CORSO INTEGRATO DI INFORMATICA E BIOINFORMATICA
per il CLT in BIOLOGIA MOLECOLARE

III Esercitazione di Bioinformatica
"Python for bioinformatics"

L'obbiettivo dell'esercitazione è familiarizzare con l'uso del linguaggio di programmazione Python in ambito bioinformatico.
Scaricheremo un file contenente tutte le sequenze di precursori di microRNA conosciuti (per varie specie) e queste verranno "lette" attraverso Python, filtrate per ottenere solo i precursori umani, e calcoleremo delle statistiche descrittive sulle lunghezze delle sequenze.

I microRNA (miRNA) sono dei piccoli trascritti non codificanti di circa 22 nucleotidi che possono essere implicati nei meccanismi regolazione dell'espressione genica. I miRNA sono generati a partire da dei trascritti primari più lunghi, chiamati precursori dei miRNA, che si ripiegano su se stessi formando una struttura simile ad una forcina ("hairpin"). I precursori vengono processati da degli enzimi che ne tagliano le sequenze per formare i miRNA maturi.

Uso della Shell di Linux (preparazione dei dati):

Scaricare il file compresso contenente i dati per l'esercitazione dal seguente link:
ftp://mirbase.org/pub/mirbase/CURRENT/hairpin.fa.gz

Aprire un terminale dei comandi (cercare "terminale" tra i programmi disponibili)
Decomprimere il file nella vostra home directory (~/nomeutente); la ~ (tilde) è il carattere che indica la home directory dell'utente (/home/username):

```
gunzip hairpin.fa.gz
```

Dare un'occhiata ai file contenuti nella home directory:

```
ls
```

Ispezionare il file appena creato con:

```
less hairpin.fa
```
premere q per tornare alla shell.

Verificare il numero di sequenze nel file creato usando l'opzione -c del comando grep:

```
grep -c ">"  hairpin.fa
```

Se vogliamo ottenere informazioni sul comando grep:

```
man grep
```

oppure

```
grep -h
```

Programmazione in Python (processamento dei dati)

Dal terminale avviare l'interprete interattivo di Python (`python, ipython, idle`).

Importare la libreria Biopython (http://biopython.org) contenente routines di tipo bioinformatico utili alla manipolazione di file FASTA.
Biopython non è una libreria standard di Python e se non presente deve essere installata separatamente.
La libreria è già installata nelle macchine del laboratorio.
Caricheremo solo un modulo (SeqIO) della libreria che utilizzeremo per maneggiare i file FASTA.

```
>>> from Bio import SeqIO
```

Maggiori informazioni sul modulo SeqIO: http://biopython.org/wiki/SeqIO e
http://biopython.org/DIST/docs/api/Bio.SeqIO-module.html

Apriamo il file contenente tutte le sequenze:

```
>>> fileHandler = open("hairpin.fa")
```

Leggiamo le sequenze in formato FASTA utilizzando il metodo to_dict del modulo SeqIO

```
>>> recordDict = SeqIO.to_dict(SeqIO.parse(fileHandler, "fasta"))
```

Visualizziamo le chiavi del dizionario (i nomi delle sequenze fasta):

```
>>> len(recordDict.keys())
```

Visualizziamo le prime 10 chiavi (nomi dei precursori):

```
>>> recordDict.keys()[0:10]
```

(in Python gli indici cominciano da 0. La posizione '0' è inclusa, la posizione '10' è esclusa)

Visualizziamo il trascritto con miRBase ID hsa-let-7a-1

```
>>> seqObject = recordDict.get("hsa-let-7a-1")
```

Visualizziamo la sequenza nucleotidica accedendo all'attributo *seq* dell'oggetto seqObject:

```
>>> print seqObject.seq
```

Visualizziamo l'identificativo del trascritto:

```
>>> print seqObject.name
```

Qual'è la lunghezza del trascritto hsa-let-7a-1?

```
>>> len(seqObject)
>>> len(seqObject.seq)
```

Il file hairpin.fa contiene sequenze di varie specie. Noi vogliamo estrarre solo quelle umane, che sono identificabili dal nome che comincia per “hsa”, ad es. come prima “hsa-let-7a-1”.
N.B: fare attenzione all'indentazione del codice.

```
>>>  human_precursors = {}
     for n in recordDict.keys():
       if n.startswith(“hsa”):
         human_precursors[n] =  recordDict[n]
```

Vogliamo ora calcolare la lunghezza media delle nostre sequenze.
La variabile `hsaLenghtList` (una lista in python) conterrà le lunghezze delle sequenze considerate.

```
>>> hsaLenghtList = []

>>> for record in human_precursors.keys():
      length = len(human_precursors[record])
      hsaLenghtList.append(length)
```

Visualiziamo la lista contenente le lunghezze dei primi 10 trascritti

```
>>> hsaLenghtList[0:10]
```

Da quanti nucleotidi è composto il trascritto più lungo?

```
>>> max(hsaLenghtList)
```

...e quello più corto?

```
>>> min(hsaLenghtList)
```

Calcoliamo ora la media delle lunghezze dei trascritti

```
>>> 1.0 * sum(hsaLenghtList) / len(hsaLenghtList)
```

Filtriamo ora le sequenze più corte di 60 nucleotidi

```
>>> filteredSequences = []
>>> for record in recordDict.keys():
       sequence = recordDict[record]
      if len(sequence) > 60:
        filteredSequences.append(sequence)
```

Quanti trascritti sono più lunghi di 60 nt?

```
>>> len(filteredSequences)
```

Q: Quanti trascritti sono più lunghi di 60 nt e più corti di 100?

Apriamo un file in scrittura (“w”) dove salveremo i trascritti più lunghi di 60 basi

```
>>> outputFile = open("my_sequences.fasta", "w")
```

Scriviamo le sequenze in formato FASTA

```
>>> SeqIO.write(filteredSequences, outputFile, "fasta")
>>> outputFile.close()

>>> quit()
```

Il prompt torna al terminale. Verificate l'output del vostro programma:

```
less my_sequences.fasta
```

Esercizi aggiuntivi
Q: Quanti precursori umani sono annotati? (Suggerimento: contare le chiavi del dizionario appena creato)
Q: confrontare la lunghezza media dei precursori umani con la lunghezza media di tutti i precursori annotati in miRBase (hairpin.fa)
Q: qual'è il nome del precursore umano più lungo?
Q: qual'è il nome del precursore più lungo annotato in miRBase? Che specie è?