Laborator 6 – Bioinformatică

I. Alinierea de secvenţe

1. Definirea problemei

Alinierea de secvente reprezintă găsirea subsecvenţelor similare dintre două sau mai multe secvenţe. Alinierile se realizează utilizând tehnici de potrivire de şiruri de caractere ("*string matching*"). Potrivirea de şiruri de caractere nu este o problemă simplă şi este utilizată foarte des în bioinformatică. Astfel, găsirea algoritmilor eficienţi şi optimi pentru alinieri este foarte importantă.

Există algoritmi complecşi pentru alinieri, bazaţi pe tehnici de programare dinamică sau alte tehnici dificile de programare. Dar algoritmii utilizaţi diferă în funcţie de tipurile de secvenţe care trebuie aliniate şi de tipurile de alinieri.

Aplicaţiile alinierilor în bioinformatică sunt numeroase, precum:

1. determinarea relaţiilor filogenetice între specii (clasificarea evolutivă);
2. clasificarea proteinelor şi stabilirea anumitor funcţii impuse de structură;
3. căutarea anumitor gene în cadrul unei secvenţe de ADN;
4. compararea anumitor secvenţe de ADN (criminalistică);
5. studierea materialelor genetice virale sau bacteriene în scopul găsirii de medicamente adecvate.

Există mai multe tipuri de alinieri:

1. Aliniere globală

Reprezintă alinierea completă a unei secvenţe cu o altă secvenţă, de exemplu:

```
and.--e-.can.sequence--.very.well.now
||||   |  |||| |    |    ||||||     |
and.they.can.lear-n.too.very.soo--n--
```

Există mai multe modalităţi de potrivire globală a celor două şiruri. Se urmăreşte găsirea celor mai bune potriviri, cu cât mai puţine goluri, ”*gaps*” (–).

2. Aliniere locală

Reprezintă găsirea unei regiuni dintr-o secvenţă care se potriveşte cu o regiune din altă secvenţă, de exemplu:

```
 we.try.to.help.people
    |||||||| |
you.try.to.learn.bioinformatics
```

Exemplu de alinieri diferite pentru două şiruri:

```
Globală:
GTGTAKIKKAVAV
G--TAK-KKA-AV

Locală:
GTGTAKIKK-AVAV
--GTAK-KKAAV--
```

3. Găsire de secvenţe scurte într-o secvenţă lungă:

```
          match
          |||||
we.shall.match.the.words
```

4. Aliniere multiplă între mai multe secvenţe

Scorul unei alinieri

Aşa cum se poate observa, există mai multe alinieri posibile pentru aceleaşi secvenţe. Scorul unei alinieri reprezintă o valoare care estimează cât de bună este o aliniere. Pentru a alege cea mai bună aliniere trebuie să se ţină cont de semnificaţia biologică a secvenţelor şi de proprietaţile structurale ale elementelor componente. Astfel, în urma datelor experimentale şi statistice, s-au definit matrici de scoruri pentru alinieri. În calcularea scorului final al unei alinieri, potrivirile de caractere pot avea ponderi diferite sau egale, în funcţie de problemă.

Exemplu:

Un exemplu de matrice de scoruri pentru alinierea de secvenţe de nucleotide este următoarea:

$$N = \begin{array}{c|ccccc} & A & T & C & G & - \\ A & 1 & -1 & -1 & -1 & -5 \\ T & -1 & 1 & -1 & -1 & -5 \\ C & -1 & -1 & 1 & -1 & -5 \\ G & -1 & -1 & -1 & 1 & -5 \\ - & -5 & -5 & -5 & -5 & 0 \end{array}$$

Scorul unei alinieri de două secvenţe se poate calcula adunând valorile corespunzatoare din matrice pentru fiecare două nucleotide comparate.

Astfel, pentru următoarea aliniere:

```
ATTTCAGG----ATTT
-T--CAGGGCCTATTT
```

scorul va fi:

```
S = N(A, -) + N(T, T) + N(T, -) + N(T, -) + N(C, C) + N(A, A) + N(G, G) +
N(G, G) + N(-, G) + N(-, C) + N(-, C) + N(-, T) + N(A, A) + N(T, T) + N(T,
```

T) + N(T, T) = – 5 + 1 – 5 – 5 + 1 + 1 + 1 + 1 – 5 – 5 – 5 – 5 + 1 + 1 + 1 + 1 = 9 – 35 = –26

Un exemplu de matrice utilizată pentru alinierea de secvenţe de aminoacizi este matricea PAM (Point Accepted Mutation) care a fost calculată observând diferenţe dintre proteine înrudite. Alt exemplu este reprezentat de matricile BLOSUM (BLOcks of amino acid SUbstitution Matrix) care se utilizează, de asemenea, pentru alinierea secvenţelor de aminoacizi. Probabilităţile utilizate în calcularea matricilor BLOSUM sunt calculate pe baza blocurilor de secvenţe conservate găsite în alinieri multiple de proteine. Cea mai utilizată matrice din seria BLOSUM este BLOSUM62.

Utilitare pentru alinierea de secvenţe

1. *CLUSTAL-W* este un program care realizează alinieri globale între două sau mai multe secvenţe de aminoacizi sau nucleotide. Se utilizează pentru compararea anumitor secvenţe.

2. *BLAST* (Basic Local Alignment Search Tool) este un program care realizează tot alinieri, dar se utilizează pentru alinieri locale, fiind foarte util pentru căutarea de secvenţe de ADN sau proteine în bazele de date. Astfel, se pot determina funcţiile anumitor secvenţe pe baza similitudinii cu alte secvenţe cunoscute şi, de asemenea, se pot studia relaţii filogenetice între diferite specii.

2. Aplicaţii. CLUSTAL-W şi BLAST

1. Să se acceseze *CLUSTAL-W* la următoarea adresă:

   http://www.ebi.ac.uk/Tools/clustalw2/index.html

   a. Să se testeze gradul de asemănare între ribonucleazele pancreatice din fişierele FASTA pe baza scorului afişat de program.

   b. Cât este scorul pentru alinierile ribonucleazei de la om cu cea de la cimpanzeu? Dar de la om cu cea de la cangur?

   c. Să se calculeze scorurile obţinute în urma unei alinieri multisecvenţă (cel puţin 3 ribonucleaze diferite).

2. Să se acceseze următoarea adresă:

   http://www.uniprot.org/

   Să se caute secvenţele în format FASTA pentru hemoglobină (subunitatea alpha) de la om şi cea de la şoarece şi să se compare utilizând CLUSTAL-W.

3. Să se caute pe *UniProt* secvenţa proteinei cu numărul de identificare *P26367*. Este o proteină codificată de gena *PAX-6* care controlează dezvoltarea ochilor la numeroase specii de organisme.

a. Să se salveze secvenţa proteinei în format FASTA.

b. Să se acceseze *PSI-BLAST* şi să se introducă secvenţa proteinei:

http://blast.ncbi.nlm.nih.gov/

Programul va găsi toate organismele din baza de date la care gena *PAX-6* este prezentă. Căutarea se realizează în baza de date cu proteine şi se afişează alinierile cu scorul cel mai bun.

3. Algoritmi pentru alinierea de secvenţe

A. Un algoritm euristic pentru alinierea a două secvenţe

Algoritmul de mai jos rezolvă problema alinierii globale a doua secvenţe printr-o metodă "greedy" euristică. Un algoritm "greedy" urmăreşte rezolvarea problemei prin alegerea la fiecare pas a optimului local, sperând că în final se ajunge la optimul global.

Acest algoritm nu este simetric, uneori întoarce alinieri diferite în funcţie de ordinea în care sunt date secvenţele ca parametri. Apelul *align_asym(seq1, seq2)* poate întoarce alt rezultat decât apelul *align_asym(seq2, seq1)*.

```
def align_asym(seq1, seq2):
    i = 0
    while i < len(seq1) and i < len(seq2):
        if seq1[i] == seq2[i]:
            i += 1
        else:
            k = i
            min_idx1 = len(seq1)
            min_idx2 = len(seq2)
            while k < min_idx2 and k < len(seq1):
                idx = seq2.find(seq1[k], i)
                if idx >= 0 and min_idx2 > idx:
                    min_idx1 = k
                    min_idx2 = idx
                k += 1
            if min_idx1 > min_idx2:
                seq2 = seq2[:min_idx2] \
                       + '-' * (min_idx1 - min_idx2) \
                       + seq2[min_idx2:]
            elif min_idx1 < min_idx2:
                seq1 = seq1[:min_idx1] \
                       + '-' * (min_idx2 - min_idx1) \
                       + seq1[min_idx1:]

            i = max(min_idx1, min_idx2) + 1

    i = len(seq1)
    j = len(seq2)
    if i < j:
        seq1 = seq1 + '-' * (j - i)
    else:
        seq2 = seq2 + '-' * (i - j)

    return (seq1, seq2)
```

Se exemplifică funcţionarea algoritmului pe două secvenţe:

```
secv1: RQASPQT
secv2: RTSPTA
```

Algoritmul funcţionează astfel:

i = contor de parcurgere secvenţe

cât timp nu s-a ajuns la sfârşitul nici unei secvenţe

dacă caracterele curente din cele două secvenţe sunt egale, atunci:

i += 1

altfel:

k ← *i*;

*min_idx*1 = *lungimea secv*1;

*min_idx*2 = *lungimea secv*2;

se parcurge *sec*1 de la *i* către sfârşit cu contorul *k*, fără ca valoarea *k* să depăşească *min_idx*2

se caută caracterul de pe poziţia *k* din *secv*1 în *secv*2

dacă îl găseşte şi indexul din *secv*2 (*idx*) este mai mic decât *min_idx*2 (cel salvat pentru perechea anterioară) atunci:

*min_idx*1 ← *k*;

*min_idx*2 ← *idx*;

(salvează indicii noii perechi)

se aliniază caracterele prin introducerea de caractere "–";

dacă una din secvenţe este mai scurtă decât cealălaltă, la sfârşitul alinierii, se completează secvenţa mai scurtă cu caractere "–".

În urma alinierii celor două secvenţe, se va obţine următorul rezultat:

```
RQASPQT-
RT-SP-TA
```

Alte alinieri posibile mai proaste ar putea fi:

```
RQASPQT----
R-----TSPTA

RQ---ASPQT
RTSPTA----
```

Aplicaţii. Algoritm euristic pentru alinierea a două secvenţe

1. Să se scrie o funcţie care afişează un şir de caractere format din "+" şi "-", care arată dacă acele caractere de la poziţia *i* din două secvenţe deja aliniate coincid sau nu. De exemplu:

   ```
   RQASPQT-
   RT-SP-TA
   +--++-+-
   ```

2. Să se testeze algoritmul "*greedy*" de aliniere pe secvenţele a două ribonucleaze pancreatice de la specii diferite, citite din fişiere în format FASTA. De exemplu, să se compare cea de la om cu cea de la cimpanzeu si cea de la om cu cea de la balenă. Ce se observă, care sunt mai asemanatoare?

3. Algoritmul nu găseşte întotdeauna cea mai bună aliniere. Să se găsească un exemplu de două şiruri pentru care se observă în mod evident că alinierea obţinută nu este cea mai bună. Să se explice de ce algoritmul "*greedy*" nu găseşte alinierea cea mai bună.

4. Să se scrie o funcţie, *simple_score(seq1, seq2)*, care calculează scorul unei alinieri globale, calculat ca procentul de caractere aliniate corect din lungimea secvenţelor aliniate: (*nr_caractere_aliniate* / *lungime secv*) * 100.

5. Să se scrie o funcţie, *align(seq1, seq2, score)*, care:
   - primeşte ca parametri două secvenţe nealiniate şi o funcţie care calculează scorul;
   - apelează *align_asym(seq1, seq2)* şi *align_asym(seq2, seq1)* şi decide care aliniere este mai bună (are scor mai mare);
   - funcţia întoarce cele două secvenţe obţinute în urma alinierii mai bune.
6. Să se exemplifice funcţionarea corectă a funcţiilor implementate, pe secvenţe sugestive din fişiere FASTA. Să se compare pe rând ribonucleaza pancreatică de la om cu cele de la alte specii, salvate în fişiere FASTA. Cum diferă scorurile? Ce se observă?