Trabajo Práctico

Síntesis de Proteínas

Taller de Álgebra 1

2^{do} cuatrimestre 2016

Observaciones generales:

- El trabajo se debe realizar en grupos de tres personas.
- El código fuente debe enviarse por mail a la lista de docentes de la materia: `algebra1-doc@dc.uba.ar`, indicando los integrantes del grupo.
- El programa debe correr usando `ghci` que está instalado en los laboratorios del DC.
- Se evaluará la correctitud, claridad y modularidad del código entregado.
- La fecha límite de entrega es:
  - comisiones de los miércoles: martes 8 de noviembre a las 23:59 hs
  - comisiones de los viernes: jueves 10 de noviembre a las 23:59 hs

El DNA (ácido desoxirribonucleico) está compuesto por una secuencia de bases nucleotídicas unidas entre sí formando una estructura de hélice de doble cadena. A través de una serie de complejos procesos bioquímicos las secuencias nucleotídicas en el DNA de un organismo son traducidas a proteínas necesarias para la vida. El objetivo de este trabajo práctico es escribir una serie de funciones para poder determinar las proteinas codificadas en una cadena de DNA.

Las bases nucleotídicas que forman el DNA son ADENINA, CITOSINA, GUANINA y TIMINA (en adelante nos referiremos a ellas como `A`, `C`, `G` y `T` respectivamente). Estas bases se unen entre sí formando una cadena simple que corresponde a la mitad de la estructura de doble hélice. La otra mitad es una cadena similar, pero cada nucleótido es reemplazado por su nucleótido complementario. Las bases `A` y `T` son complementarias y también lo son `C` y `G`. Estas dos cadenas simples se unen entre sí por apareamiento de bases complementarias formando el DNA de doble cadena.

En general, un fragmento de DNA se describe simplemente con las bases que forman la cadena primaria. La cadena complementaria puede obtenerse escribiendo el complemento de bases de la primaria. Por ejemplo, la secuencia `TACTCGTAATTCACT` representa una cadena de DNA cuyo complemento es `ATGAGCATTAAGTGA`. Nótese que `A` siempre aparece apareada con `T`, y `C` con `G`.

A partir de la cadena primaria de DNA se genera una cadena de RNA (ácido ribonucleico) conocido como RNA mensajero (mRNA) en un proceso llamado transcripción. El mRNA transcripto es idéntico a la hebra complementaria con excepción de la TIMINA que es reemplazada por otro nucleótido llamado URACILO (`U`). Por ejemplo, la cadena de mRNA para el fragmento de DNA del ejemplo anterior es `AUGAGCAUUAAGUGA`.

Lo que determina la estructura de la proteína que va a ser sintetizada está codificado en la secuencia de bases del mRNA. El mRNA puede interpretarse como una secuencia de codones, dónde cada codón está compuesto exactamente por tres bases contiguas. El codón `AUG` marca el inicio de una secuencia proteica y cualquiera de los codones `UAA`, `UAG` o `UGA` marca su fin. El o los codones comprendidos entre los codones de inicio y terminación representan la secuencia de aminoácidos que conformarán la proteína. Por ejemplo, el codón `AGC` del mRNA corresponde al aminoácido Serina (Ser), `AUU` a Isoleucina (Ile) y `AAG` a Lysina (Lys). Así, la proteína formada por el mRNA del ejemplo anterior es Ser-Ile-Lys. La siguiente tabla muestra todas las traducciones de codones a aminoácidos:

Primera base del codón	Segunda base del codón				Tercera base del codón
	U	**C**	**A**	**G**	
U	Phe	Ser	Tyr	Cys	**U**
	Phe	Ser	Tyr	Cys	**C**
	Leu	Ser	—	—	**A**
	Leu	Ser	—	Trp	**G**
C	Leu	Pro	His	Arg	**U**
	Leu	Pro	His	Arg	**C**
	Leu	Pro	Gln	Arg	**A**
	Leu	Pro	Gln	Arg	**G**
A	Ile	Thr	Asn	Ser	**U**
	Ile	Thr	Asn	Ser	**C**
	Ile	Thr	Lys	Arg	**A**
	Met	Thr	Lys	Arg	**G**
G	Val	Ala	Asp	Gly	**U**
	Val	Ala	Asp	Gly	**C**
	Val	Ala	Glu	Gly	**A**
	Val	Ala	Glu	Gly	**G**

Nótese que la secuencia `AUG` corresponde tanto a una secuencia de inicio como al aminoácido Metionina (Met). Así, el primer `AUG` del mRNA será la secuencia de inicio, pero los subsiguientes serán traducidos a Metioninas (siempre y cuando no aparezca un codón de fin de secuencia entre ellos).

Cada cadena de DNA puede tratarse tanto de una hebra primaria como de su complementaria y en cada caso puede corresponder tanto a la secuencia *forward* como a la *reverse* (es decir, que debe ser leída tanto de izquierda a derecha como de derecha a izquierda). Además, los codones de inicio y terminación pueden no aparecer en los extremos de una secuencia. Por ejemplo, la proteína Ser-Ile-Lys puede corresponder a cualquiera de las secuencias (i) `ATACTCGTAATTCACTCC`, (ii) `TATGAGCATTAAGTGAGG` , (iii) `CCTCACTTAATGCTCATA` o (iv) `GGAGTGAATTACGAGTAT`:

(i)

$$\begin{aligned}\texttt{mRNA(ATACTCGTAATTCACTCC)} &= \big(\texttt{COMP(ATACTCGTAATTCACTCC)}\big)\big[\texttt{T:U}\big]\\ &= \texttt{(TATGAGCATTAAGTGAGG)[T:U]}\\ &= \texttt{U}\mathbf{\texttt{AUG}}\texttt{AGCAUUAAG}\mathbf{\texttt{UGA}}\texttt{GG}\end{aligned}$$

(ii)

$$\begin{aligned}\texttt{mRNA}\big(\texttt{COMP(TATGAGCATTAAGTGAGG)}\big) &= \texttt{mRNA(ATACTCGTAATTCACTCC)}\\ &= \big(\texttt{COMP(ATACTCGTAATTCACTCC)}\big)\big[\texttt{T:U}\big]\\ &= \texttt{(TATGAGCATTAAGTGAGG)[T:U]}\\ &= \texttt{U}\mathbf{\texttt{AUG}}\texttt{AGCAUUAAG}\mathbf{\texttt{UGA}}\texttt{GG}\end{aligned}$$

(*iii*)

```
mRNA(REV(CCTCACTTAATGCTCATA)) = mRNA(ATACTCGTAATTCACTCC)
                              = (COMP(ATACTCGTAATTCACTCC))[T:U]
                              = (TATGAGCATTAAGTGAGG)[T:U]
                              = UAUGAGCAUUAAGUGAGG
```

(*iv*)

```
mRNA(REV(COMP(GGAGTGAATTACGAGTAT))) = mRNA(REV(CCTCACTTAATGCTCATA))
                                    = mRNA(ATACTCGTAATTCACTCC)
                                    = (COMP(ATACTCGTAATTCACTCC))[T:U]
                                    = (TATGAGCATTAAGTGAGG)[T:U]
                                    = UAUGAGCAUUAAGUGAGG
```

Para poder obtener las proteinas codificadas en una secuencia de DNA se cuenta con los siguientes tipos de datos:

- `data BaseNucleotidica = A | C | G | T | U deriving Show`
- `type CadenaDNA = [BaseNucleotidica]`
- `type CadenaRNA = [BaseNucleotidica]`
- `type Codon = (BaseNucleotidica, BaseNucleotidica, BaseNucleotidica)`
- `data Aminoacido = Phe | Ser | Tyr | Cys | Leu | Trp | Pro | His | Arg | Gln | Ile | Thr | Asn | Lys | Met | Val | Ala | Asp | Gly | Glu deriving Show`
- `type Proteina = [Aminoacido]`

Se cuenta también con una función que dado un Codón devuelve el aminoácido correspondiente:

- `traducirCodonAAminoacido:: Codon -> Aminoacido`

Se pide implementar las siguientes funciones:

- `complementarBase :: BaseNucleotidica -> BaseNucleotidica`
  Dada una base nucleotídica devuelve su base complementaria.
- `complementarCadenaDNA :: CadenaDNA -> CadenaDNA`
  Dada una cadena de DNA devuelve su cadena complementaria.
- `obtenerCadenaReverseDNA :: CadenaDNA -> CadenaDNA`
  Dada una cadena de DNA devuelve su cadena *reverse*.
- `transcribir :: CadenaDNA -> CadenaRNA`
  Dada una cadena de DNA devuelve su transcripción a RNA.
- `obtenerProteinas :: CadenaDNA -> [Proteina]`
  Dada una cadena de DNA devuelve una lista de las proteinas codificandas. En caso de que una secuencia codifique más de una proteína, todas deben estar presentes en la lista que devuelva la función. El orden en que deben aparecer es: 1) las codificadas por la secuencia original, 2) por la secuencia reversa, 3) por la secuencia complementaria, 4) por la secuencia complementaria reversa. Algunas secuencias válidas de DNA no codifican proteínas; para esos casos la función debe devolver la lista vacía.

A modo de ejemplo pueden observarse la siguiente entrada y la salida esperada:
`obtenerProteinas [A,T,A,C,T,C,G,T,A,A,T,T,C,A,C,T,C,C]` ⇝ `[[Ser,Ile,Lys]]`
`obtenerProteinas [T,T,A,A,T,A,C,G,A,C,A,T,A,A,T,T,A,T]` ⇝ `[[Leu,Tyr],[Ser,Tyr]]`
`obtenerProteinas [G,C,C,T,T,G,A,T,A,T,G,G,A,G,A,A,C,T,C,A,T,T]` ⇝ `[]`

Para la implementación de la función `obtenerProteinas` se sugiere utilizar funciones auxiliares. Algunas que podrían ser de útilidad son:

- `obtenerProteinaDeRNA :: CadenaRNA -> [Proteina]`
  Devuelve las secuencia proteica codificada por una cadena de RNA dada.
- `sincronizaConCodonDeFin :: CadenaRNA -> Bool`
  Devuleve `True` si existe un codón de fin a una distancia múltiplo de 3 del inicio de una cadena de RNA dada y `False` en caso contrario.

En el archivo `tp.hs` se encuentran:

- las definiciones de los tipos `BaseNucleotidica`, `CadenaDNA`, `CadenaRNA`, `Codon`, `Aminoacido` y `Proteina`;
- la definición de la función `traducirCodonAAminoacido:: Codon -> Aminoacido`;
- los encabezados de las funciones que deben implementarse.