III. Formate de fişiere pentru secvenţe biologice. Baze de date biologice

1. Noţiuni introductive

Bioinformatica se ocupă cu prelucrarea cantităţii enorme de date biologice. În urma secvenţierii unui genom sau a unei proteine, secvenţa respectivă, alcatuită din cele patru nucleotide, în cazul unui genom, sau din aminoacizi, în cazul unei proteine, este stocată într-o bază de date publică pentru a putea fi studiată de numeroşi cercetători.

Există diferite programe de analiză şi prelucrare a secvenţelor biologice. Astfel, pentru ca datele referitoare la o secvenţă să poată circula între diferite aplicaţii informatice, au fost definite mai multe formate de fişiere utilizate în bioinformatică, precum:

FASTA
ASN. 1
GenBank
PDB
XML

Pentru a putea publica descoperirea unei noi gene sau a unui fragment genomic, trebuie trimisă secvenţa către bazele de date biologice publice pentru a putea obţine un număr de acces („accession number"). Numărul de acces este unic pentru fiecare secvenţă trimisă, pentru a putea fi identificată printre numeroasele secvenţe descoperite zilnic de cercetători din toată lumea.

2. Baza de date biologice

NCBI (National Center for Biotechnology Information) pune la dispoziţie baze de date biologice, împreună cu un SGBD (care include motorul de căutare *Entrez* şi diferite programe şi utilitare pentru gestiunea şi prelucrarea înregistrărilor): http://www.ncbi.nlm.nih.gov

Secvenţele de nucleotide (ADN) sau cele de aminoacizi (proteine) au un număr de acces unic într-o bază de date. Motorul de căutare *Entrez* caută informaţii disponibile în mai multe baze de date, astfel se poate obţine aceeaşi secvenţă de mai multe ori cu numere de acces diferite. NCBI pune la dispoziţie o bază de date *RefSeq* în care sunt incluse informaţii non-redundante.
Secvenţele din *RefSeq* au numere de acces de forma: NT_123456 – regiuni de cromozomi; NC_123456 – cromozom; NP_123456 – proteina; NG_123456 – gena;

Genomurile secvenţiate până la ora actuală pot fi găsite pe NCBI: http://www.ncbi.nlm.nih.gov/sites/genome

a. Să se acceseze următoarea adresă de la NCBI (National Center for Biotechnology Information): http://www.ncbi.nlm.nih.gov/guide/genomes-maps/
b. Să se acceseze baza de date *Genome*.
c. Să se acceseze un grup de organisme şi, apoi, un număr de acces (*accession number)*
d. pentru un genom cromozomial al unui organism, de exemplu: NC_005562 .
e. Să se acceseze baza de date *RefSeq*: NC_005562.
f. Să se salveze informaţia găsită, în formatele: FASTA, ASN. 1, GenBank.
g. Să se salveze doua secvente genomice diferite (unul procariot si unul eucariot) în fiecare din cele trei formate.

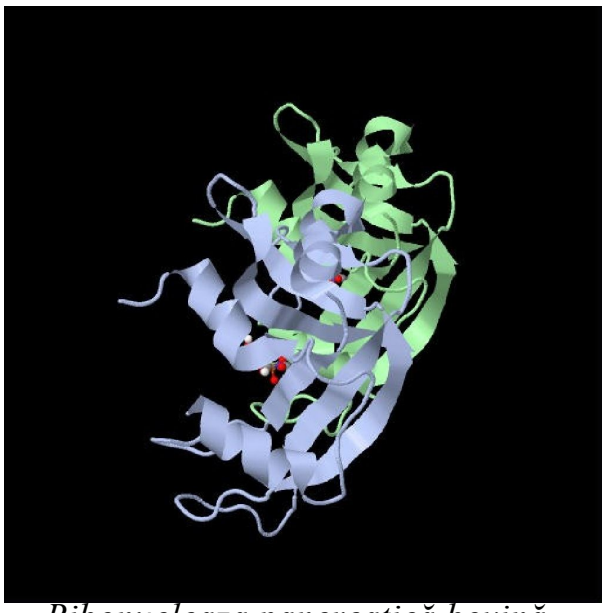
3. Formatul FASTA (Fast Alignment)

FASTA reprezintă un format de fişier foarte utilizat pentru date de tip secvenţă, precum lanţuri de nucleotide sau de aminoacizi. Formatul este foarte simplu:

- pe prima linie conţine un header, caracterul ">", urmat de informaţii despre secvenţă (numărul de identificare, specia);
- următoarele linii reprezintă secvenţa de aminoacizi sau nucleotide.
- liniile au maxim 120 de caractere, dar în general nu depăşesc 80. Fiecare linie se termină cu caracterul de sfârşit de linie "\n".

De exemplu, un fisier FASTA care descrie secventa unei proteine, arata astfel:

>sp|P61823|RNAS1_BOVIN Ribonuclease pancreatic OS=Bos taurus GN=RNASE1
MALKSLVLLSLLVLVLLLVRVQPSLGKETAAAKFERQHMDSSTSAASS
SNYCNQMMKSRNL
TKDRCKPVNTFVHESLADVQAVCSQKNVACKNGQTNCYQSYSTMSIT
DCRETGSSKYPNC AYKTTQANKHIIVACEGNPYVPVHFDASV



Ribonucleaza pancreatică bovină

Literele reprezintă aminoacizii proteinei secvenţiate. În acest caz este reprezentată enzima ribonucleaza pancreatică bovină. Ribonucleazele sunt proteine enzimatice care degradează moleculele de ARN care nu mai sunt utile în cadrul celulei.

Să se descarce o secvenţă de nucleotide şi o secvenţă proteică de pe www.ncbi.nlm.nih.gov în format FASTA. **Să se implementeze în Python următoarele cerinţe:**

a. Să se citească secvenţa de nucleotide, linie cu linie, şi să se afişeze într-un fişier text .fasta, concomitent cu citirea.
b. Să se concateneze două fişiere FASTA într-un singur fişier FASTA.
c. Sa se citeasca secvenţa proteinei si să se memoreze într-o variabilă, sub forma unui şir de caractere fara spatii. Să se afişeze variabila obţinută.

4. Formatul ASN. 1

Seq-entry ::= set { informatii si secvente
}

1. conţine informaţii despre genomul secvenţiat;
2. conţine informaţii despre autorii care au participat la secvenţiere;
3. secvenţa de ADN apare imediat după apariţia şirului **ncbi2na '** ;
4. secvenţa de ADN se termină la întâlnirea caracterului **'** ;
5. pot exista mai multe regiuni de ADN care împreună alcatuiesc cromozomul respectiv, fiecare regiune de ADN fiind semnalată de prezenţa campului ncbi2na;
6. secvenţa de ADN este afişată într-un mod mai economic (fiind redusă la jumătate ca dimesiune), astfel:
   - cele 4 nucleotide din ADN sunt codificate de 2 biţi, astfel:
     A – 00
     C – 01
     G – 10
     T – 11
   - se afişează câte 2 nucleotide (4 biţi) ca un număr Hexazecimal:
     0000 – 0
     0001 – 1
     0010 – 2
     0011 – 3
     0100 – 4
     0101 – 5
     0110 – 6
     0111 – 7
     1000 – 8
     1001 – 9

1010 – A
1011 – B
1100 – C
1101 – D
1110 – E
1111 – F

De exemplu, următoarea secvenţă de nucleotide devine următorul şir în format ASN. 1: ATTCGAAC → 3C81

a. **Să se citească dintr-un fişier format ASN. 1** prima secvenţă de ADN care apare şi să se scrie într-un alt fişier în format FASTA.

Observaţii:

- Pentru citire se utilizează doar funcţia *readline()*. Citirea se face linie cu linie, concomitent cu scrierea. Nu se citeşte toată secvenţa de ADN într-o variabilă, deoarece este prea mare.
- Nu se utilizează cicluri *for/while* pentru efectuarea conversiei, ci se utilizează funcţiile:
  - *map(f, data)* – aplică funcţia *f* pe elementele din *data* (*data* poate fi lista, sir de caractere, etc.) şi întoarce rezultatele într-o lista
  - *sir.join(lista)* – concatenează stringurile din lista *lista* şi pune între ele şirul *sir* pentru care este apelată – va întoarce şirul obţinut în urma concatenării.

  Exemplu:

```
a= ['1','2','3','4','5','6']
print '&'.join(a)
 print ''.join(a)
```

- Pentru citirea primei secvenţe de ADN (prima regiune de ADN) **nu** se utilizează cicluri imbricate. Pentru rezolvarea cerintei **a.** nu se utilizeaza cicluri *for/while* imbricate.

b. **Sa se gaseasca un fisier ASN.1 care ontine mai multe regiuni**. Să se citească toate regiunile de ADN din fişierul în format ASN. 1 şi să se scrie un fişier multisecvenţă FASTA, de forma:

>Regiunea1
ATTTGGG.....
.....

>Regiunea2
CCCTTAA.....
.....

Pentru rezolvarea cerintei **b.** se utilizeaza maxim 2 cicluri *for/while* imbricate.

Observaţii:

- Funcţia *sir.find('xyz')* întoarce indicele de început al subşirului 'xyz' în şirul *sir*.
- Să se utilizeze dicţionarul *code* şi funcţia *conv* pentru conversie şi să se implementeze funcţia: *DNAfromASN1(fin, fout)* care primeşte ca intrare fişierul ASN. 1 din care va citi şi un fişier de ieşire în care va scrie.

```
code = {
'0' : 'AA', '1' : 'AC', '2' : 'AG', '3' : 'AT', '4' : 'CA', '5' : 'CC', '6' : 'CG', '7' : 'CT', '8'
: 'GA', '9' : 'GC', 'A' : 'GG', 'B' : 'GT', 'C' : 'TA', 'D' : 'TC', 'E' : 'TG', 'F' : 'TT'
}

def conv(x):
  return code[x]


def DNAfromASN1(fin, fout):
  pass

if __name__ == '__main__':
  DNAfromASN1('seq.asn1', 'seq.fasta')
```