Short-project : GC content

Emmanuelle Becker

Table des matières

1 Introduction 1
2 Creating a class Window 1
3 Constructors of the Window class 2
4 Some additional methods 2
5 Creation of a SequenceNucleotide class and first tests 2
6 Exploration of a human chromosome 1 sequence with a sliding window 3
7 Exploration of a random sequence with a sliding window. 3
8 Theoretical questions 3

1 Introduction

DNA is composed of 4 bases : A, C, G and T, the A pairing with T and C with G. The rate of the G and C bases, called ”GC content” is often used in bioinformatics. For more information, I invite you to consult : http://en.wikipedia.org/wiki/CG_rate.

In this exercise, we will read part of a human chromosome, and calculate levels of GC content on this chromosome portion. We will compare the result with a randomly generated sequence with the same overall GC content, to see if the GC distribution is identical in both sequences (random sequence vs. human chromosome).

2 Creating a class Window

In a gcContent package, first create a Window class with two private attributes :

— seq, a sequence of type String;
— n, an integer corresponding to the size of the window.

Since these fields are private, do not forget to create accessors. You will not need modifiers.
3 Constructors of the Window class

We will now define two constructors for this class by using polymorphism, which allows to give two methods the same name, if their context (= their signature) are different:

— a first constructor $\text{Window(String } s\text{)}$, to which the required sequence $s$ is passed;
— a second constructor $\text{Window(int } len, \text{ double } p\text{)}$, which will create a random sequence of length $len$, whose percentage in GC is $p$.

Note 1: Having a percentage in GC of $p$ means that:

— each character has a probability $p/2$ of being a C and $p/2$ of being a G;
— there is necessarily a percentage in AT of $(1 - p)$, so the character A has a probability of $(1 - p)/2$ and the character T has a probability of $(1 - p)/2$;
— notice that $(p/2) + (p/2) + (1 - p)/2 + (1 - p)/2 = 1$;

Note 2: Use $\text{Math.Random()}$ methods.

4 Some additional methods

In this $\text{Windows}$ class, we will insert public methods:

— the method $\text{percentageGC()}$, which returns a double giving the GC percentage of our sequence;
— overload the $\text{toString()}$ method, which displays the following text:

"The current sequence, of length ??, has a percentage CG of ??\%." where are the ?? are replaced by equated ad values.

5 Creation of a SequenceNucleotide class and first tests

To verify that the $\text{Windows}$ class is working, create in the same package a $\text{SequenceNucleotide}$ class that contains a public static void main(String[] args) method.

For example, you can test with the following code:

```java
public static void main(String[] args){
    Fenetre iFen=new Window(1000,0.7) ;
    double p=iFen.percentageGC() ;
    System.out.println("The probability p is "+p) ;
    System.out.println(iFen) ;
}
```

Normally, if your random constructor and percentage calculation method work well, you should find that the GC rate is around 0.7 ...
6 Exploration of a human chromosome 1 sequence with a sliding window

A DNA sequence has been placed in the "masequence.txt" file. To read this sequence, we propose the following instructions:

```java
Scanner scanner=new Scanner(new FileReader("path_to_file"));
String maSequence=scanner.nextLine();  // pour sauter la 1ere ligne */
maSequence=scanner.next();  // pour lire la suite */
```

First, calculate the GC percentage of the human sequence (for example, you can build an iFen instance of the Windows class with the complete sequence and use the System.out.println(ifen)).

Then, we propose to calculate the percentage of GC of this sequence with a sliding window of size 1000 and a step of 500.

7 Exploration of a random sequence with a sliding window.

This time, we propose to create a random sequence of the same length as "masequence.txt" and whose percentage of GC is identical to that of "masequence.txt". Next, calculate the percentage of GC bases on this sequence with a sliding window of size 1000 and a step of 500.

Note: For example, you can construct a window with the random constructor, and then use the getSequence() method to retrieve this random sequence.

8 Theoretical questions

Try to make a graph with the results of the previous sections (eg with Libre Office calc, R, gnuplot or whatever suits you), and compare.

Considering that:

1. our sequence is a population $P$ of nucleotides (population in the statistical sense ...);
2. that the character "GC" is present with a frequency $f$ in $P$;
3. a window of size $t$ is a sample $E_i$ of $P$ of size $t$;

Which probability law should describe the frequency $f_i$ of the character GC on the samples $E_i$ of size $t$?

Do not hesitate to make a test of conformity to the said law by using R, in the case of the random sequence and in the case of the theoretical sequence.

Good luck!