

課題 1-2 (塩基使用頻度解析その2)

0.1 課題 1-2

マイコプラズマ菌全ゲノムファイルから配列を読み込めるようにプログラムを変更し、そのA,T,G,Cの数を調べる。1-1の結果と比較し、担当遺伝子配列の塩基使用頻度に偏りがなかったか考察する。

1-1 と 1-2 は非常に良く似ています。実はプログラムのにはほぼ同じとも言えます。できる人は一緒にやった方が効率が良いでしょう。要は、データファイルの違いだけなのです。1-1 で使うデータファイルはエディタで加工しているので、純粋にデータのみです。しかし、1-2 で使うmg.seq はコメントその他を含んでいますので、そこからデータのみを取り出すことを考えなければいけません。しかしそこをクリアすれば両者の間に違いはほとんどありません。

別々にプログラムを書いて手作業で結果を比較しても、もちろん良いのですがせっかくですから、一気にそこまでコンピュータにやらせることを考えてみましょう。

とりあえず課題をよく読んで、問題の意味、解法を考えてみて下さい。それからプログラムの概略を考えます。

1 プログラムデザイン

1.1 データ読み込み部の改良

1-1 の検索部分は下のようでした。

```
#!/usr/local/bin/perl

my($A,$T,$G,$C,$seq);

open(FILE, *****);

while(<FILE>){
    ***** # 改行コードとりのぞき
    ***** # $seq に読み込んだ行を連結
}
close(FILE);

$A = *****;
$T = *****;
$G = *****;
$C = *****;
```

さて、何を付け足せば良いでしょうか。少しデータファイルを見てみましょう。

```
LOCUS       L43967      580074 bp    DNA   circular   BCT           05-NOV-1998
DEFINITION  Mycoplasma genitalium G37 complete genome.
ACCESSION   L43967
KEYWORDS
(中略)
ORIGIN
1 taagttatta tttagttaat acttttaaca atattattaa ggtatttaaa aaatactatt
(中略)
580021 gaaatgatca tatatttaaa tgattataat atttctttaa tactaaaaaa atac
//
```

データの前にまずコメントがあります。その中の 1 つの "ORIGIN" 以降は塩基データなのでまずこれが出てくるまで読みとったものを無視するようにすれば良いでしょう。次にデータファイルの中には塩基情報以外の終わりには "/" と書くことになっています。これも頭に入れておいて下さい。

以上のことから課題 1-1 との違いを考慮すると、例えばこうなります。

```
#!/usr/local/bin/perl

my($A,$T,$G,$C,$seq);

open(FILE, *****);

while(<FILE>){

    if(*****){
        while(<FILE>){
            last if (/¥/¥//);          /// が含まれる行を見つけたら終了
            *****                  # 改行コード、数字とりのぞき
            *****                  # $seq に読み込んだ行を連結
        }
    }

}

close(FILE);

$A = *****;
$T = *****;
$G = *****;
$C = *****;
```

2 出力データの比較

2.1 小数演算

課題 1-2 では 1-1 の結果と比較することが求められています。母集団の大きさが決定的になるデータを比較するのですから、出力結果を割合(パーセント)でも表示することを考えましょう。それも少しだけ厳密に比べたいので小数第二位くらいまであればいいですね。

ところでコンピュータで小数を扱うには多少注意が必要です。例えば A (アデニン) の割合を出したければ、式はこうなりますね。

```
$percent=($A/length($seq))*100;
```

length() 関数は変数の長さを調べる関数です。文字列の場合、総文字数を返します。つまり、\$seq を length() 関数で見ているのですから、マイコプラズマゲノム全体の塩基数が返されることになります。

2.2 小数の出力

小数の出力は C と同じように printf を使って整形できます。普通に print \$percent と出力してもいいのですが、これでは小数点以下 8 桁くらいまで表示されて少々見にくいですね。

```
printf("A:  %.2f¥n", $percent);
```

上のように f の前に .2 を書いておくと、小数点以下第二位まで表示という意味になります。同様に %.4f ならば小数点以下第四位までです。

3 発展：1-1 と 1-2 を同時に処理する

さて、上の 2 点に注意して 1-1 を再利用すれば、でき上がったも同然です。2 つのプログラムはほとんど変わらないということがよくわかったと思います。そこで余力のある人は、せっかくですから 2 つを統合することを考えてみましょう。例えばプログラムを実行すると比較するファイル名を聞いてきて、必要があればファイル名を入力する、そうでなければデフォルトでも OK。で、走らせると一気に比較までしてくれる。非常に楽でいいですね。こうなると初歩の練習課題も実際に使えるプログラムへと 1 歩 1 歩近付いていくでしょう。