Sanger sequencing

Frederick Sanger (1918-2013) was a British biochemist who won the Nobel Prize in chemistry twice, one of two people who have done so in the same category (the other is John Bardeen in physics), the fourth person overall with two Nobel Prizes, and the third person overall with two Nobel Prizes in the sciences. In 1958, he was awarded a Nobel Prize in chemistry "*for his work on the structure of proteins, especially that of insulin*". In 1980, Walter Gilbert and Sanger shared half of the chemistry prize "*for their contributions concerning the determination of base sequences in nucleic acids*". The other half was awarded to Paul Berg "*for his fundamental studies of the biochemistry of nucleic acids, with particular regard to recombinant DNA*".

Sanger's second Nobel Prize was awarded for a method that he and his colleagues developed in 1977, currenly known as **Sanger sequencing**. It was the most widely used sequencing method for approximately 25 years. More recently, Sanger sequencing has been supplanted by **Next Generation sequencing** methods, especially for large-scale, automated genome analyses. However, the Sanger method remains in wide use, for smaller-scale projects, validation of Next-Gen results and for obtaining especially long contiguous DNA sequence reads (> 500 nucleotides). The following animation gives a detailed description of how Sanger sequencing works.

In this assignment we represent a **DNA sequence** as a string that only contains the uppercase letters `A`, `C`, `G` and `T`, representing the individual bases (or nucleotides) of the sequence. In addition, all you need to know about the Sanger method is that sequencing a DNA molecule yields a large collection of **fragments**, of which only the length and the last base are known. We will represent such a fragment as a string that starts with zero or more dashes (`-`) representing the unknown bases at the start of the DNA sequence), followed by one of the four base letters. As an example, the following string represents a fragment of length 6 that tells us that the sixth base letter of the DNA sequence is a `G`.

```
-----G
```

All fragments that are obtained after sequencing a DNA molecule are stored in a text file that contains one fragment per line. Because the length of each fragment is determined by a stochastic process, it is possible that there are no fragments of a given length or that there are multiple fragments having the same length. Due to errors during the sequencing process, it is quite possible that some fragments have the same length but a different base at the end. As a result, for some positions the correct base letter cannot be derived unambiguously.

```
----T
```

```
C
-C
-----G
C
-----C
---T
----A
----C
-----G
```

Based on the fragments in the sample file above, we can infer that the first base of the DNA sequence is a C (confirmed by two fragments), that the second base is also a C, and that the fourth base is a T. There is no information about the third base because there are no fragments of length 3. The fifth base can either be an A, a C or a T, and the sixth base can be a C or a G.

The *International Union of Pure and Applied Chemistry* (IUPAC) has developed a coding system (the **IUPAC code**) that assigns a unique uppercase letter to each combination of one, two, three and four different bases. For example, the combination "A or C or T" is represented by the letter H, the combination "C or G" by the letter S, and the combination "A or C or G or T" by the letter N. Based on the information in the sample file above, the DNA sequence can be reconstructed as CCNTHS. For this reconstruction we have also used the letter N at positions in the sequence for which no information can be derived from the text file.

Assignment

In this assignment you have to process two types of text files. An **IUPAC file** contains one line for each possible combination of one, two, three and four different base letters, that consists of a unique uppercase letter, followed by a space and the base letters of the combination in alphabetic order. All uppercase letters in the first column are different. A **fragments file** contains all fragments obtained after sequencing a DNA molecule using the Sanger method, with each fragment on a separate line. Your task:

- Write a function IUPAC that takes the location of an IUPAC file as its argument. The function must return a dictionary that maps all possible combinations of base letters in the given file onto the uppercase letter that is used to represent this combination. The uppercase letter can be found in the first column of the file.
- Write a function code that takes two arguments: *i*) a sequence of one or more base letters (a string, a list, a tuple or a set), and *ii*) a dictionary that maps all possible combinations of one, two, three and four different base letters (represented as a string containing the base letters in alphabetic order) onto the unique uppercase letter that represents the combination. The function must return the uppercase letter that corresponds to the combination of bases that is passed as the first argument to the function, taking into account the mapping determined by the dictionary that is passed as the second argument to the function.
- Write a function fragments that takes the location of a fragments file as its argument. The function must return a dictionary that maps each position in the DNA sequence for which information can be found in the given text file onto the set of bases on that position according to the given text file. The first position in the DNA sequence is position 1, the second position is position 2, and so on.
- Write a function sequence that takes the locations of a fragments file and an IUPAC file. The function must return the DNA sequence that can be reconstructed from the fragments in the given fragments file. The reconstruction must use the IUPAC code that is defined in the given IUPAC file. Positions in the DNA sequence for which no information is available in the fragments file have to be filled up by the letter N. The final position of the DNA sequence corresponds to the last position for which information can be derived from the fragments file.

Example

The following interactive session assumes that the text files fragments1.txt, fragments2.txt and IUPAC.txt are located in the current directory.

```
>>> codes = IUPAC('IUPAC.txt')
>>> codes
{'G': 'G', 'C': 'C', 'ACGT': 'N', 'ACT': 'H', 'ACG': 'V', 'AC': 'M', 'CT': 'Y', 'AT': 'W', 'AG': 'R', 'CG': 'S', 'T': 'T', 'A': 'A', 'GT': 'K', 'AGT': 'D', 'CGT': 'B'}
```

```
>>> code('ATC', codes)
'H'
>>> code(['G', 'C', 'G', 'C', 'C', 'C'], codes)
'S'
>>> code(('A', 'T', 'T', 'T', 'A', 'T', 'A', 'T', 'A', 'A'), codes)
'W'
>>> code({'G', 'A', 'T'}, codes)
'D'

>>> fragments('fragments1.txt')
{1: {'C'}, 2: {'C'}, 4: {'T'}, 5: {'C', 'A', 'T'}, 6: {'C', 'G'}}
>>> fragments('fragments2.txt')
{1: {'T'}, 2: {'G'}, 3: {'A'}, 4: {'C'}, 6: {'G'}}

>>> sequence('fragments1.txt', 'IUPAC.txt')
'CCNTHS'
>>> sequence('fragments2.txt', 'IUPAC.txt')
'TGACNG'
```

Frederick Sanger (1918-2013) was een Britse biochemicus die er als één van vier mensen in geslaagd is om twee Nobelprijzen in de wacht te slepen. In 1958 won hij een Nobelprijs voor zijn werk rond het bepalen van de structuur van eiwitten — in het bijzonder die van insuline — en in 1980 won hij samen met Walter Gilbert een Nobelprijs voor hun bijdrage aan het bepalen van de basevolgorde van een DNA sequentie (allebei Nobelprijzen voor de chemie).

Voor dit laatste ontwikkelden ze in 1977 een nieuwe methode die **Sangersequenering** genoemd wordt. Deze methode was gedurende 25 jaar de meest gebruikte sequeneringsmethode, maar ze is vandaag wat op het achterplan verdwenen door de opkomst van de **Next Generation Sequencing** methoden. Ze wordt echter nog vaak gebruikt voor kleinere projecten, voor de validatie van Next-Gen resultaten en om langere sequentiereads (> 500 basen) te bekomen. In onderstaande animatie wordt de werking van de methode in detail uitgelegd.

In deze opgave stellen we een **DNA sequentie** voor als een string die enkel bestaat uit de hoofdletters `A`, `C`, `G` en `T`, die de verschillende basen (of nucleotiden) voorstellen. Voorts moet je van de Sangermethode enkel weten dat het sequeneren van een DNA sequentie een groot aantal **fragmenten** oplevert, waarvan enkel de lengte en de laatste base gekend zijn. We zullen zo een fragment voorstellen door een string die bestaat uit nul of meer koppeltekens (`-`) die onbekende basen voorstellen aan het begin van de sequentie, gevolgd door één van de vier baseletters. Zo stelt onderstaande string bijvoorbeeld een fragment van lengte 6 voor, wat

aangeeft dat de zesde base van de DNA sequentie een G is.

```
-----G
```

Alle fragmenten die het sequeneren van een DNA sequentie oplevert, worden opgeslaan in een tekstbestand dat één fragment per regel bevat. Omdat de lengte van elk fragment op een stochastische manier bepaald wordt, is het mogelijk dat er geen fragmenten zijn met een bepaalde lengte of dat er verschillende fragmenten zijn met dezelfde lengte. Door fouten in het sequeneringsproces is het goed mogelijk dat er fragmenten zijn met dezelfde lengte, maar met een verschillende baseletter op het einde. Hierdoor is het niet altijd duidelijk welke base er op welke positie staat.

```
----T
C
-C
-----G
C
-----C
---T
----A
----C
-----G
```

Met de fragmenten uit bovenstaand voorbeeldbestand kunnen we bijvoorbeeld afleiden dat de eerste base van de sequentie een C is (bevestigd door twee fragmenten), de tweede base ook een C, en de vierde base een T. Over de derde base hebben we geen informatie, omdat er geen fragmenten zijn van lengte 3. De vijfde base kan een A, een C of een T zijn, en de zesde base kan een C of een G zijn.

De *International Union of Pure and Applied Chemistry* (IUPAC) heeft een code opgesteld (de **IUPAC code**) die aan alle verschillende combinaties van één, twee, drie of vier verschillende basen een verschillende hoofdletter uit het alfabet toekent. Zo wordt de combinatie "A of C of T" aangeduid met de letter H, de combinatie "C of G" met de letter S, en de combinatie "A of C of G of T" met de letter N. Op basis van de informatie in bovenstaand voorbeeldbestand kan de DNA sequentie dus gereconstrueerd worden als CCNTHS. Hierbij hebben we de letter N ook gebruikt voor posities in de sequentie waarover geen informatie gekend is.

Opgave

In deze opgave werken we met twee soorten tekstbestanden. Een **IUPAC bestand** bevat voor alle verschillende combinaties van één, twee, drie of vier verschillende baseletters een regel met daarop een unieke hoofdletter, gevolgd door een spatie en de baseletters van de combinatie in alfabetische volgorde. Alle hoofdletters in de eerste kolom van het bestand zijn verschillend. Een **fragmentenbestand** bevat alle fragmenten die de Sangersequenering van een DNA sequentie heeft opgeleverd, elk op een afzonderlijke regel. Gevraagd wordt:

- Schrijf een functie IUPAC waaraan de locatie van een IUPAC bestand moet doorgegeven worden. De functie moet een dictionary teruggeven die alle combinaties van baseletters uit het gegeven bestand afbeeldt op de hoofdletter waarmee de combinatie voorgesteld wordt. Deze hoofdletter is terug te vinden in de eerste kolom van het bestand.
- Schrijf een functie code waaraan twee argumenten moeten doorgegeven worden: *i*) een reeks van één of meer baseletters (een string, een lijst, een tuple of een verzameling), en *ii*) een dictionary die alle verschillende combinaties van één, twee, drie of vier verschillende baseletters (voorgesteld als een string met de baseletters in alfabetische volgorde) afbeeldt op de unieke hoofdletter waarmee de combinatie voorgesteld wordt. De functie moet de hoofdletter teruggeven die correspondeert met de combinatie van basen die als eerste argument aan de functie wordt doorgegeven, rekening houdend met de afbeelding die bepaald wordt door de dictionary die als tweede argument aan de functie wordt doorgegeven.
- Schrijf een functie fragmenten waaraan de locatie van een fragmentenbestand moet doorgegeven worden. De functie moet een dictionary teruggeven waarin alle posities van de gesequeneerde DNA sequentie waarover informatie te vinden is in het gegeven fragmentenbestand worden afgebeeld op de

verzameling van basen op die positie volgens het fragmentenbestand. De eerste positie van de sequentie is positie 1, de tweede positie 2, enzoverder.

- Schrijf een functie `sequentie` waaraan de locaties van een fragmentenbestand en een IUPAC bestand moeten doorgegeven worden. De functie moet de DNA sequentie teruggeven die kan gereconstrueerd worden uit de fragmenten in het gegeven fragmentenbestand. Hierbij moet gebruik gemaakt worden van de IUPAC codes die gedefinieerd worden in het gegeven IUPAC bestand. Indien het fragmentenbestand geen informatie bevat over een bepaalde positie in de DNA sequentie, dan moet die positie ingevuld worden met de letter `N`. De laatste positie van de sequentie correspondeert met de laatste positie waarover informatie kan teruggevonden worden in het fragmentenbestand.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat de tekstbestanden fragmenten1.txt, fragmenten2.txt en IUPAC.txt zich in de huidige directory bevinden.

```
>>> codes = IUPAC('IUPAC.txt')
>>> codes
{'G': 'G', 'C': 'C', 'ACGT': 'N', 'ACT': 'H', 'ACG': 'V', 'AC': 'M', 'CT': 'Y', 'AT': 'W', 'AG': 'R', 'CG': 'S', 'T': 'T', 'A': 'A', 'GT': 'K', 'AGT': 'D', 'CGT': 'B'}

>>> code('ATC', codes)
'H'
>>> code(['G', 'C', 'G', 'C', 'C', 'C'], codes)
'S'
>>> code(('A', 'T', 'T', 'T', 'A', 'T', 'A', 'T', 'A', 'A'), codes)
'W'
>>> code({'G', 'A', 'T'}, codes)
'D'

>>> fragmenten('fragmenten1.txt')
{1: {'C'}, 2: {'C'}, 4: {'T'}, 5: {'C', 'A', 'T'}, 6: {'C', 'G'}}
>>> fragmenten('fragmenten2.txt')
{1: {'T'}, 2: {'G'}, 3: {'A'}, 4: {'C'}, 6: {'G'}}

>>> sequentie('fragmenten1.txt', 'IUPAC.txt')
'CCNTHS'
>>> sequentie('fragmenten2.txt', 'IUPAC.txt')
'TGACNG'
```