

Elemente de biologie moleculară

1. Enunțați “Dogma Centrală” a biologiei moleculare.
2. Sfârșitul genei de β -hemoglobină la om este următoarea secvență:

1	4	7	10	13	16	19
<i>CTG</i>	<i>GCC</i>	<i>CAC</i>	<i>AAG</i>	<i>TAT</i>	<i>CAC</i>	<i>TAA</i>

- a. Care este secvența de aminoacizi în care se traduce această secvență?
 - b. O mutație ‘silent’ este o mutație în secvența de nucleotide care lasă neschimbată secvența corespunzătoare de aminoacizi. Indicați o mutație silent dată de schimbarea unei singure nucleotide în secvența din enunț.
 - c. Indicați o mutație de o singură nucleotidă care ar duce la trunchierea prematură a proteinei produsă de gena din enunț.
 - d. Indicați o mutație de o singură nucleotidă care produce extensia proteinei.
3. Ce este o insulă CpG și care este legătura cu problema identificării genelor?
 4. Diagrama de mai jos ilustrează fluxul de informație genetică din celulă. Plasați pe această diagramă cât mai multe din cuvintele următoare (nu neaparat toate): codon, transcriere, ARN, tranziție, ADN, translație (traducere), nucleotidă, evoluție, proteină, amino-acid, replicare, recombinare.

GAA CAA GGT CGA CAT TTA ATG ATG

↓

CUU GUU CCA GCU GUA AAU UAC UAC

↓

E Q G R H L M M

Addenda: Codul genetic

		Second letter			
		U	C	A	G
First letter	U	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">UUU</div> <div style="border: 1px solid black; padding: 2px;">UUC</div> </div> <i>Phenil-alanine</i> <div style="border: 1px solid black; padding: 2px;">UUA</div> <div style="border: 1px solid black; padding: 2px;">UUG</div> <i>Leucine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">UCU</div> <div style="border: 1px solid black; padding: 2px;">UCC</div> <div style="border: 1px solid black; padding: 2px;">UCA</div> <div style="border: 1px solid black; padding: 2px;">UCG</div> </div> <i>Serine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">UAU</div> <div style="border: 1px solid black; padding: 2px;">UAC</div> </div> <i>Thyrosine</i> <div style="border: 1px solid black; padding: 2px; background-color: red;">UAA</div> <div style="border: 1px solid black; padding: 2px; background-color: red;">UAG</div> <i>STOP codon</i> <i>STOP codon</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">UGU</div> <div style="border: 1px solid black; padding: 2px;">UGC</div> </div> <i>Cysteine</i> <div style="border: 1px solid black; padding: 2px; background-color: red;">UGG</div> <i>STOP codon</i> <div style="border: 1px solid black; padding: 2px;">UGG</div> <i>Tryptophan</i>
	C	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">CUU</div> <div style="border: 1px solid black; padding: 2px;">CUC</div> <div style="border: 1px solid black; padding: 2px;">CUA</div> <div style="border: 1px solid black; padding: 2px;">CUG</div> </div> <i>Leucine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">CCU</div> <div style="border: 1px solid black; padding: 2px;">CCC</div> <div style="border: 1px solid black; padding: 2px;">CCA</div> <div style="border: 1px solid black; padding: 2px;">CCG</div> </div> <i>Proline</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">CAU</div> <div style="border: 1px solid black; padding: 2px;">CAC</div> </div> <i>Histidine</i> <div style="border: 1px solid black; padding: 2px;">CAA</div> <div style="border: 1px solid black; padding: 2px;">CAG</div> <i>Glutamine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">CGU</div> <div style="border: 1px solid black; padding: 2px;">CGC</div> <div style="border: 1px solid black; padding: 2px;">CGA</div> <div style="border: 1px solid black; padding: 2px;">CGG</div> </div> <i>Arginine</i>
	A	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">AUU</div> <div style="border: 1px solid black; padding: 2px;">AUC</div> <div style="border: 1px solid black; padding: 2px;">AUA</div> </div> <i>Isoleucine</i> <div style="border: 1px solid black; padding: 2px; background-color: green;">AUG</div> <i>Methionine; START codon</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">ACU</div> <div style="border: 1px solid black; padding: 2px;">ACC</div> <div style="border: 1px solid black; padding: 2px;">ACA</div> <div style="border: 1px solid black; padding: 2px;">ACG</div> </div> <i>Threonine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">AAU</div> <div style="border: 1px solid black; padding: 2px;">AAC</div> </div> <i>Asparagine</i> <div style="border: 1px solid black; padding: 2px;">AAA</div> <div style="border: 1px solid black; padding: 2px;">AAG</div> <i>Lysine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">AGU</div> <div style="border: 1px solid black; padding: 2px;">AGC</div> </div> <i>Serine</i> <div style="border: 1px solid black; padding: 2px;">AGA</div> <div style="border: 1px solid black; padding: 2px;">AGG</div> <i>Arginine</i>
	G	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">GUU</div> <div style="border: 1px solid black; padding: 2px;">GUC</div> <div style="border: 1px solid black; padding: 2px;">GUA</div> <div style="border: 1px solid black; padding: 2px;">GUG</div> </div> <i>Valine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">GCU</div> <div style="border: 1px solid black; padding: 2px;">GCC</div> <div style="border: 1px solid black; padding: 2px;">GCA</div> <div style="border: 1px solid black; padding: 2px;">GCG</div> </div> <i>Alanine</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">GAU</div> <div style="border: 1px solid black; padding: 2px;">GAC</div> </div> <i>Aspartic acid</i> <div style="border: 1px solid black; padding: 2px;">GAA</div> <div style="border: 1px solid black; padding: 2px;">GAG</div> <i>Glutamic acid</i>	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; padding: 2px;">GGU</div> <div style="border: 1px solid black; padding: 2px;">GGC</div> <div style="border: 1px solid black; padding: 2px;">GGA</div> <div style="border: 1px solid black; padding: 2px;">GGG</div> </div> <i>Glycine</i>

Third letter

Alinieri de perechi de secvențe

1. Care este alinierea globală pentru secvențele `APPLE` și `HAPPE`? Identificați toate alinierea optime precum și drumurile corespunzătoare folosind scorul $+1$ pentru potrivire (“match”), și -1 pentru nepotrivire (“mismatch”), ștergere și inserție.

2. Calculați scorurile următoarelor alinieri:
 - a. aliniere globală, $\text{match} = 1$, $\text{mismatch} = 0$, $\text{gap} = -1$
AATCGAGGGCTC
AGT-GA--GCCC
 - b. aliniere globală, $\text{match} = 1$, $\text{mismatch} = 0$, $\text{gap opening} = -2$, $\text{gap extension} = -1$.
CCTCTAACGGATGT
C--CTGA-GGTT--

3.
 - a. Construiți alinierea globală pentru secvențele `GTCCGTCAAT` și `GTCGGTAA` folosind algoritmul de programare dinamică Needleman–Wunsch. Scoruri: $\text{match} = 1$, $\text{mismatch} = 0$, $\text{gap} = -1$. Indicați alinierea optimă și scorul ei.
 - b. Construiți alinierea locală pentru secvențele `GTCCGTCAAT` și `TCCTTC` folosind algoritmul de programare dinamică Smith–Waterman. Scoruri: $\text{match} = 1$, $\text{mismatch} = 0$, $\text{gap} = -1$. Indicați alinierea optimă și scorul ei.

4. Fie secvențele $v = \text{TACGGGTAT}$ și $w = \text{GGACGTACG}$. Considerăm că recompensa pentru potrivire (“match”) este $+1$ iar penalizarea pentru nepotrivire (“mismatch”), inserții și ștergeri este -1 .
 - (a) Completați tabela de programare dinamică pentru alinierea globală a secvențelor v și w .
Trasați săgețile corespunzătoare informațiilor memorate pentru backtracking.
Care este scorul optim pentru alinierea globală?
Cărei alinieri îi corespunde acest scor?
 - (b) Similar, pentru alinierea locală a secvențelor v și w .
 - (c) Presupunem că penalizarea pentru un spațiu (“gap”) afin este de -20 pentru deschiderea spațiului și -1 pentru extinderea lui.
Care este alinierea globală optimă în acest caz și scorul ei?

5. Enunțați algoritmul Smith-Waterman.

8.

		H	E	A	G	A	W	G	H	E	E	
	0	-12	-14	-16	-18	-20	-22	-24	-26	-28	-30	
P	-12	-2	-2	-13	-15	-18	-19	-22	-24	-26	-27	-29
A	-14	-14	-1	-3	-8	-15	-13	-21	-22	-25	-27	-28
W	-16	-16	-3	-3	-6	-11	-18	2	-10	-12	-14	-16
H	-18	-6	10	0	-2	-2	-2	-3	-2	10	0	0
E	-20	-18	0	6	-1	-3	-1	-3	-3	0	6	6
A	-22	-20	-2	-1	5	0	5	-3	0	-2	-1	-1
E	-24	-22	0	6	-1	-3	-1	-3	-3	0	6	6

Q1: Ce credeți că reprezintă tabelul de mai sus?

A1:

Q2: Acest tabel poate fi obținut cu ajutorul unui algoritm care aplică două seturi de relații:

A2:

- relații de inițializare:
- relații de recurență:

Q3: Care este output-ul algoritmului desemnat mai sus?

(I.e., Care este rezultatul interpretării datelor din tabelul dat?)

A3:

- numeric:
- nenumeric:

9. Două secvențe de ADN se aliază folosind algoritmul Needleman–Wunsch. Se folosesc următoarele scoruri: +3 pentru match, +1 pentru *tranziții* (A-G și C-T) și -1 pentru celelalte perechi de litere (*traversări*).

Se obține următoarea matrice de programare dinamică:

		<i>G</i>	<i>A</i>	<i>T</i>	<i>T</i>	<i>A</i>	<i>C</i>	<i>A</i>	<i>T</i>	<i>A</i>
	0	-4	-5	-6	-7	-8	-9	-10	-11	-12
<i>G</i>	-4	3	-1	-2	-3	-4	-5	-6	-7	-8
<i>C</i>	-5	-1	2	0	-1	-4	-1	-5	-5	-8
<i>C</i>	-6	-2	-2	3	1	-2	-1	-2	-4	Z
<i>A</i>	-7	-3	1	-1	2	4	0	2	-2	-1
<i>G</i>	-8	-4	-2	0	-2	3	3	1	1	-1
<i>G</i>	-9	-5	-3	-3	-1	-1	2	4	0	2
<i>T</i>	-10	X	-6	0	0	-2	0	1	7	3
<i>A</i>	-11	-7	-3	-4	-1	3	-1	3	3	10
<i>A</i>	-12	-8	-4	-4	Y	2	2	2	2	6
<i>G</i>	-13	-9	-7	-5	-5	-2	1	3	1	5

- a. Care sunt scorurile pentru deschidere de gap (d) și respectiv extensie de gap (e) folosite în calcularea acestei matrice?
 - b. Care sunt valorile pentru cele trei elemente X, Y, Z din matrice? Justificați în detaliu.
 - c. Derivați o aliniere optimă. Scrieți alinierea respectivă. Calculați scorul ei în mod direct.
 - d. Este aceasta unica aliniere optimă?
10. a. Care este semnificația elementului (i, j) din matricea de programare dinamică pentru alinierea globală a două secvențe?
- b. Dar semnificația statistică a termenului $s(x_i, y_j)$ din formula de recurență $F(i, j) = F(i - 1, j - 1) + s(x_i, y_j)$?
11. Matricea de programare dinamică de mai jos este obținută folosind un scor unic pentru match perfect (a), un scor unic pentru mismatch (b) și o penalizare liniară (d) pentru gap-uri.

		<i>T</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>C</i>	<i>G</i>	<i>T</i>	<i>A</i>
	0	-6	-12	-18	-24	-30	-36	-42	-48
<i>T</i>	-6	5	-1	-7	-13	-19	-25	-31	-37
<i>T</i>	-12	-1	3	-3	-2	-8	-14	-20	-26
<i>C</i>	-18	-7	-3	8	2	3	-3	-9	-15
<i>A</i>	-24	-13	-9	2	6	0	1	-5	-4
<i>T</i>	-30	-19	-15	-4	7	4	-2	6	0
<i>A</i>	-36	-25	-21	-10	1	5	2	0	11

Indicați

- a. valorile a , b și d
- b. alinierea/alinierea optime și scorul/scorurile lor.

12. a. Ce implementează programul C de mai jos?

b. Inserați o linie de cod care să imprime rezultatul programului.

c. Sunt suficiente câteva mici modificări pentru a transforma acest program într-altul, care este de asemenea foarte util. Specificați fiecare dintre aceste modificări sub forma de cod în comentariu, plasat la locul adecvat.

```
#define SEQX "TTCATA"
#define SEQY "TGCTCGTA"

#define MATCH 5
#define MISMATCH -2
#define INDEL -6

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int
main(void)
{
    char    *x,*y;
    int     M,N;
    int     i,j;
    int     **S;
    int     sc;

    M = strlen(SEQX);
    N = strlen(SEQY);
    x = malloc(sizeof(char) * (M+2));
    y = malloc(sizeof(char) * (N+2));
    strcpy(x+1, SEQX);
    strcpy(y+1, SEQY);

    S = malloc(sizeof(int *) * (M+1));
```

```

for (i = 0; i <= M; i++)
    S[i] = malloc(sizeof(int) * (N+1));

S[0][0] = 0;
for (i = 1; i <= M; i++) S[i][0] = i * INDEL;
for (j = 1; j <= N; j++) S[0][j] = j * INDEL;

for (i = 1; i <= M; i++)
    for (j = 1; j <= N; j++)
    {
        if (x[i] == y[j]) S[i][j] = S[i-1][j-1] + MATCH;
        else S[i][j] = S[i-1][j-1] + MISMATCH;

        sc = S[i-1][j] + INDEL;
        if (sc > S[i][j]) S[i][j] = sc;

        sc = S[i][j-1] + INDEL;
        if (sc > S[i][j]) S[i][j] = sc;
    }

free(x); free(y);
for (i = 0; i <= M; i++) free(S[i]);
free(S);
exit(0);

```

13. Demonstrați că la alinierea globală cu complexitate de spațiu liniară timpul de execuție se dublează.

14. Aplicați algoritmul de aliniere globală (Needleman-Wunsch) respectiv de aliniere locală (Waterman-Smith) pentru a găsi cele mai bune alinieri pentru secvențele *GAGGC* și *GAGA*, folosind scorurile următoare: +1 match, -1 mismatch, -2 gap (liniar).

Scrieți care sunt cele două alinieri și cât sunt scorurile lor.

		<i>G</i>	<i>A</i>	<i>G</i>	<i>A</i>
<i>G</i>					
<i>A</i>					
<i>G</i>					
<i>G</i>					
<i>C</i>					

		<i>G</i>	<i>A</i>	<i>G</i>	<i>A</i>
<i>G</i>					
<i>A</i>					
<i>G</i>					
<i>G</i>					
<i>C</i>					

15. a. Aplicați algoritmul Waterman-Eggert pentru a găsi cele mai bune două alinieri locale pentru secvențele *GAGGC* și *GAGA*, folosind scorurile următoare: +1 match, -1 mismatch, -2 gap (liniar).
b. Scrieți care sunt cele două alinieri și cât sunt scorurile lor.

		G	A	G	A
G					
A					
G					
G					
C					

		G	A	G	A
G					
A					
G					
G					
C					

16. Indicați toate alinierea optime corespunzătoare matricii de aliniere locală:

	A	G	G	C	C	A	T	G	T	T	G	G	C	A	A	A	C	G
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	1	1	0	0	0	0	1	0	0	1	1	0	0	0	0	1
G	0	0	1	2	0	0	0	0	1	0	0	1	2	0	0	0	0	1
C	0	0	0	0	3	1	0	0	0	0	0	0	0	3	1	0	0	1
A	0	1	0	0	1	2	2	0	0	0	0	0	1	4	2	1	0	0
T	0	0	0	0	0	0	1	3	1	1	1	0	0	0	2	3	1	0
G	0	0	1	1	0	0	0	1	4	2	0	2	1	0	0	1	2	0
C	0	0	0	0	2	1	0	0	2	3	1	0	1	2	0	0	0	3
T	0	0	0	0	0	1	0	1	0	3	4	2	0	0	1	0	0	1
A	0	1	0	0	0	0	2	0	0	1	2	3	1	0	1	2	1	0
A	0	1	0	0	0	0	1	1	0	0	0	1	2	0	1	2	3	1
T	0	0	0	0	0	0	0	2	0	1	1	0	0	1	0	0	1	2
G	0	0	1	1	0	0	0	0	3	1	0	2	1	0	0	0	0	3
C	0	0	0	0	2	1	0	0	1	2	0	0	1	2	0	0	0	1

17. (prelucrare după [Dan Gusfeld], cap. 11, pr 32, pag 247)

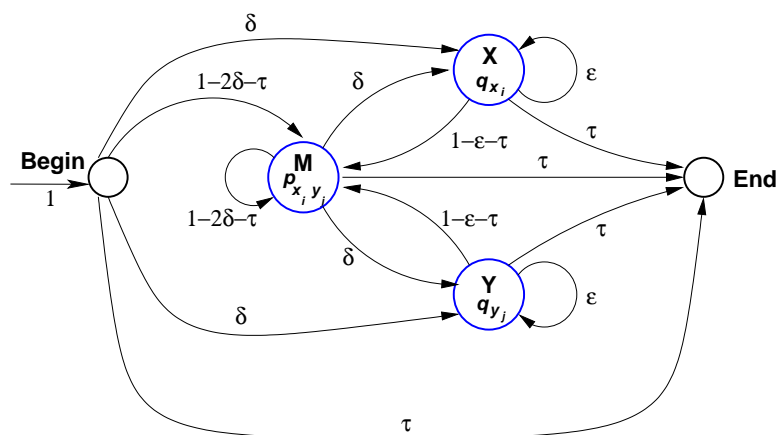
Fie P un pattern (o secvență) de lungime n iar T un text (secvență) de lungime m . Vrem să calculăm o aliniere locală dintre T și P^m , acesta din urmă fiind concatenarea lui P cu el însuși de m ori.

[Problema aceasta ("tandem repeat" — repetiții în tandem) apare în studiul structurii secundare a proteinelor, și anume la identificarea substructurilor repetitive.]

- Problema repetițiilor în tandem poate fi rezolvată printr-un algoritm de complexitate $\mathcal{O}(nm^2)$. Care/cum anume?
- Există o soluție de complexitate $\mathcal{O}(nm)$? Detaliați.

HMM Pereche

1. a. Fie următorul HMM pereche pentru alinierea globală a două secvențe genetice folosind gap-uri afine:



- b. Pentru un astfel de HMM pereche, indicați drumurile pe care sunt generate perechile de secvențe de mai jos, precum și probabilitățile de producere a lor:

HEAGAWGHEE
-P-A-WHEAE

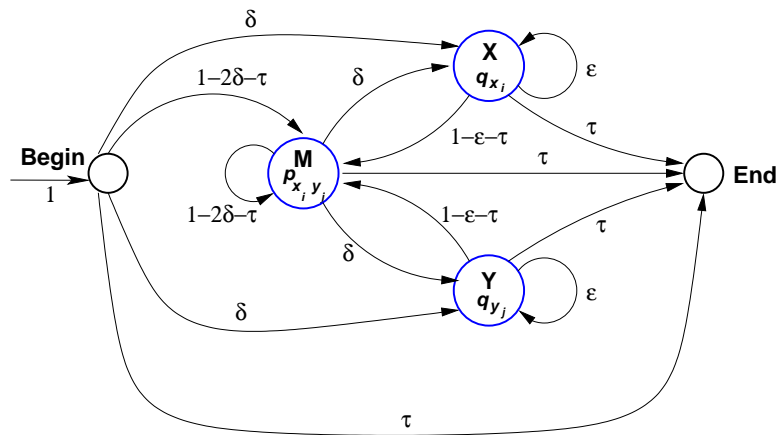
HEAGAWGHE-E
--P-AW-HEAE

- c. Cu ce algoritm se poate calcula $P(x, y, \pi^*)$, unde π^* este calea optimă de aliniere a secvențelor x și y ?
- d. Cu ce algoritm se poate calcula $P(x, y)$, probabilitatea ca două secvențe x și y să fie relaționate/înrudite indiferent de calea de aliniere?
- e. Ce reprezintă și cum se calculează $P(\pi^* \mid x, y)$?
- f. Adevărat sau fals?
 $P(x, y, \pi^*) > P(x, y)$
- g. Definiți relațiile de inițializare, recurență și terminare pentru algoritmul Viterbi (variantea log-odds) pentru HMM pereche.
- h. Adaptați procedura de back-tracing a algoritmului Viterbi pentru HMM pereche astfel încât să generați în mod probabilist alinieri suboptimale.

2. a. Construiți un HMM pereche pentru aliniere globală de secvențe cu gap-uri afine de scor nu de forma $d + ek$ ci $\min\{d_1 + e_1k, d_2 + e_2k, \dots, d_s + e_s k\}$, având cel mult $2s + 1$ stări. (Ignorați stările Begin și End.)
- b. La alinierea unei perechi de secvențe $x = x_1x_2 \dots x_n$ și $y = y_1y_2 \dots y_m$, vi se dă informația *a priori* că un anumit caracter x_A ($1 < A < n$) trebuie să fie

aliniat cu unul din caracterele y_B, y_C sau y_D ($1 < B < C < D < m$). (Se presupune că probabilitatea *a priori* a acestor trei evenimente este uniformă.) Cum veți modifica algoritmul Viterbi pentru a găsi cea mai probabilă aliniere cu această restricție?

3. Modelul Markov pereche care face alinierea globală a două secvențe folosind gap-uri afine este:



Fie corpusul constituit din următoarele alinieri de perechi de secvențe:

HEAGAWGHEE	HEAGAWGHEE	HEAGAWGHEE	HEAGAWGHEE
-P-A-WHEAE	-P--AWHEAE	P---AWHEAE	-P-A-WHEAE
HEAGAWGHE-E	HEAGAWGHE-E	HEAGAWGHE-E	HEAGAWGHEE
-PA--W-HEAE	--P-AW-HEAE	-P--AW-HEAE	--PA-WHEAE

a. Indicați drumul parcurs de (automatul reprezentând) HMM pereche la generarea primei alinieri de mai sus.

b. Care este probabilitatea drumului de la punctul a. în funcție de δ, ϵ și τ ?

c. Putem estima (în sensul verosimilității maxime — MLE) probabilitățile de emisie pentru HMM pereche folosind corpusul dat. De exemplu, $p_{AE} = \frac{5}{53}$ iar $q_A = \frac{12}{27}$. Similar pentru parametrii δ, ϵ și τ .

Exprimați $P(x, y, \pi_1 | \mu)$, probabilitatea emiterii primei alinieri din corpus în funcție de (estimările pentru) parametrii δ, ϵ și τ, p_{ab} și q_a , unde a și b sunt simbolii care apar în cele două secvențe.

d. Cum se poate calcula $P(x, y | \mu)$, probabilitatea alinierii celor două secvențe independent de drum?

e. Cum se poate calcula $P(\pi_1 | x, y, \mu)$, probabilitatea a posteriori a producerii primei alinieri din corpus?

f. Cum se poate calcula $P(x, y, W \diamond W \mid \mu)$, probabilitatea ca simbolul W din prima secvență să fie aliniat cu simbolul W din a doua secvență?

4. [HMM pereche vs programare dinamică
pentru alinieri de perechi de secvențe]

Enumerați foarte succint principalele avantaje ale folosirii HMM pereche în raport cu algoritmi clasici de programare dinamică pentru alinieri de perechi de secvențe.

(a)

(b)

(c)

(d)

(e)

Filogenetică

1. [Filogenetică bazată pe distanțe]

Fie alinierea multiplă de mai jos:

Rat	GWTYREKTHGAL
Mouse	GWTYKEKSHGAL
Horse	AWTYKEKTHGGI
Man	AWSYRERTHGGI

- Scriveți matricea de distanțe mutuale dintre aceste secvențe, folosind distanța Hamming.
- Folosind această matrice, construiți arborele UPGMA (Maximum Linkage) corespunzător acestor secvențe.
- Verificați dacă distanțele măsurate pe arborele UPGMA coincid cu cele calculate anterior (în sens Hamming) pentru secvențele date. Cum explicați această situație? Se putea prevedea dinainte această situație?

2. [Filogenetică bazată pe distanțe]

- (a) Completați matricea de mai jos calculând distanța Hamming dintre șirurile date.

	$x_1 = \text{ATCC}$	$x_2 = \text{ATGC}$	$x_3 = \text{TTCG}$	$x_4 = \text{TCGG}$
$x_1 = \text{ATCG}$				
$x_2 = \text{ATGC}$				
$x_3 = \text{TTCG}$				
$x_4 = \text{TCGG}$				

- (b) Pentru matricea de distanțe obținută la punctul 1, verificați dacă sunt îndeplinite condițiile de
- aditivitate
 - ultrametrică.
- (c) Pentru aceeași matrice construiți arborii
- UPGMA
 - NJ (neighbour-joining).
- (d) La arborele UPGMA obținut la punctul 3.a, aplicați algoritmi
- Fitch
 - Sankoff.
- (Atenție!...)
- Care este scorul optim de parcimonie?

d. Indicați, dacă este cazul, cel puțin două soluții de filogenie pentru această problemă. (O astfel de soluție este reprezentată de o *etichetare a nodurilor*; etichetarea se obține prin backtracing, la aplicarea algoritmilor de parcimonie.)

(e) **Bonus:**

Ce ar însemna să aplicați algoritmul Nearest Neighbour Interchange (de parcimonie în sens larg) pentru același input ca la punctul 4?

3. Filogenetică bazată pe distanțe

Folosind algoritmul UPGMA (average linkage), construiți arborele filogenetic al speciilor A, B, C și D, pentru care matricea de distanțe este

	A	B	C	D
A		4	12	17
B			12	17
C				15

4. Filogenetică bazată pe distanțe

Fie matricea de distanțe alăturată.

a. Sunt aceste distanțe ultrametrice?

b. Construiți arborele UPGMA corespunzător acestor distanțe.

D_{ij}	x_1	x_2	x_3	x_4
x_1		1.25	0.95	1.31
x_2			1.24	1.30
x_3				1.13
x_4				

5. Filogenetică bazată pe distanțe

Fie următoarele secvențe genetice:

secvența A: ACGCGTTGGGCGATGGCAAC

secvența B: ACGCGTTGGGCGACGGTAAT

secvența C: ACGCATTGAATGATGATAAT

secvența D: ACACATTGAGTGATAATAAT

a. Calculați matricea distanțelor Hamming (numărul de nepotriviri) dintre ele.

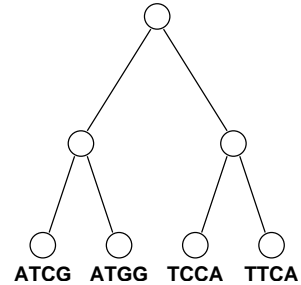
b. Sunt aceste distanțe aditive? (Justificați răspunsul.)

c. Dar ultrametrice? (Justificați răspunsul.)

d. Aplicând algoritmul Neighbour-Joining, construiți arborele filogenetic fără rădăcină corespunzător acestor secvențe.

6. [Filogenetică bazată pe caractere]

Aplicați algoritmul lui Fitch (inclusiv procedura de backtracing) pe arborele din figura alăturată. Notați mutațiile direct pe desen. Dacă există mai multe soluții, indicați-le pe toate.



7. [Filogenetică bazată pe caractere]

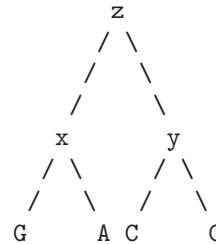
Pentru algoritmul lui Sankoff de rezolvare a problemei parsimoniei mici ponderate (“Weighted Small Parsimony Problem”), concepeți o procedură de backtracing care să reconstruiască asignarea optimală a caracterelor.

8. [Filogenetică bazată pe caractere]

Pentru arborele filogenetic de mai jos, construiți asignarea cea mai parcimonioasă, folosind matricea de costuri alăturată.

costuri:

	A	C	G	T
A	0	4	1	4
C	4	0	4	1
G	1	4	0	4
T	4	1	4	0



Arătați vectorii de costuri pentru fiecare nod x , y , z precum și costul de parsimonie obținut în final. Dacă există mai multe soluții optimale, le veți identifica pe toate.

Alinieri de secvențe multiple

1. Fie secvențele ATC, CAGC, CGC și scorurile

$$s(a, b) = \begin{cases} 1, & \text{dacă } a \text{ și } b \text{ sunt litere identice} \\ 0, & \text{dacă } a \text{ și } b \text{ sunt litere diferite} \\ -1, & \text{dacă } b \text{ este gap } (-) \end{cases}$$

- a. Găsiți toate alinierea multiple optime pentru secvențele date. Care este scorul optim?
 - b. Vizualizați într-un paralelipiped (matrice tridimensională) drumurile corespunzătoare acestor alinieri optime.
 - c. Folsind rezultatele de la punctele precedente, indicați o margine inferioară pentru numărul de elemente care vor fi calculate de varianta *branch-and-bound* a algoritmului de programare dinamică multidimensională (generalizarea algoritmului Needleman–Wunsch) la rezolvarea acestei probleme.
 - d. Construiți profilul corespunzător uneia din alinierea optime obținute.
2. Să se demonstreze inegalitatea folosită de algoritmul MSA (Carillo-Lipman) pentru limitarea spațiului de căutare în matricea de programare multidimensională:

$$S(a^{kl}) \geq \sigma(a) + S(\hat{a}^{kl}) - \sum_{k' < l'} S(\hat{a}^{k'l'})$$

unde

a este o aliniere multiplă optimală; scorul acestei alinieri este $S(a)$;

a^{kl} este alinierea perechii de secvențe k, l din a ;

\hat{a}^{kl} este alinierea optimală a perechii de secvențe k, l ;

$\sigma(a)$ este scorul unei alinieri (de obicei ne-optimală) a secvențelor din a , obținută (de exemplu) folosind un algoritm euristic de aliniere multiplă.

3. (prelucrare după [Borodovsky & Ekisheva], pr 6.2, pag 165)

Se consideră secvențele $x_1 = CTCACA$, $x_2 = CAC$ și $x_3 = GTAC$. Fie matricea de scoruri

S	A	C	G	T
A	0	-2	-1	-2
C		0	-2	-1
G			0	-2
T				0

și penalizările de spațiu $d = -3$ și $e = -2$.

a. Să se calculeze scorul următoarei alinieri multiple (în sensul “sum of pairs”):

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

b. Folosind algoritmul Needleman-Wunsch, calculați scorurile optime pentru perechile de secvențe (x_1, x_2) , (x_1, x_3) , (x_2, x_3) .

c. Cu rezultatele de la punctele a. și b., calculați pragul

$$\beta^{12} = \sigma(a) + S(\hat{a}^{12}) - \sum_{k' < l'} S(\hat{a}^{k'l'})$$

folosit în aplicarea algoritmului MSA.

d. Calculați mulțimea de indici B^{12} corespunzătoare pragului β^{12} .

Remember: B^{12} este formată din perechile de indici (a, b) care au proprietatea că există cel puțin o aliniere a secvențelor (x_1, x_2) de scor mai mare sau egal cu β^{12} astfel încât drumul corespunzător acestei alinieri trece prin (a, b) .

Atenție: veți avea nevoie să aplicați aici algoritmul Needleman-Wunsch în variantă backward (nu forward, cum se aplică în general). În matricea care reprezintă suma matricilor de programare dinamică pentru cele două variante de aliniere (forward și backward), veți identifica elementele de scor mai mare sau egal cu β^{ij} .

e. Cum va proceda în continuare algoritmul MSA pentru a obține soluția optimă de aliniere a secvențelor x_1, x_2, x_3 ? Explicați succint.

4. Secvențele genetice x_1, x_2, x_3 și x_4 de mai jos reprezintă fragmente din familia I-imunoglobinelor.

```
ILDMDVVEGSAARFDCKVEGYPDPEVMWFKDDNPVKESRHFQIDYDEEGN
RDPVKTHEGWGVMPLCPNPPAHYPGLSYRWLLNEFPNFIPTDGRHFVSQTT
ISDTEADIGSNLRWGCAAAGKPRPMVRWLRNGEPLASQNRVEVLA
LRLIPAARGGEISILCQPRAAPKATILWSKGTEILGNSTRVTVTSD
```

Pentru fiecare pereche (x_1, x_2) , (x_1, x_3) , etc. vi se dau alinierea globală care au fost obținute folosind matricea de scoruri PAM260 și penalizarea de spațiu $d = -8$:

```
A12  ILDMDVVEGSAARFDCKVEG-YPDPEVMWFKDDNPVKESRHFQIDYDEEGN
      RDPVKTHEGWGVMPLCPNPPAHYPGLSYRWLLNEFPNFIPTD-GRHFVSQTT

A13  ILDMDVVEGSAARFDCKVEGYPDPEVMWFKDDNPVKESRHFQIDYDEEGN
      ISDTEADIGSNLRWGCAAAGKPRPMVRWLRNGEPL-ASQN-RV--EVL-

A14  ILDMDVVEGSAARFDCKVEGYPDPEVMWFKDDNPVKESRHFQIDYDEEGN
      LRLIPAARGGEISILCQPRAAPKATILWSKGTE-ILGNST-RV--TVTSD

A23  RDPVKTHEGWGVMPLCPNPPAHYPGLSYRWLLNEFPNFIPTDGRHFVSQTT
      ISDTEADIGSNLRWGCAAAGKPRPMV-RWLRNGEP--LASQNRV--EVL-

A24  RDPVKTHEGWGVMPLCPNPPAHYPGLSYRWLLNEFPNFIPTDGRHFVSQTT
      LRLIPAARGGEISILCQPRAA-PKATILW-SKG-TEILGNSTRVTVT-SD

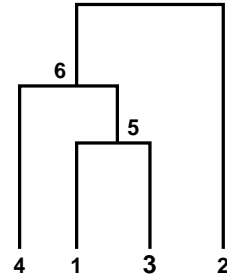
A34  ISDTEADIGSNLRWGCAAAGKPRPMVRWLRNGEPLASQNRVEVLA-
      LRLIPAARGGEISILCQPRAAPKATILWSKGTEILGNSTRVTVTSD
```

Scorurile acestor alinieri sunt:

$$S_{12} = 31, S_{13} = 44, S_{14} = 13, S_{23} = 15, S_{24} = 16, S_{34} = 45.$$

a. Construiți alinierea multiplă a acestor secvențe folosind algoritmul Feng-Doolittle.

Se presupune că arborele de ghidare obținut prin aplicarea algoritmului Fitch-Margoliash pentru secvențele de mai sus este cel din figura alăturată.



Reminder: La algoritmul Feng-Doolittle, alinierea unei secvențe y cu o aliniere multiplă MA este determinată de alinierea de scor maxim (y, y_i) unde $y_i \in MA$. (Această regulă se generalizează imediat la cazul alinierii a două alinieri multiple.)

b. Dacă în locul matricii de scoruri PAM260 (vezi mai sus) vom folosi matricea BLOSUM62, ce credeți că este posibil să se modifice:

- scorurile S_{ij}
- arborele de ghidare Fitch-Margoliash
- alinierea multiplă (Feng-Doolittle)

5. Fie secvențele: $s_1 = \text{NFLS}$, $s_2 = \text{NFS}$, $s_3 = \text{NKYLS}$, $s_4 = \text{NYLS}$.

Se consideră următoarele scoruri (de tip pereche):

$s(a, a) = 1$, $s(a, b) = 0$ pentru $a \neq b$, $s(a, -) = -1$, și $s(-, -) = 0$, unde a și b pot fi oricare din caracterele ce apar în secvențele date.

a. Pentru primul pas al algoritmului Barton-Sternberg (MSA cu rafinare iterativă) este necesară mai întâi calcularea scorurilor Viterbi la alinierea globală a secvențelor date, două câte două.

Completați în tabelul de mai jos aceste scoruri.

S	s_1	s_2	s_3	s_4
s_1	•			
s_2	•	•		
s_3	•	•	•	
s_4	•	•	•	•

b. Indicați ordinea în care vor fi folosite secvențele s_1, s_2, s_3, s_4 pentru constituirea alinierii multiple inițiale (care va fi eventual îmbunătățită în iterațiile ulterioare) de către acest algoritm.

Observație: se vor face alinieri de secvențe la profiluri.

1	2	3	4

c. Care este complexitatea algoritmului naiv de programare dinamică multi-dimensională pentru alinieri de secvențe multiple? (Folosiți \bar{L} = lungimea medie a secvențelor, N numărul de secvențe.)
Particularizați pentru secvențele date mai sus.

d. În algoritmul MSA Carillo-Lipman, spațiul de căutare a drumului optim se poate reduce foarte mult folosind pragurile (*bounds*)

$$\beta^{kl} = \sigma(a) + S(\hat{a}^{kl}) - \sum_{k' < l'} S(\hat{a}^{k'l'}),$$

unde $S(\hat{a}^{k'l'})$ reprezintă scorul Needleman-Wunsch pentru alinierea secvențelor $s_{k'}$ și $s_{l'}$.

Calculați β^{12} , folosind drept $\sigma(a)$ scorul alinierii multiple rezultate de la punctul b. de mai sus (adică output-ul primului pas al algoritmului Barton-Sternberg).

e. (Bonus!) Calculați B^{12} , mulțimea perechilor de valori i, j (unde $1 \leq i \leq \text{length}(s_1)$ și $1 \leq j \leq \text{length}(s_2)$) având proprietatea următoare: la alinierea globală a secvențelor s_1 și s_2 există un drum π_{ij} care

- trece prin elementul (i, j) din matricea de programare dinamică pentru aliniere, și
- are un scor mai mare sau egal cu β^{12} .

6. [HMM Profil]

Fie următoarea aliniere de secvențe genetice, având coloanele marcate (X) pentru pozițiile de match:

	X	X	.	.	.	X
bat	A	G	-	-	-	C
rat	A	-	A	G	-	C
cat	A	G	-	A	A	-
gnat	-	-	A	A	A	C
goat	A	G	-	-	-	C
	1	2	.	.	.	3

A.

- (a) Desenați HMM profil corespunzător acestei alinieri multiple.
- (b) În acest HMM profil indicați care sunt drumurile pe care sunt generate fiecare din secvențele date.
- (c) Stabiliți conform metodei MLE (Maximum Likelihood Estimate)
 - probabilitățile de emisie de caractere dintr-o stare match (M2) și o stare insert (I2);
 - probabilitățile de tranziție în (și respectiv dintr-o) stare delete (D2).
- (d) Îmbunătățiți probabilitățile de la punctul precedent (făcând smoothing) cu ajutorul regulii lui Laplace.

B.

Presupunem că dispunem și de alte secvențe genetice care fac parte din aceeași familie ca și secvențele date mai sus.

- (e) Ce algoritm se aplică pentru a stabili alinierea fiecăreia dintre aceste secvențe la modelul HMM profil stabilit anterior?
- (f) Definiți relațiile de inițializare, recurență și terminare pentru algoritmul indicat la punctul precedent.
- (g) Se pot îmbunătăți valorile probabilităților stabilite mai sus (punctul A3/A4) folosind astfel de secvențe noi? Dacă da, cum anume?

Bonus:

- (h) Care sunt câteva dintre avantajele folosirii HMM profil față de metodele algoritmice pentru aliniere de secvențe multiple care au fost prezentate anterior la curs?

7. [HMM Profil]

Fie următoarea aliniere multiplă:

1	2	3	4	5
<i>N</i>	–	<i>F</i>	<i>L</i>	<i>S</i>
<i>N</i>	–	<i>F</i>	–	<i>S</i>
<i>N</i>	<i>K</i>	<i>Y</i>	<i>L</i>	<i>S</i>
<i>N</i>	–	<i>Y</i>	<i>L</i>	<i>S</i>

- a. Completați matricea de frecvențe (profilul) care corespunde acestei alinieri multiple.

	1	2	3	4	5
<i>N</i>					
<i>K</i>					
<i>F</i>					
<i>Y</i>					
<i>L</i>					
<i>S</i>					
–					

- b. Care este scorul următoarei alinieri a secvenței $s = NKGYS$ la profilul de la punctul a.?

1	2	–	3	4	5
<i>N</i>	<i>K</i>	<i>G</i>	<i>Y</i>	–	<i>S</i>

Se vor folosi scorurile $s(a, a) = 1$, $s(a, b) = 0$ pentru $a \neq b$, $s(a, -) = s(-, a) = -1$, și $s(-, -) = 0$, unde a și b pot fi oricare din caracterele ce apar în secvențele date.

c. Folosiți euristica sugerată în “Biological Sequence Analysis” (Durbin et al. 1998) pentru a marca (cu X) coloanele alinierii multiple date mai sus.

d. Desenați modelul Markov profil corespunzător marcajului de la punctul c.

e. Completați coloanele 0 și 1 ale tabelului de mai jos în vederea estimării probabilităților de tranziție și a probabilităților de emisie pentru HMM profil de la punctul d. (Această estimare ar urma să folosească metoda MLE, Maximum Likelihood Estimation).

		0	1	2	3	4
match emissions	<i>N</i>					
	<i>K</i>					
	<i>F</i>					
	<i>Y</i>					
	<i>L</i>					
	<i>S</i>					
insert emissions	<i>N</i>					
	<i>K</i>					
	<i>F</i>					
	<i>Y</i>					
	<i>L</i>					
	<i>S</i>					
state transitions	<i>M – M</i>					
	<i>M – D</i>					
	<i>M – I</i>					
	<i>I – M</i>					
	<i>I – D</i>					
	<i>I – I</i>					
	<i>D – M</i>					
	<i>D – D</i>					
	<i>D – I</i>					

f. Care este probabilitatea de emisie a secvenței $s = NKGYS$ în modelul Markov profil obținut mai sus?
 (Bonus:) Ce remarcă puteți face?

8. [HMM Profil]

Fie următoarea aliniere multiplă de secvențe:

N*FLS
N*F-S
NKYLS
N*YLS

unde * reprezintă o inserție iar - este un spațiu (gap).

- Construiți un HMM (model Markov ascuns) profil corespunzător alinierii multiple de mai sus. Stabiliți în prealabil numărul de stări match, conform euristicii sugerate de Durbin et al. în *Biological Sequence Analysis*, 1998.
- Pentru acest model Markov profil, calculați care este probabilitatea observării fiecăreia din secvențele date mai sus.

9. [HMM Profil] (prelucrare după [Borodovsky & Ekisheva], pr 5.7, pag 138)

Fie următoarea aliniere multiplă:

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

- Faceți diagrama pentru modelul Markov ascuns de tip profil care corespunde acestei alinieri multiple.

Atenție:

- Pentru marcarea coloanelor, folosiți euristica simplă sugerată în cartea lui Durbin et al.
- Estimați probabilitățile de tranziție și de emisie în sensul verosimilității maxime (MLE).
- Nu vor fi incluse în diagramă arcele (tranzițiile) care au probabilitatea 0 și nici stările pentru care toate emisiile sunt de probabilitate 0.

- Indicați calea Viterbi de producere a secvenței GCCAG și probabilitatea de emisie corespunzătoare.

- Cum puteți proceda pentru a include secvența de la punctul b. în alinierea multiplă dată mai sus?

Bioinformatică, Master OC, anul II

— Subiecte de examen —

Nume student:

Email:

1. Rearanjări de genomuri (*Jones & Pevzner, Cap. 5*)

Aplicați algoritmul `BREAKPOINTREVERSALSORT` intrării $\pi = 34658172$. Indicați toate permutările intermediare.

Fiindcă acest algoritm este un algoritm de aproximare (“approximation algorithm”), este posibil să existe o secvență de inversări (“reversals”) mai scurtă decât cea găsită de `BREAKPOINTREVERSALSORT`. Puteți găsi o astfel de secvență de inversări? Știți care este cea mai scurtă secvență posibilă de inversări?

2. Rearanjări de genomuri (*Jones & Pevzner, Cap. 5*)

Nu orice moleculă de ADN este sub formă de segment liniar. Unele organisme simple au genomul în formă circulară, fără început și fără sfârșit. Un astfel de genom poate fi vizualizat ca o secvență de întregi scrisă de-a lungul perimetrului unui cerc. Două secvențe circulare vor fi considerate echivalente dacă unul din cercuri poate fi rotit astfel încât să se obțină secvența de numere de pe al doilea cerc.

Concepeți un algoritm aproximativ pentru sortarea unui genom circular prin inversări (“reversals”), adică transformați acest genom în permutarea circulară identică. Evaluați garanția de reușire (“performance guarantee”) a acestui algoritm.

3. Pattern matching (*Jones & Pevzner, Cap. 9*)

Concepeți un algoritm eficient care să găsească într-un șir cea mai lungă repetiție (“repeat”) cu cel mult o nepotrivire (“mismatch”).

4. Pattern matching (*Jones & Pevzner, Cap. 9*)

Concepeți un algoritm care să generalizeze algoritmul `APPROXIMATEPATTERNMATCHING` astfel încât între pattern și text să existe maximum k nepotriviri (“mismatches”), ștergeri sau inserțiuni.

5. MicroRNA-uri

Comentați în cuvinte proprii schema de mai jos.

