.  Do complete multiple alignment now (Slow/Accurate)
   2.  Produce guide tree file only
   3.  Do alignment using old guide tree file

   4.  Toggle Slow/Fast pairwise alignments = SLOW

   5.  Pairwise alignment parameters
   6.  Multiple alignment parameters

   7.  Reset gaps before alignment? = OFF
   8.  Toggle screen display         = ON
   9.  Output format options

   S.  Execute a system command
   H.  HELP
   or press [RETURN] to go back to main menu


Your choice:
```

```
 ************************************************************
 ******** CLUSTAL W (1.83) Multiple Sequence Alignments  ********
 ************************************************************


    1. Sequence Input From Disc
    2. Multiple Alignments
    3. Profile / Structure Alignments
    4. Phylogenetic trees

    S. Execute a system command
    H. HELP
    X. EXIT (leave program)


Your choice:
```

系統樹の計算は、メニューの４である。“4”を入力し、系統樹メニューにする。

```
Your choice: 4


****** PHYLOGENETIC TREE MENU ******


   1.  Input an alignment
   2.  Exclude positions with gaps?       = OFF
   3.  Correct for multiple substitutions? = OFF
   4.  Draw tree now
   5.  Bootstrap tree
   6.  Output format options

   S.  Execute a system command
   H.  HELP
   or press [RETURN] to go back to main menu


Your choice:
```

まずは、系統樹のもとになるマルチプルアラインメントを読み込ませる。これはメニューの１となる。“1”を入力すると、アラインメントが納められたファイルを聞かれるので、先程作成された *YBHB_ECOLI_homolog.aln* を指定する。すると読み込まれたファイルに納められていた情報が表示され、系統樹メニューに戻る。

```
Your choice: 1

Sequences should all be in 1 file.

7 formats accepted:
NBRF/PIR, EMBL/SwissProt, Pearson (Fasta), GDE, Clustal, GCG/MSF, RSF.

Enter the name of the sequence file: YBHB_ECOLI_homolog.aln

Sequence format is Clustal
Sequences assumed to be PROTEIN

Sequence 1: gi|40889976|pdb|1VI3|A     213 aa
Sequence 2: gi|15826202|pdb|1FJJ|A     213 aa
Sequence 3: gi|15826205|pdb|1FUX|A     213 aa
Sequence 4: gi|85544553|pdb|2EVV|A     213 aa

****** PHYLOGENETIC TREE MENU ******

   1.  Input an alignment
   2.  Exclude positions with gaps?       = OFF
   3.  Correct for multiple substitutions? = OFF
   4.  Draw tree now
   5.  Bootstrap tree
   6.  Output format options

   S.  Execute a system command
   H.  HELP
   or press [RETURN] to go back to main menu

Your choice:
```

系統樹メニューに戻ったら、“4”を入力する。すると、計算結果を格納するファイル名を聞かれる。enter キーを押せば、[]内の名前のファイルとなる。

```
Your choice: 4

Enter name for PHYLIP    tree output file   [YBHB_ECOLI_homolog.ph]: ↵

Phylogenetic tree file created:   [YBHB_ECOLI_homolog.ph]

****** PHYLOGENETIC TREE MENU ******

   1.  Input an alignment
   2.  Exclude positions with gaps?       = OFF
   3.  Correct for multiple substitutions? = OFF
   4.  Draw tree now
   5.  Bootstrap tree
   6.  Output format options

   S.  Execute a system command
   H.  HELP
   or press [RETURN] to go back to main menu

Your choice:
```

計算が終了すると系統樹メニューに戻るので、enter キーを押し、メインメニューに戻る。メインメニューでは“X”を入力すると *clustalw* を終了することができる。

```
Your choice: ⏎


 ************************************************************
 ******** CLUSTAL W (1.83) Multiple Sequence Alignments  ********
 ************************************************************


    1. Sequence Input From Disc
    2. Multiple Alignments
    3. Profile / Structure Alignments
    4. Phylogenetic trees

    S. Execute a system command
    H. HELP
    X. EXIT (leave program)

Your choice: X
```

以上が、*clustalw* を使ったマルチプルアラインメントと系統樹の計算の基本的な流れである。パラメータの変更も同様の対話形式で行うことができる。

系統樹の表示

Linux 環境で系統樹を表示するには、NJplot が便利である。このソフトウエアは、ClustalW の計算結果から、有根系統樹型の系統樹を表示してくれる。Windows 環境で使った TreeView も使うことができるが、インストールにやや難しいところがあるのでここでは扱わない。

NJplot の配布元は

http://pbil.univ-lyon1.fr/software/njplot.html

である。上記の URL に WWW ブラウザでアクセスすると、Linux 用の NJplot をダウンロードできる ftp サーバーへのリンク（下記の URL）が張られている。

ftp://pbil.univ-lyon1.fr/pub/mol_phylogeny/njplot/

なお、この ftp サーバーからは、無根系統樹を表示できる *unrooted.linux* というプログラムもダウンロードできる。

ダウンロードした *njplot.linux* を *bin* ディレクトリに移動させ、次に、実行権限を与える。具体的には以下のように入力する。

```
$ mv ~/Desktop/njplot.linux ~/works/bin ⏎
$ chmod +x ~/works/bin/njplot.linux ⏎
$ rehash ⏎
```

practice ディレクトリにおいて以下のように入力すれば、先程作成したマルチプルアラインメント（*YBHB_ECOLI_homolog.aln*）をもとにした系統樹が表示される。

```
$ cd ~/works/practice ⏎
$ njplot.linux YBHB_ECOLI_homolog.ph ⏎
```

NJplot の“File”メニューにある“Save as PDF”を選ぶと、表示した系統樹を PDF 形式で保存できる。PDF 形式は非常によく使われるフォーマットではあるが、Office 系のソフトウエアとの相性が悪い。そこで、PDF 形式で保存したファイルを、PNG 形式などの Office 系のソフトウエアが扱える画像形式に変換しておくと良い。それには、*convert* というコマンドを用いる。例えば、PDF 形式で保存したファイルの名前を *YBHB_ECOLI_homolog.pdf* とすると、

```
$ convert YBHB_ECOLI_homolog.pdf YBHB_ECOLI_homolog.png ⏎
```

と入力する。これにより、Microsoft Office や OpenOffice.org などのソフトウエアからでも読み込むことができる PNG 形式の系統樹が得られる。

11 演習

LOCUS ID が“CHEA_ECOLI”である Swiss-Prot エントリーを query として、Swiss-Prot に対して BLAST を実行しなさい。なお、e-value は 1e-6 とする。次に配列類似性が見られたアミノ酸配列のマルチプルアラインメントを ClustalW を使って作成しなさい。最後に、その系統樹を作成しなさい。

12 付録

blastall の代表的なオプション

-p : 用いるアルゴリズム（blastp, blastn, blastx, tblastx, tblastn） （必須）

-i : 入力ファイルを指定（デフォルトは標準入力）

-o : 出力ファイルを指定（デフォルトは標準出力）

-d : データベースを指定（formatdb でフォーマット済みであること）

-e : e-value を指定（デフォルトは 10.0）

-F : フィルタの使用を指定（デフォルトは Low complexity フィルタを使用する。フィルタを使用しない場合は、“-F F”とする）

-v : 検索結果における一行表示の最大数（デフォルトは 500）

-b : 検索結果におけるアラインメントの最大表示数（デフォルトは 250）

formatdb の代表的なオプション

-i : フォーマットするファイルの名前を指定

-p : フォーマットするデータがタンパク質か核酸かを指定（デフォルトは“-p T”（タンパク質）。核酸の配列をフォーマットする場合は、“-p F”とする）

-o : 検索用インデックスを作成するかどうかを指定（デフォルトは“F”。作成する場合は“-o T”とする）

fastacmd の代表的なオプション

-s : ID の直接指定（指定する ID は""で囲むこと）

-i : ID が記述されたフィルの指定

-d : 配列を取得するデータベースを指定（formatdb の実行時に“-o T”が指定してあることが必要

-o : 結果を格納するファイル名を指定（デフォルトは標準出力）

-L : 取得したい配列の範囲を指定（デフォルトは“-L 0,0”。 10 残基目から 150 残基目までを取得したければ、“-L 10,150”、10 残基目から最後の残基までを取得したければ、“-L 10,0”、 最初の残基から 150 残基目までを取得したければ、“-L 0,150”）

tRNAscan-SE の代表的なオプション

その他の詳細なオプション情報については、*"tRNAScan-SE -h"*で確認してほしい。

・予測対象を選択する上でも必ず付けるオプション

-B or -P : bacterial tRNAs を探索

-A : archaeal tRNAs を探索

-O : organellar (mitochondrial/chloroplast) tRNAs を探索

-G : general tRNA model (cytoplasmic tRNAs from all 3 domains included)を索用

・予測方法の選択のためのオプション

-C : 共分散モデル分析のみを使用した探索（感度は高いが時間が非常に掛かる。）

・出力オプション

-o <file> : tRNA 遺伝子領域予測結果の出力(リスト版)

-f <file> : tRNA 遺伝子領域予測結果の出力(２次構造も記載)

-m <file> : tRNA 遺伝子領域予測結果を基にした統計情報が出力される。

(speed, # tRNAs found in each part of search, etc)

ps_scan.pl の代表的なオプション

-s : 非特異的に見つかる PROSITE モチーフを結果から除去

-p : 検索したい PROSITE モチーフを指定

-d : PROSITE モチーフデータベースを指定

-o : 結果出力のフォーマットを指定

その他の実習でててきたプログラムの使い方

g3-iterated.csh

$ g3-iterated.csh （FASTA 形式の塩基配列データが入ったファイル） （prefix）

extract

$ extract （FASTA 形式の塩基配列データが入ったファイル） （g3-iterated.csh から得られた.predict ファイル）

transeq

$ transeq （FASTA 形式の塩基配列データが入ったファイル） （結果を格納するファイル）