Curs 7.
Alinierea secvențelor:
matrici de scor (substitutie): PAM si BLOSUM
metode euristice de aliniere (FASTA si BLAST)

Biblio:

Cap 2. din “Biological sequence analysis”, Durbin et al

cap. 6 din “An introduction to Bioinformatics algorithms”, N.Jones, P. Pevzner

cap. 9 din “An introduction to Bioinformatics algorithms”, N.Jones, P. Pevzner

Alinierea secventelor - reminder

- Scopul alinierii: obținerea de informații privind similaritatea dintre secvențe
- Elemente cheie ale procesului de aliniere:
  - Stabilirea tipului de aliniere (globală, locală, multiplă)
  - Stabilirea matricilor de scor (pt nucleotide/aminoacizi) care vor fi utilizate pentru a evalua calitatea alinierii
  - Alegerea algoritmului de construire a alinierii (algoritm exact vs. algoritm aproximativ)
  - Stabilirea metodelor statistice utilizate pentru a evalua calitatea alinierii

Matrici de scor(substitutie)

- O matrice de scor (sau de substituție) conține pentru fiecare pereche de elemente (nucleotide sau aminoacizi) o valoare care exprimă șansa ca elementele respective să apară în alinierea unor secvențe corelate din punct de vedere biologic
- O matrice de scor are dimensiunea:
  - 4x4 - în cazul secvențelor de nucleotide
  - 20x20 – în cazul secvențelor de aminoacizi

- Observații:
  - Matricile de scor sunt simetrice prin urmare este suficient să se rețină doar partea inferior (sau superior) triunghiulară
  - Penalizările pentru gap-uri se pot specifica separat; în general nu depind de tipul de element care e aliniat cu gap-ul făcându-se diferență doar între penalizarea inițierii (gap opening) și cea a continuării (gap extension) unei secvențe de gap-uri

Matrici de scor (substitutie)

Matrici de scor pentru secvențe de aminoacizi (20 de aminoacizi):

- Scorul M(i,j) asociat unei perechi de aminoacizi se poate calcula în două moduri:
  - Pe baza proprietăților fizico chimice ale aminoacizilor din pereche (polarizare,hidrofobicitate,dimensiune etc.)
  - Pe baza unui model probabilist construit folosind frecvența de apariție a perechii în secvențe despre care se cunoaște că au evoluat pornind de la același strămoș (M(i,j) reflectă frecvența situațiilor în care aminoacidul i înlocuiește aminoacidul j în secvențe înrudite). In acest caz scorul se estimează pornind de la secvențe aliniate despre care se cunoaște că sunt corelate; alinierea preliminară a acestora se bazează pe scoruri stabilite simplu: +1 pentru potriviri și -1 pentru nepotriviri și inserări/ștergeri

Matrici de scor (substitutie)

Model probabilist pentru construirea matricilor de scor

Context: fie x[1..k] și y[1..k] două secvențe aliniate și două modele posibile:

- R (random): cele două secvențe sunt întâmplătoare (elementele din cele două secvențe pot fi modelate prin variabile aleatoare independente => probabilitatea de a observa perechea (x[i],y[i]) este egală cu produsul probabilităților de a observa separat x[i] respectiv y[i])
- C (correlated): cele două secvențe sunt corelate (probabilitatea perechii (x[i],y[i]) nu mai este neapărat produsul probabilităților

Probabilitatea de a observa o alinere dată (x,y) depinde de modelul considerat:

$$P(x,y \mid R) = \prod_{i=1}^{k} p(x_i)\,p(y_i) \qquad P(x,y \mid C) = \prod_{i=1}^{k} p(x_i, y_i)$$

$$p(a) = \text{probabilitatea ca } a \text{ sa fie in secventa}$$

$$p(a,b) = \text{probabilitatea ca } a \text{ sa fie aliniat cu } b$$

Matrici de scor (substitutie)

Model probabilist pentru construirea matricilor de scor

$$P(x, y \mid R) = \prod_{i=1}^{k} p(x_i)\, p(y_i) \qquad P(x, y \mid C) = \prod_{i=1}^{k} p(x_i, y_i)$$

$p(a) =$ probabilitatea ca a sa fie in secventa

$p(a,b) =$ probabilitatea ca a sa fie aliniat cu b

Raportul dintre cele două probabilități furnizează o măsură a șansei de a observa perechi de elemente aliniate în secvențe similare în raport cu șansa de a observa aceleași perechi în secvențe aleatoare:

$$\frac{P(x, y \mid C)}{P(x, y \mid R)} = \prod_{i=1}^{k} \frac{p(x_i, y_i)}{p(x_i)\, p(y_i)}$$

Raport de verosimilități (odds ratio)

In statistică se lucrează cu logaritmul acestui raport:

$$\log \frac{P(x, y \mid C)}{P(x, y \mid R)} = \log \prod_{i=1}^{k} \frac{p(x_i, y_i)}{p(x_i)\, p(y_i)} = \sum_{i=1}^{k} \log \frac{p(x_i, y_i)}{p(x_i)\, p(y_i)}$$

Matrici de scor (substitutie)

In aceste ipoteze log-raportul verosimilităților (log-odds ratio) corespunzător alinierii dintre x și y este:

$$\sum_{i=1}^{k} \log \frac{p(x_i, y_i)}{p(x_i)\,p(y_i)}$$

Prin urmare, pentru fiecare pereche (x_i,y_i) de elemente aliniate se poate asocia un scor de potrivire:

$$M(x_i, y_i) = \log \frac{p(x_i, y_i)}{p(x_i)\,p(y_i)}$$

Obs:

- In cazul in care perechea (x_i,y_i) apare aliniată mai frecvent în secvențe similare (biologic înrudite) decât în secvențe întâmplătoare atunci $M(x_i,y_i)$ este pozitivă

Matrici de scor (substitutie)

- La construirea matricilor de scor pe baza log-raportului de verosimilități elementul principal este reprezentat de estimarea probabilitătilor $p(x_i,y_i)$, $p(x_i)$, $p(y_i)$

- Probabilitățile individuale ale aminoacizilor se pot estima pe baza frecvențelor lor de apariție în cât mai multe secvențe reale

- Un exemplu de estimări (propuse de Dayhoff în 1978) este:

Gly	0.089	Val	0.065	Arg	0.041	His	0.034
Ala	0.087	Thr	0.058	Asn	0.040	Cys	0.033
Leu	0.085	Pro	0.051	Phe	0.040	Tyr	0.030
Lys	0.081	Glu	0.050	Gln	0.038	Met	0.015
Ser	0.070	Asp	0.047	Ile	0.037	Trp	0.010

Matrici de scor (substitutie)

- In funcție de modul de estimare a probabilităților $p(x_i,y_i)$ (pe baza unor frecvențe calculate pornind de la secvențe reale) există două clase principale de matrici de scor:

  - Matrici de tip PAM ("Point Accepted Mutations"): se pornește de la secvențe aliniate foarte similare (care diferă în mai puțin de 15% din aminoacizii constituenți – astfel de proteine foarte similare sunt întâlnite de exemplu la cimpanzeu și la om). O mutație punctuală acceptată (Point Accepted Mutation) se referă la o mutație non-fatală care afectează o poziție.

    (propuse in 1978)

  - Matrici de tip BLOSUM ("Block Substitution Matrix"): se folosesc secvențe local aliniate aparținând unor proteine similare (din baza de date BLOCKS)

    (propuse în 1992)

Matrici PAM

- Au fost propuse de către Dayhoff et al (1978) pornind de la alinieri ale secvențelor asociate unor proteine foarte similare despre care au presupus că au evoluat una din cealaltă într-o etapă de evoluție; pe baza acestor alinieri se construiește o primă matrice M în care M(a,b) reprezintă scorul substituției lui a în b (și invers) într-o singură etapă de evoluție.

- Pornind de la M(a,b) se poate calcula probabilitatea (condiționată) ca elementul a să se transforme în elementul b:

$$P(b \mid a) = \frac{M(a,b)}{\sum_{c} M(a,c)}$$

- Matricea T care conține pe linia a, coloana b probabilitatea P(b|a) poate fi interpretată ca o matrice de tranziție (T(a,b) e probabilitatea de trecere într-un pas de la a la b)

Matrici PAM

- Folosind proprietățile matricilor de tranziție rezultă că puterea m a matricii T va conține probabilitățile de tranziție în m etape; aceasta este ideea construirii matricilor PAM(m)
- Pentru a construi $PAM_{(1)}$ din T se impune condiția ca numărul mediu de substituții să fie 1%, adică:

$$\sum_{a,b} p(a)\,p(b)T(a,b) = 0.01$$

- Acest lucru se poate realiza prin scalarea elementelor lui T:

$$U^{(1)}(a,b) = \sigma T(a,b), \quad U^{(1)}(a,a) = \sigma T(a,a) + (1-\sigma)$$

$$\sigma = 3/\log(2)$$

- $U^{(1)}$ este tot o matrice de tranziție, astfel că pentru a obține probabilități de tranziție P(b|a,m) în m etape este suficient să se ridice $U^{(1)}$ la puterea m. Folosind $U^{(m)}$ scorul de substituție va fi:

$$PAM_{(m)}(a,b) = \log \frac{U^{(m)}(a,b)}{p(b)}$$

Matrici PAM

Exemplu: PAM_{250}

Valorile calculate sunt rotunjite la cel mai apropiat intreg

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-4	-4	-4	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	-1	-1	-3	0	-1	-1	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	3				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-2	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Matrici PAM

Ce matrice trebuie aleasă?

- Pentru a alinia secvențe despre care se știe că sunt similare se folosesc variante PAM(m) corespunzătoare unor valori mici ale lui m

- Pentru a alinia secvențe cu un grad mai mic de similaritate se folosesc variante PAM(m) corespunzătoare unor valori mari ale lui m

Obs: Matricile PAM nu sunt adecvate pentru alinierea secvențelor divergente (întrucât se bazează pe extrapolarea unor informații colectate de la secvențe înrudite). Din acest motiv au fost introduse matricile de tip BLOSUM

Matrici BLOSUM

BLOSUM = BLocks of Amino Acid **SU**bstitution **M**atrix

[Henikoff&Henikoff, 1992]

- Scorurile se deduc prin observarea frecvențelor substituțiilor în blocuri local aliniate aparținând unor proteine având diferite grade de similaritate (din baza de date BLOCKS (http://blocks.fhcrc.org/ -> http://www.ebi.ac.uk/interpro/))

- Valorile din matrice se calculează folosind aceeași tehnica ca și în cazul matricilor PAM: BLOSUM(i,j)=log(f(i,j)/(f(i)f(j))) / lambda

  (lambda este un factor de scalare folosit pentru a obține valori ușor de prelucrat – de exemplu valori întregi)

- In funcție de gradul de similaritate dintre secvențele utilizate la calculul elementelor matricii există diferite variante de matrici specificate prin indici diferiți

Matrici BLOSUM

BLOSUM = BLocks of Amino Acid **SU**bstitution **M**atrix

[Henikoff&Henikoff, 1992]

- Indicele asociat matricii indică procentul maxim de aminoacizi identici în proteinele în baza cărora se construiește matricea
  - BLOSUM50 s-a construit pornind de la secvențe de aminoacizi ce sunt identice în maxim 50% dintre poziții

- Matricile de indice mic se folosesc la alinierea secvențelor despre care se presupune că nu sunt foarte similare, pe când matricile de indice mare se folosesc în cazul secvențelor presupuse a fi puternic corelate

Matrici BLOSUM

Construire BLOSUM(L)

- Se grupează secvențele în clase astfel încât procentul de elemente identice între secvențele din aceeași clasă este cel puțin L%
- Se calculează F(a,b) numărul de apariții ale perechii (a,b) pe poziții corespunzătoare în secvențe aparținând unor clase diferite (între secvențele aliniate gradul de potrivire este cel mult L%)
- Se calculează

$$p(a,b) = \frac{F(a,b)}{\sum_{(c,d)} F(c,d)}, \quad p(a) = \sum_{b} p(a,b)$$

Matrici BLOSUM

BLOSUM 62

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

s(a,b)=

(log(p(a,b)/(p(a)*p(b)))/lambda

cu rotunjire la cel mai apropiat intreg

lambda=log(2)/2

Algoritmi euristici de aliniere

- Algoritmii de aliniere globală (Needleman-Wunsch) și locală (Smith-Waterman) sunt costisitori în cazul secvențelor lungi (ordinul de complexitate este O(mn) pentru secvențe având lungimile m respectiv n)

- In cazul căutării în baze de date, secvența de interogare are lungimea de ordinul sutelor (eventual miilor) pe când baza de date corespunde unei secvențe cu lungimea de ordinul 10^9 – 10^{10}

- Pentru căutarea unei secvențe într-o bază de secvențe se preferă metode euristice (sub-optimale) care permit identificarea rapidă a secvențelor similare

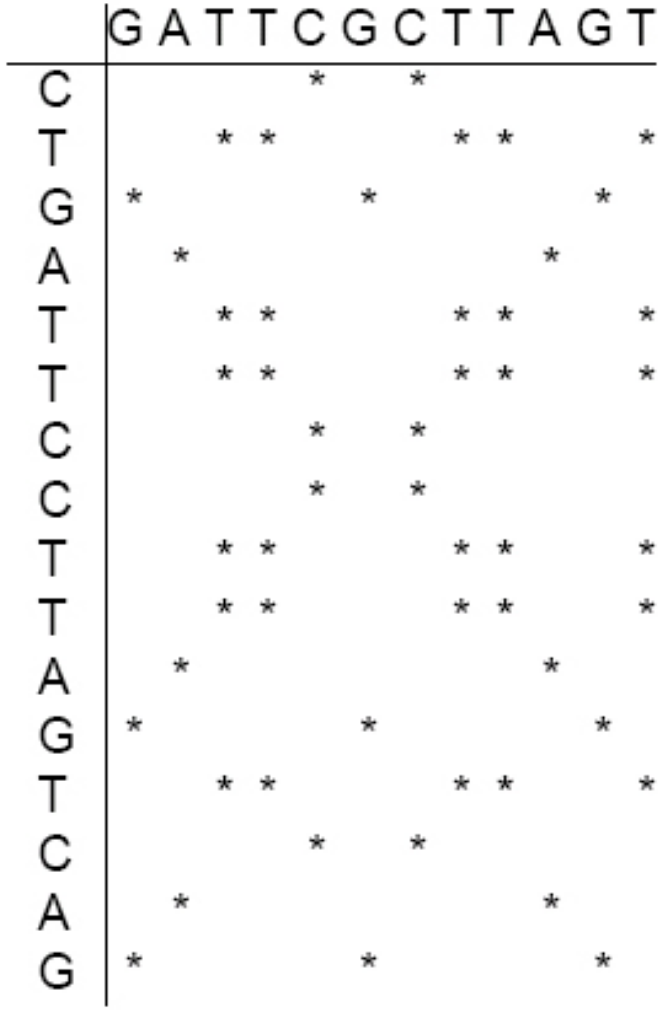
- Metodele euristice se bazează pe ideea identificării unor potriviri scurte dar semnificative (de scor mare) și construirea alinierii prin extinderea acestor potriviri

Algoritmi euristici de aliniere

- Ideea de bază a metodelor euristice de aliniere este cea a filtrării:
  - Lungimea potrivirilor exacte căutate depinde de tipul de secvență (mai scurte la secvențe de aminoacizi și mai lungi la secvențe de nucleotide) și de algoritmul folosit
  - Extinderea potrivirilor exacte se bazează pe calcularea unor scoruri și utilizarea unui prag de acceptare care de asemenea depind de varianta de algoritm
- Din punct de vedere intuitiv potrivirile exacte corespund unor diagonale vizibile in matricea de puncte asociată perechii de secvențe

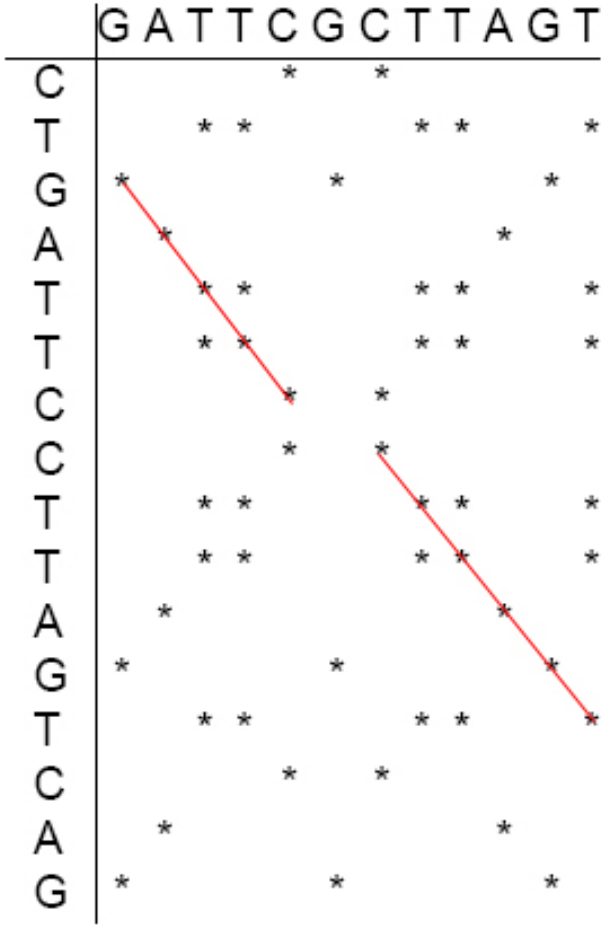
Metode euristice: matrici de puncte

- Matricile de puncte (dot matrix): permit ilustrarea similarității între două secvențe
- Algoritmii euristici permit identificarea diagonalelor corespunzătoare unor potriviri exacte (fără gap-uri) pe care ulterior le combină (atât timp cât pierderea în scorul de potrivire nu este prea mare).
- Obs: in implementarea algoritmilor matricile de puncte nu sunt construite explicit



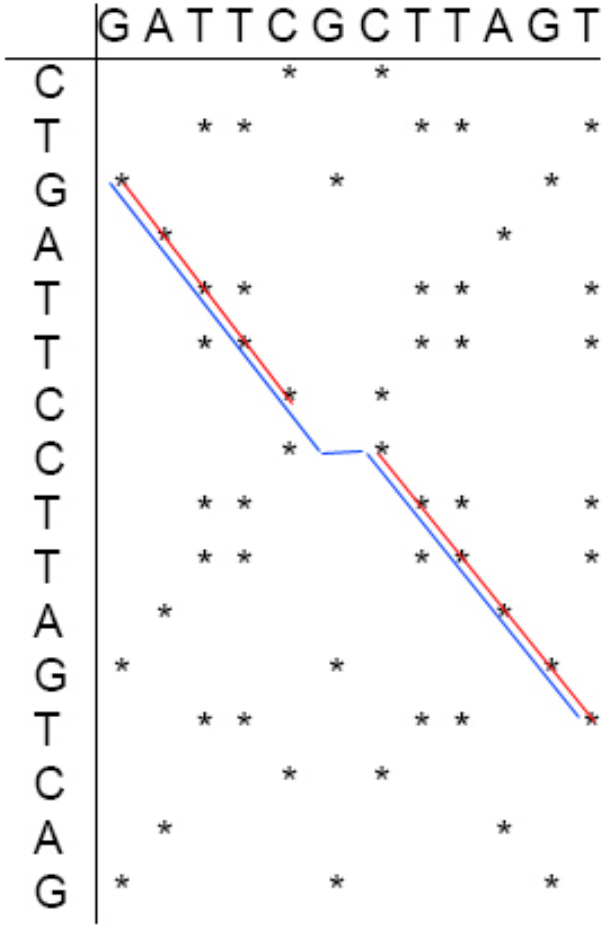
Metode euristice: matrici de puncte

- Se identifică diagonalele având lungimea mai mare decât un anumit prag
- Diagonalele continue (fără întreruperi) indică potriviri exacte



Metode euristice: matrici de puncte

- Se extind diagonalele și se încearcă conectarea lor acceptându-se un număr mic de nepotriviri/inserții/ștergeri
- Concatenarea diagonalelor exprimă potriviri aproximative de-a lungul unor subșiruri mai lungi

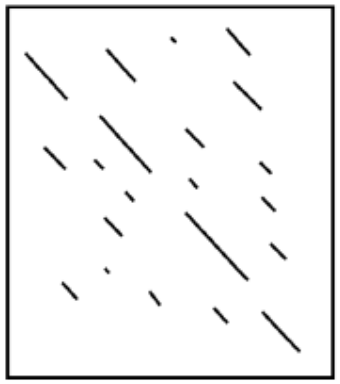


Metode euristice: algoritmul FASTA

- FASTA = Fast Alignment (Lipman & Pearson,1985)
- Caracteristici:
  - A fost primul instrument de căutare în bazele de secvențe
  - Folosește o strategie de “hashing” pentru a găsi potriviri cu secvențe scurte de cate k simboluri:
    - k=2 pentru secvențe de aminoacizi
    - k=6 pentru secvențe de nucleotide;

    Obs: O astfel de secvență este numită k-tuplu
  - Download: www.ebi.ac.uk/fasta33/

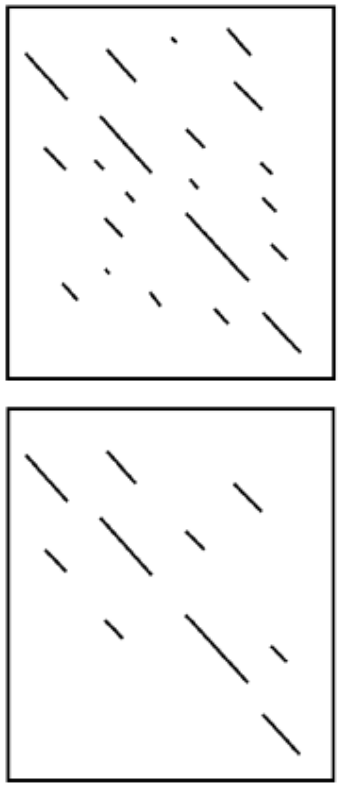
Metode euristice: FASTA



Etape

- **Etapa1**: identifică k-tuplurile comune secvenței de interogare și bazei de date
  - Se construiește un tabel de căutare cu intrări corespunzătoare fiecărui k-tuplu și pozițiile în care acesta se găsește în fiecare dintre secvențe (pentru baza de date acest tabel de căutare se construiește o singură dată într-o etapă de preprocesare)
  - Se identifică k-tuplurile comune celor două secvențe

Metode euristice: FASTA



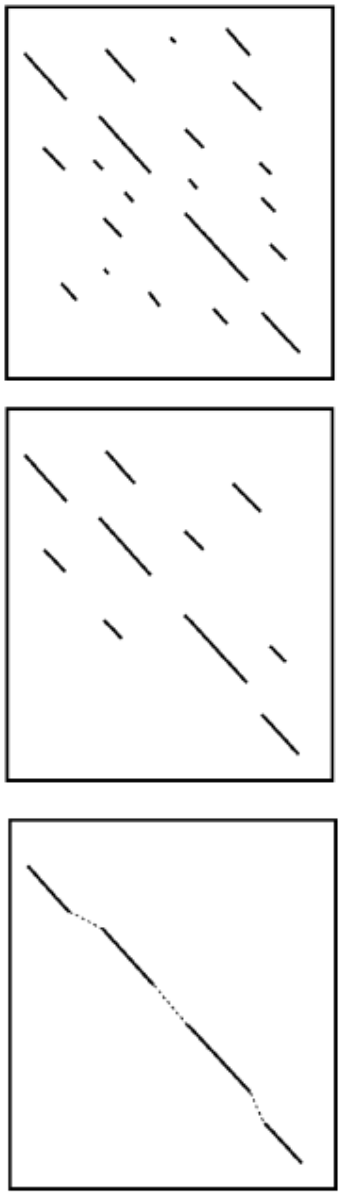
Etapa 2:

- Se ordonează k-tuplurile comune după i-j (i e indicele de start în secvența 1, j e indicele de start în secvența 2); în felul acesta k-tuplurile ce fac parte din aceeași diagonală vor fi pe poziții apropiate
- Se concatenează k-tuplele comune consecutive (constituindu-se diagonale mai lungi) penalizându-se spațiile libere dintre ele care corespund nepotrivirilor (scorul secvenței concatenate e suma lungimilor secvențelor de potriviri minus numărul de nepotriviri)
- Se selectează secvențele comune (constituite din k-tuple comune consecutive ce formează diagonale în matricile de puncte) având scorul de potrivire cel mai mare (de exemplu se selecteaza primele 10 astfel de potriviri)

Metode euristice: FASTA

Etapa 3:

- Se încearcă extinderea potrivirilor exacte prin introducerea de gap-uri
  - Unește segmentele de pe diagonale “învecinate”
  - Problema unirii diagonalelor poate fi formulată ca o problemă de identificare a unei căi de scor maxim într-un graf în care:
    - Nodurile corespund diagonalelor identificate la pasul anterior (eticheta unui nod este scorul asociat diagonalei)
    - Muchiile unesc nodurile ce corespund diagonalelor ce ar putea fi concatenate prin introducere de gap-uri (nodul p se concatezează cu nodul q dacă indicii de linie și coloană ai diagonalei corespunzătoare lui p sunt fiecare dintre ei mai mici decât indicii elementelor din diagonala corespunzătoare lui q)
    - Scorul unei muchii este negativ si proporțional cu numărul de gap-uri care s-ar introduce



Metode euristice: FASTA

- **Etapa 4**.
  - Pornind de la diagonala de scor maxim identificată în Etapa 3 se aplică un algoritm de aliniere locală (Smith Waterman) pentru o bandă din matricea de puncte fixată în jurul diagonalei
  - In felul acesta există șansa să se găsească o aliniere locală de scor mare în vecinătatea diagonalei identificate la Etapa 3

Metode euristice: FASTA

- **Etapa 5**. Analiza statistică a similarității.

Idee: se estimează cât este de plauzibil ca scorul de potrivire obținut să corespundă unor secvențe necorelate (de exemplu secvențe generate aleator). In acest scop se generează secvențe aleatoare, se calculează scorul mediu corespunzător acestora și se utilizează pentru calculul statisticii Z.

- Statistica Z (Z-score)
  - Măsoară abaterea față de scorul mediu al unei căutări
  - Scorul mediu corespunde interogărilor care conduc la secvențe necorelate cu cea de interogare
  - Potrivirea este considerată cu atât mai semnificativă cu cât scorul este mai mare
  - Exemple de interpretare a statisticii Z:
    - Z>15 - potrivire foarte semnificativă
    - 5<=Z<=15 – potrivire destul de semnificativă
    - Z<5 – potrivire puțin plauzibilă

Metode euristice: BLAST

- **B**asic **L**ocal **A**lignment **S**earch **T**ool
  [Altschul, Gish, Lipman, Miller, Myers (1990)]

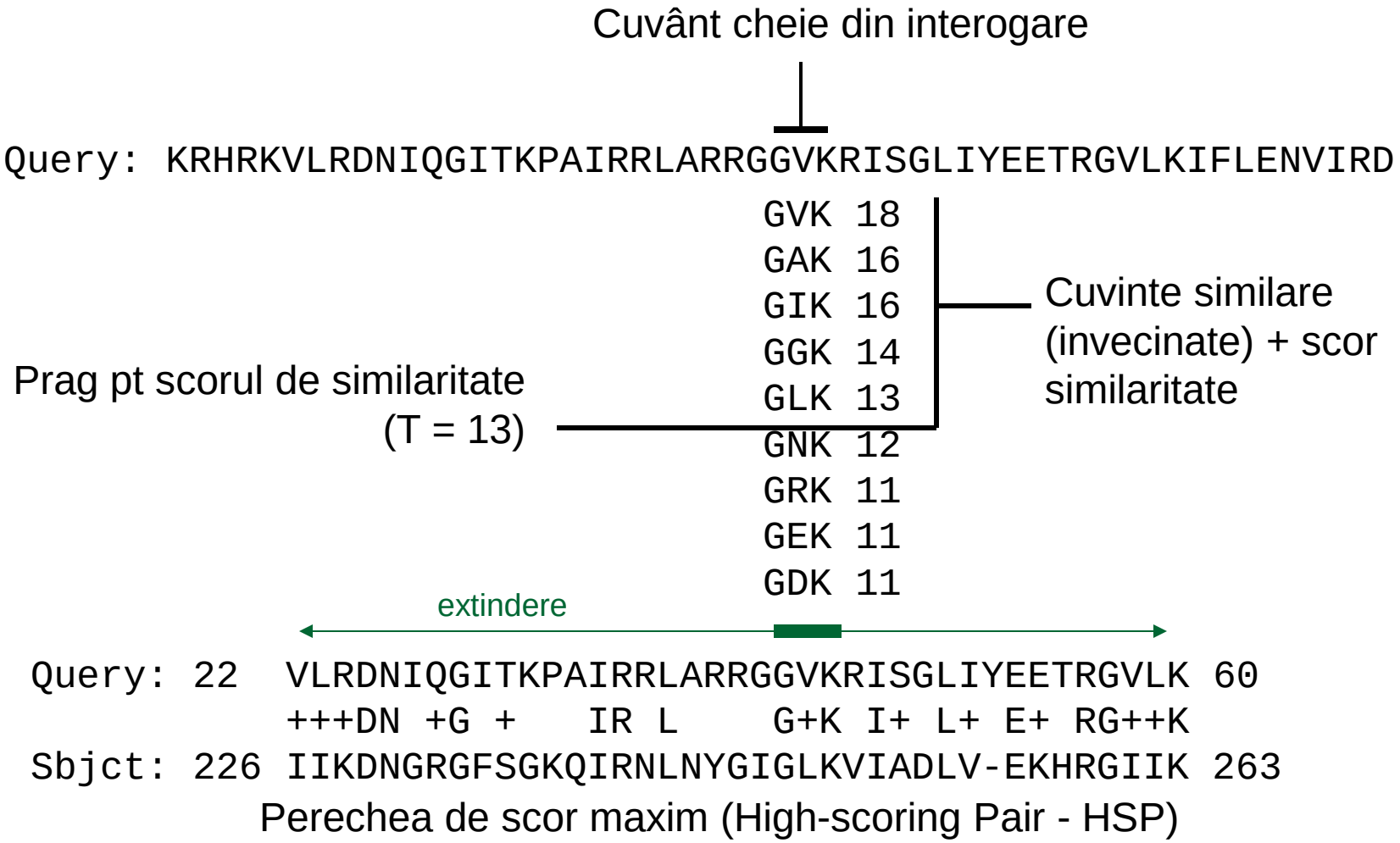
- http://www.ncbi.nlm.nih.gov/blast/Blast.cgi

- Varianta inițială:
  - Identifică segmente fără gap- uri (cu scor mare de similaritate)
  - Se bazează pe o analiza statistică a similarității dintre segmente care permite discriminarea între secvențele cu adevărat similare și cele similare din întâmplare
  - Folosește o matrice de scor clasică (ex: PAM250 sau BLOSUM62)

Metode euristice: BLAST

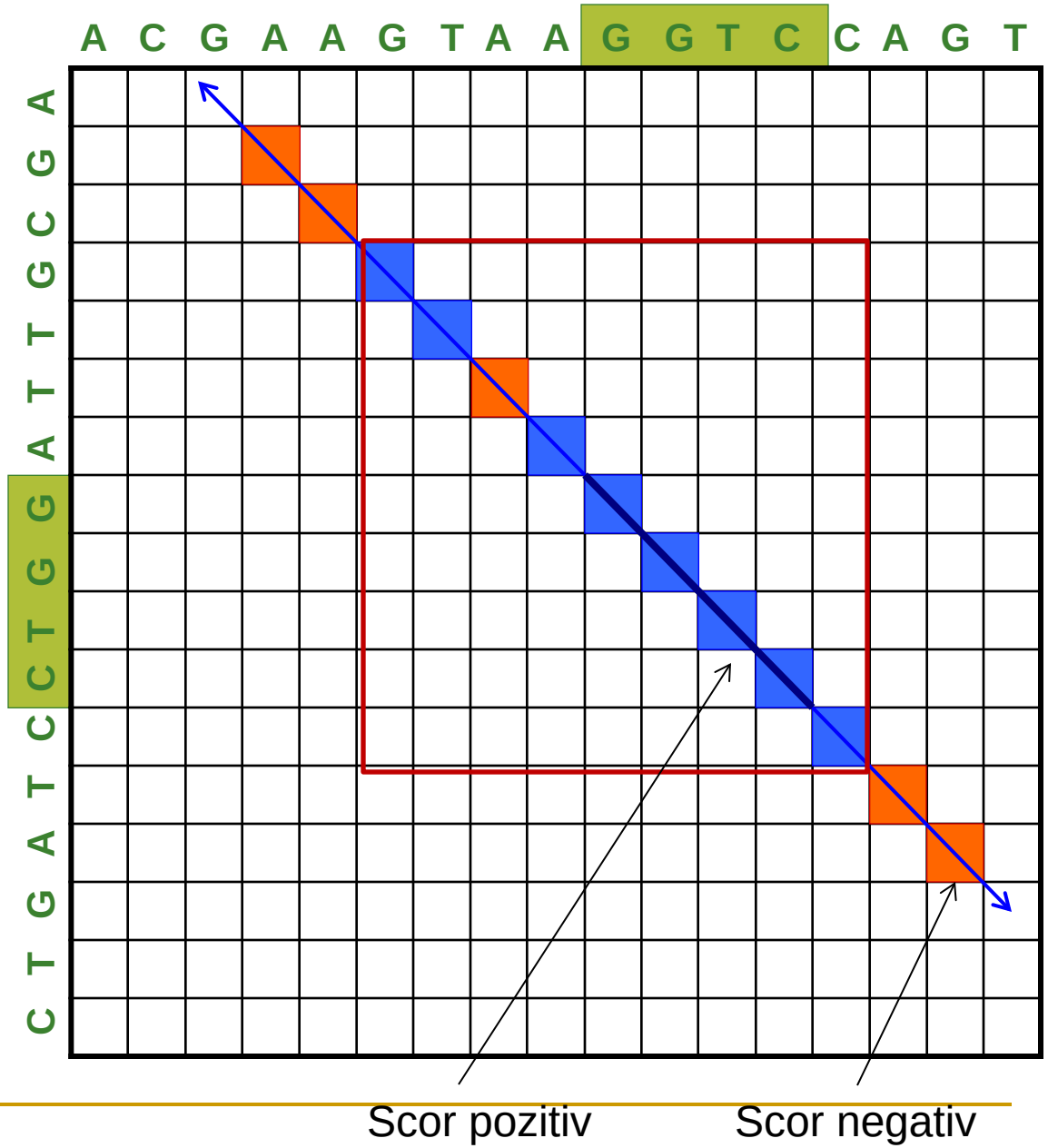
- Etape:
  - **Etapa 1:** Crează o *listă cu cuvinte* din secvența de interogare precum și cuvinte suficient de similare cu acestea. Un cuvânt conține cca 3 simboluri în cazul secvențelor de aminoacizi și 11 în cazul secvențelor de nucleotide. Similaritatea dintre cuvinte se calculează folosind o matrice de scor și cuvintele se consideră similare doar dacă scorul depășește un prag.
  - **Etapa 2:** Se caută aceste cuvinte în baza de date;
  - **Etapa 3:** Se extind potrivirile de la nivelul cuvintelor până când se identifică un *segment local maximal* (se caracterizează prin faptul că scorul nu poate fi mărit nici prin adăugarea nici prin eliminarea de elemente).
  - **Etapa 4:** Potrivirile astfel identificate se ordonează descrescător după scor și pentru fiecare se estimează *semnificația statistică* a similarității.

Metode euristice: BLAST



BLAST

- *w* = 4 (lungime cuvânt)
- Cuvântul cu care se potrivește exact **GGTC**
- Se extind diagonalele până când scorul de potrivire devine mai mic decât 50% din scorul potrivirii inițiale sau până când scorul începe sa descrească
- Rezultat
  - **GTAAGGTCC**
  - **GTTAGGTCC**



Sursa: slide-uri Serafim Batzoglou (Stanford)

BLAST

Analiza statistică a potrivirii

Scop: stabilește dacă potrivirea este determinată de existența unei similarități reale între secvența de interogare și cea din baza de date sau este doar întâmplătoare

Instrument: test statistic

Ipoteza nulă: cele două secvențe sunt independente

$$P(j,k) = p_j p'_k$$

Probabilitatea ca simbolul k să apară în a doua secvență

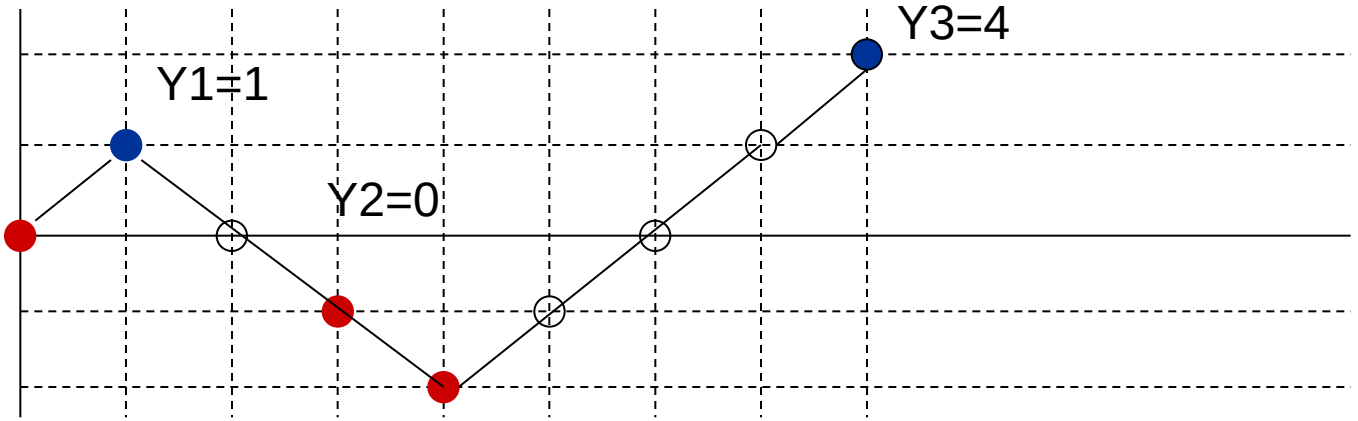
Probabilitatea ca simbolul j sa fie aliniat cu simbolul k

Probabilitatea ca simbolul j să apară în prima secvență

BLAST

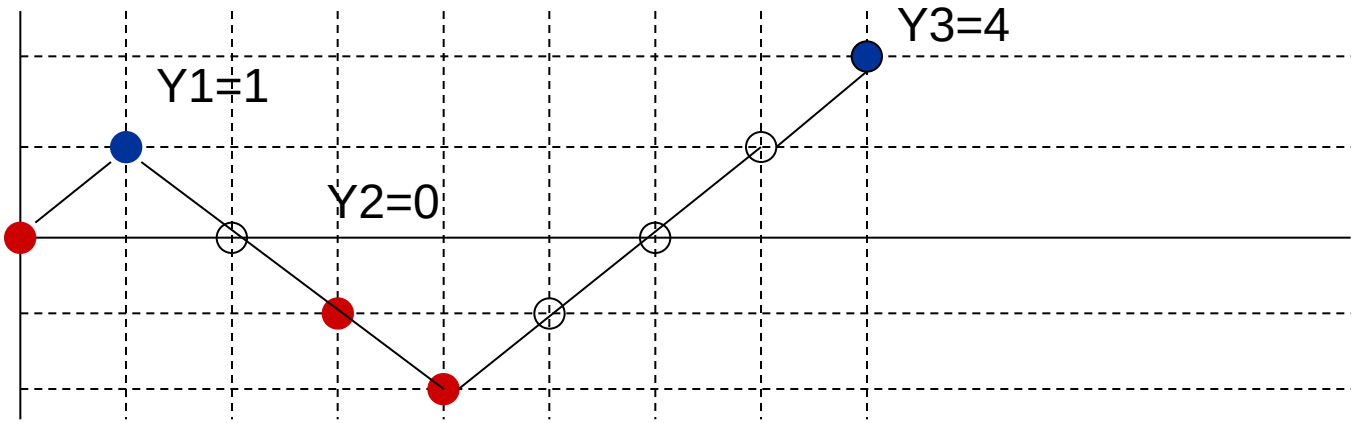
- Statistica testului.
  - Fie S(j,k) scorul substituirii lui j cu k
  - Scorul corespunzător poziției i din cele două secvențe este suma scorurilor asociate perechilor de elemente până la poziția i inclusiv. Evoluția scorului în funcție de i poate fi descrisă printr-un proces de tip mers aleator (random walk)
  - Exemplu: GGTGTAGA
    GACCTAGA

Fiecare potrivire este recompensată cu 1; fiecare nepotrivire este penalizată cu 1



BLAST

- Statistica testului.
  - Yi reprezintă “înălțimea maximă” dintre două puncte de descreștere (i și i+1) – un punct este considerat de descreștere dacă atinge un nivel mai mic decât cele atinse până în momentul respectiv (pct roșu pe grafic)
  - Statistica testului este Y_{max}= max{Y1,Y2,…}



BLAST

- Statistica testului.
  - Daca ipoteza nulă este adevarată atunci Y_{max} are repartiția

$$P(Y_{\max} \geq S) \cong 1 - e^{-y}$$

$$y = Kmne^{-\lambda S}$$

$$\sum_{j,k} p_j p_k^{'} e^{\lambda S(j,k)} = 1$$

- Observații:
  - S(j,k) este scorul potrivirii dintre elementele de pe pozițiile j respectiv k din cele două secvențe
  - m și n sunt lungimile secvențelor; K poate fi interpretat ca o măsură a similarității componentelor (aminoacizilor); lambda poate fi interpretat ca un factor de scalare asociat matricii de scor
  - K și lambda se estimează numeric (valorile depind de matricea cu scoruri de substituție utilizată)
  - Daca matricea e BLOSUM62 atunci estimările pt K și lambda sunt: K=0.04, lambda=0.254

BLAST

Interpretarea valorilor statistice furnizate de catre pachetele software care implementează algoritmi de tip BLAST:

- E-value (expectation value): E=K*m*n*exp(-lambda S)
  - Numărul mediu de potriviri având scorul mai mare decât S care s-ar obține în mod aleator (în ipoteza că nu ar exista nici o corelație între secvența de interorgare și baza de date)
  - Interpretare:
    - Daca $E<10^{-50}$ secvențele sunt probabil identice
    - Daca $10^{-50}<E<0.01$ atunci secvențele sunt semnificativ corelate
    - Daca $0.01<E<10$ atunci potrivirea este nerelevantă însă poate sugera o înrudire îndepărtată
    - Daca $E>10$ secvențele sunt probabil necorelate

- Obs: E-valoarea este influențată de dimensiunea bazei de date
  Pt. a evita acest lucru se folosește și un alt indicator: bit score

BLAST

Interpretarea valorilor statistice furnizate de catre pachetele software care implementează algoritmi de tip BLAST:

- Bit-score: este o variantă normalizată a măsurii similarității dintre secvențe

  S'=(lambda*S-log K)/log 2

  unde S este scorul clasic de similaritate iar lambda și K sunt ca în definiția de la E-valoare

- Interpretare: cu cât S' este mai mare cu atât este mai semnificativă potrivirea

BLAST

Observație:

- în varianta clasică BLAST nu permite gap-uri
- Varianta propusă în [S. F. Altschul et al. Gapped BLASTand PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17):3389–402, 1997] permite includerea de gap-uri folosind o idee similară celei de la FASTA dar fără a restricționa căutarea într-o bandă a matricii de scor ci acceptând deviația între diagonalele corespunzătoare potrivirilor atât timp cât scorul nu scade sub un prag

BLAST

Variante de implementare

- blastn: Nucleotide-nucleotide
- blastp: Protein-protein
- blastx: Translated query vs. protein database
- tblastn: Protein query vs. translated database
- tblastx: Translated query vs. translated database (6 frames each)
- PSI-BLAST – determină membrii unei familii de proteine sau construiește o matrice specifică de scor.
- Megablast: - caută după secvențe mai lungi, cu puține diferențe
- WU-BLAST: (Wash U BLAST) – varianta optimizată