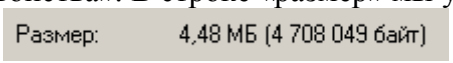
БиоИнформатика и *E. Coli* *(16 баллов)*

Е.А. Макеева, Т.И. Бидыло

Часть 1. ДНК и данные

1. *Посмотрите свойства сохраненного вами файла: сколько байт составляет его размер?* ***(1 балл)*** *Рассчитайте точное число пар оснований в геноме E. Coli.* ***(2 балла)***

Узнать свойства скачанного файла можно в проводнике Windows, кликнув по нему правой клавишей мышки и выбрав «свойства». В строке «размер» мы увидим:



т.е. файл занимает 4 708 049 байт.

Поскольку 1 байт = 8 бит, а каждый символ кодируется 8 битами, то количество байт равно количеству символов в файле.

По условию, на строки, несодержащие нуклеотиды, приходится 87 символов, значит, на строки с нуклеотидами приходится 4708049-87 = 4707962 символов. Поскольку на 70 букв нуклеотидов приходится один символ конца строки, то каждая строка содержит суммарно 71 символ. 4707962/71 ≈ 66309,32, значит, буквы нуклеотидов располагаются на 66310 строках. Столько же в этих строках символов переноса строки, следовательно, количество символов нуклеотидов (и число пар оснований в геноме) будет 4707962-66310 = **4 641 652**.

2. *Оцените, сколько примерно мегабайт (MB) будет занимать файл генома, имеющий длину в одну мегабазу (1Mbp), при таком же способе электронной записи файла.* ***(1 балл)*** *Поместится ли на обычный DVD диск файл с геном человека, имеющим длину 3 234.83 Mb?* ***(1 балл)***

Вклад первой служебной строки (заголовка) будет пренебрежимо мал по сравнению с миллионом символов, поэтому можно оценить, какой объем приходится на 1Mb исходя из имеющегося файла генома *E. Coli*. Поскольку 1MB = 1024·1024 = 1 048 576 байт, а 1Mbp это 1000·1000 = 1 000 000 пар нуклеотидов, то на 1Mbp будет приходиться 4708049/(1048576)/(4641652/1000000) ≈ 0,967 мегабайт (MB) информации. Для записи файла с геномом человека потребуется *3 234,83*·0,967 ≈ 3 128,08 MB. Поскольку обычный DVD диск помещается около 4 700 MB информации, значит, геном человека поместится с запасом.

3. *На любом языке программирования напишите программу (к решению приложите ее код), которая прочитает скачанный вами файл и сосчитает суммарное число нуклеотидов в геноме E. Coli и процентное содержание каждого из них по отдельности, чему они равны?* ***(4 балла)***

Результат работы программы (число символов нуклеотидов совпадает с расчетом, проведенным исходя из размера файла):

```
n=4641652
A=1142742, 24.6192950268568%
C=1141382, 24.5899951138086%
```

```
G=1180091, 25.423943888943%
T=1177437, 25.3667659703916%
```

Пример программы на языке Pascal (PascalABC.NET http://pascalabc.net/)

```
{Подсчет числа нуклеотидов в файле coli.txt и вычисление их процентного
содержания, PascalABC.NET}
Var
  f: Text;
  s: String;
  ch: Char;
  n, nA, nC, nG, nT: Longint;
  wA, wC, wG, wT: Real;
BEGIN
n:=0; nA:=0; nC:=0; nG:=0; nT:=0;
Assign(f,'coli.txt');
Reset(f);
Readln(f,s); {пропускаем первую служебную строку}
While (not Eof(f)) do
begin
  Read(f,ch); {читаем символ из файла}
  if (ch='A') or (ch='C') or (ch='G') or (ch='T') then
    begin
    n:=n+1;
     case ch of
      'A' : nA:=nA+1;
      'T' : nC:=nC+1;
      'C' : nG:=nG+1;
      'G' : nT:=nT+1;
     end;
    end
end;
Close(f);
wA:=100*nA/n; wC:=100*nC/n; wG:=100*nG/n; wT:=100*nT/n;
Writeln('n=', n);
Writeln('A=', nA, ', ', wA, '%');
Writeln('C=', nC, ', ', wC, '%');
Writeln('G=', nG, ', ', wG, '%');
Writeln('T=', nT, ', ', wT, '%');
END.
```

Часть 2. Поиск настоящих ДНК-палиндромов

4. *На любом языке программирования напишите программу для поиска ДНК-палиндромов в файле генома E. Coli K-12 и найдите все палиндромы с длиной ножки от 17 до 25 пар оснований (максимальную длину петли при поиске ограничьте 8 нуклеотидами). Приложите к решению текст программы и найденные палиндромы (достаточно привести номер первого нуклеотида палиндрома, число пар нуклеотидов в ножке и число нуклеотидов в петле). (7* ***баллов****)*

Всего можно найти **6** палиндромов удовлетворяющих условию (первое число – номер первого нуклеотида палиндрома, в скобках приведены длина ножки и петли):

59280 GCTAAGGTTGAAGGGGC| TGGAAC |GCCCCTTCAACCTTAGC (**17/6**)
1446340 TAACTAATTGGCGTTGCA| GTACA |TGCAACGCCAATTAGTTA (**18/5**)
2068389 TCCACGGACCGCACTCTT| ATGTC |AAGAGTGCGGTCCGTGGA (**18/5**)
2192450 AAAGCCGAAATCATTTAT| |ATAAATGATTTCGGCTTT (**18/0**)
2305067 ATACGCCACATCCGGCAT| ACC |ATGCCGGATGTGGCGTAT (**18/3**)
4520483 ATACTGTAAAGCCGGAG| ACATG |CTCCGGCTTTACAGTAT (**17/5**)

Программа, приведенная ниже, также находит меньшие палиндромы, на самом деле «перекрывающиеся» с более крупными, например:

1446340 TAACTAATTGGCGTTGC|AGTACAT|GCAACGCCAATTAGTTA 17/7
1446340 TAACTAATTGGCGTTGCA|GTACA|TGCAACGCCAATTAGTTA **18/5**
1446341 AACTAATTGGCGTTGCA|GTACA|TGCAACGCCAATTAGTT 17/5

Число таких «ложных» палиндромов невелико, поэтому, чтобы не утяжелять программу, допускалось отфильтровать их вручную.

```
{Поиск ДНК палиндромов в файле coli.txt, PascalABC.NET}

 {функция PAL нахождения обратной комплементарной последовательности}
 function PAL (S : String) : String;
 var J: Integer; Stmp: String;
  begin
  Stmp:='';
{идем с конца строки и "собираем" в Stmp обратную комплементарную строку}
   for J := 1 to length(S) do
     begin
        case S[length(S)-J+1] of
        'A' : Stmp:=Stmp+'T';
        'T' : Stmp:=Stmp+'A';
        'C' : Stmp:=Stmp+'G';
        'G' : Stmp:=Stmp+'C';
        end;
     end;
   PAL := Stmp;
 end;

Var
  f: Text;
  STR, sTmp, Stem1, Stem2, Loop: String;
  char: Char;
  n, i, StemLmin, StemLmax, StemL, LoopLmin, LoopLmax, LoopL, StrL, S2Pos :
Longint;
BEGIN

StemLmin := 17; {минимальная длина ножки}
StemLmax := 25; {максимальная длина ножки}
LoopLmin := 0;  {минимальная длина петли}
LoopLmax := 8;  {максимальная длина петли}

strL := 2*StemLmax+LoopLmax;  {число символов в строке поиска}

Assign(f,'coli.txt');
Reset(f);
Readln(f,sTmp); {чтобы пропустить первую служебную строку}

While (not Eof(f)) do
begin
Read(f,char);
{читаем символ из файла,
если прочитанный символ - нуклеотид, обрабатываем его}
if ( (char='A') or (char='C') or (char='G') or (char='T') ) then
begin
  n := n+1; {счетчик прочитанных нуклеотидов}
  STR := STR+char; {добавляем прочитанный нуклеотид в строку}
  if length(STR)=StrL then {обрабатываем, если строка уже наполнена символами}
  begin
    i := n-StrL+1; {позиция первого символа последовательности STR}
```

```
    {перебираем подстроки str с размером ножки от min до max}
    for StemL := StemLmin to StemLmax do
    begin
      Stem1 := copy(STR, 1, StemL); {берем подстроку STR от 1-го до до StemL-го
символа}
      Stem2 := PAL(Stem1); {находим обратную комплементарную последовательность}
      S2Pos := pos(Stem2, STR); {ищем ее в исходной строке STR, S2Pos -
найденное положение}
      LoopL := S2Pos-StemL-1;  {длина петли}
      Loop+ := copy(STR, StemL+1, LoopL);  {нуклеотиды петли}
      if (LoopL>=LoopLmin) and (LoopL<=LoopLmax) then {выводим, если длина петли
подходит}
            writeln (i, ' ', Stem1, '| ', Loop, ' |', Stem2, ' (', StemL, '/',
loopL, ')');
    end;

    {отбрасываем первый символ STR,
    чтобы начало строки на следующем шаге приходилось на следующий нуклеотид}
    STR:=copy(STR,2,length(STR));
  end;
end;
end;

END.
```