EURISTICI DE ALINIERE A SECVENTELOR (*PAIRWISE* ALIGNMENT)

PARTEA I

In genomica este important sa gasim cele mai bune potriviri **locale** intre anumite sub-secvente dintr-o gena (*query*) si sub-secvente din genomurile referentiale de interes. Este, evident vorba de o **aliniere locala**, iar alinierea globala intre o secventa *query* de cateva mii de nucleotide si secventa unui cromozom este practic un non-sens.

Aplicarea algoritmului Smith-Waterman (SW) de aliniere locala nu este insa eficienta atunci cand secventa *query* are cateva sute/mii de nucleotide iar secventele de referinta din bazele de date au dimensiuni de ordinul milioanelor de nucleotide. Faptul ca algoritmul SW identifica aliniamentul optim prin efectuare unor calcule exhaustive are doua dezavantaje: consum mare de resurse de calcul si de timp, ceea ce, in varianta sa clasica, il face ineficient in genomica.

In consecinta, au fost dezvoltate euristicii de alinere, care efectueaza calcule aproximative ale alinierilor locale, ceea ce conduce la obtinerea rapida a unei liste de aliniamente grupate dupa scorurile de aliniere.

Exemple de euristici de aliniere:

BLAST (https://blast.ncbi.nlm.nih.gov/Blast.cgi)

SSAHA (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC311141/)

BLAT (https://genome.ucsc.edu/cgi-bin/hgBlat)

Genome ARTIST (http://www.genomeartist.ro)

In continuare, vor fi prezentate cateva dintre principiile care stau la baza functionarii euristicii **Genome ARTIST**. Ulterior, vom acorda atentie euristicilor BLAST si BLAT.

Este util sa retinem ca Bioinformatica nu da raspunsuri definitive, ci ne ajuta sa dezvoltam sau sa verificam scenarii de lucru. Biologul este cel care, coreland datele de bioinformatica cu cele experimentale, decide care este cel mai semnificativ aliniament (din punct de vedere biologic, nu al scorului de aliniere).Teoretic, primul aliniament din lista de rezultate a unei eurisitici este, adesea, cel mai util. Uneori insa, se intampla ca un aliniament de pe locul doi sau trei din lista sa aiba mai mult sens din punct de vedere biologic, evolutionist.

Euristicile de aliniere *pairwise* (sunt aliniate doar cate doua secvente de nucleotide sau de aminocizi) nu garanteza obtinerea **aliniamentului local optim** insa, in mod frecvent, gasesc solutii foarte bune. Altfel spus, euristicile de aliniere sunt concepute pentru identificarea unor aliniamente **locale** cat mai bune (cat mai lungi si cu scor un de aliniere cat mai mare). De fapt, abordarea moderna este **implementarea algoritmului SW intr-o euristica**, insa doar pentru a verifica/rearanja alinimentele initiale generate de euristica respectiva.

Astfel, algoritmul SW este solicitat doar pentru re-alinierea unor aliniamente locale de dimensiuni relativ reduse, de ordinul zecilor sau sutelor de nucleotide, ceea ce este perfect fezabil chiar si pentru un laptop cu performante de calcul medii. Atunci cand secventa *query* se potriveste perfect cu o secventa tinta din genomul de referinta, nu sunt probleme de calcul. Cand insa comparam o gena de la om cu una de la *Drosophila melanogaster*, chiar daca in genomul insectei exista o gena omoloaga, constatam ca cele doua gene nu sunt identice, ci doar omoloage structural. Adica, in aliniamentul respectiv apar numeroase *mismatch*-uri si indeluri. Intr-o astfel de situatie se dovedesc utile eurisiticile de aliniere. Cum reusesc acestea sa calculeze aliniamente care contin blocuri de potrivire perfecta ce alterneaza cu *mismatch*-uri si indeluri?

Pentru a castiga timp atunci cand are loc alinierea intre secventa *query* si un genom de referinta, este foarte util sa procesam in prealabil secventa/secventele de referinta incarcate in baza de date a euristicii. Astfel, sa luam ca exemplu sceventa de referinta a unui cromozom de la *D. melanogaster*, de exemplu cromozomul X. Atunci cand acesta este incarcat in baza de date a

euristicii **Genome ARTIST**, secventa respectiva de nucleotide este prelucrata in vederea obtinerii unui tabel *hash*.

Un tabel *hash* (*hash table*) este **o structura de date** care permite accesarea rapida a elementelor componente in care este impartit genomul. Concret, fiecare cromozom din alcatuirea unui genom este impartit (*hashed*) in oligomeri de lungime egala *k* si este generat un tabel *hash* specific. Acesti oligomeri se mai numesc si cuvinte (*words*) sau *k*-meri. Dupa stabilirea valorii parametrului *k* (**in cazul GA, *k* = 10**), este utilizata o modalitate de ordonare (de exemplu, in ordine alfabetica) a *k*-merilor.

Astfel, fiecare *k*-mer distinct, denumit si cheie *hash* (*hash key*), este convertit intr-un numar.

Valorile numerice individuale obtinute prin conversia *k*-merilor (cheilor *hash*) poarta denumirea de valori *hash*. Vom observa in exemplul concret care urmeaza ca ordinea alfabetica a *k*-merilor este convertita in ordine crescatoare a valorilor *hash* (0, 1, 2, 3, 4, 5, etc.)

Aceasta conversie este identica pentru orice cromozom, indiferent de lungimea sa, sau de specia de la care provine.

Ceea ce este **particular/variabil** in tabelul *hash* sunt **valorile specifice** asociate fiecarui *hash* (care este un alt mod de a prezenta cheia *hash*/*k*-merul).

Valorile specifice asociate reprezinta **locatiile *k*-merului respectiv** in secventa de referinta a cromozomului. In tabelul *hash* este trecuta doar coordonata cea mai mica a unui *k*-mer, daca acesta exista concret in secventa de referinta care este indexata.

De exemplu, daca decamerul TACACGCGTC are in secventa de referinta coordonatele nucleotidice 100-109, in tabel va fi notata doar coordonata 100, care este astfel o **valoare asociata *hash*-ului corespunzator cheii de *hash* TACACGCGTC**. Daca acelasi *k*-mer TACACGCGTC se regaseste si in intervalul de referinta 2367-2376, in tabel apare valoare 2367, asociata aceluiasi *hash*, etc.

In exemplul prezentat mai jos, aceste valori specifice (coordonate) apar in coloana 5 a tabelului *hash*.

In cazul euristicii **Genome ARTIST**, valoarea *hash* este mereu aceeasi pentru un anumit decamer (cheie *hash*) indiferent de specia de la care provine secventa de referinta, dar valorile asociate acesteia variaza in functie de secventa de referinta respectiva.

Tabelul *hash* al unei secvente de referinta este construit o singura data si este depozitat pentru accesare ulterioara, atrunci cand este interogat cu o secventa *query*.

LUNGIMEA *K*-MERILOR DETERMINA DIMENSIUNILE TABELULUI *HASH*

Cu cat valoarea *k* este mai mica, cu atat exista mai putini *k*-meri distincti. Logic, un *k*-mer scurt se regaseste in foarte multe pozitii intr-o secventa de referinta de dimensiunea unui cromozom, fie el si bacterian. De exemplu, pentru $k = 2$ exista 4^2 *k*-meri distincti, adica **16**, iar pentru $k = 3$ exista 4^3 *k*-meri distincti (acesta este si motivul pentru care exista exact **64** de codoni in tabelul codului genetic).

Codul genetic nu este, evident, un tabel *hash,* insa poate fi generat un tabel *hash* pentru o secventa de referinta utilizand trimeri.

IMPORTANT: Pentru valori relativ mari ale *k*, nu toti *k*-merii teoretic posibili exista efectiv intr-o anumita secventa de referinta, adica intr-un anumit cromozom/genom.

Genome ARTIST opereaza cu *k*-meri de 10 nucleotide, ceea ce inseamna ca in tabelul *hash* sunt indexati 4^10 = 1.048.576 decameri distincti. Acestia vor avea valori *hash* de la zero la 1.048.575, intrucat, in limbajul computerului, numaratoarea incepe de la zero. Astfel, valoarea *hash* a decamerului AAAAAAAAAA = 0, iar valoarea *hash* a decamerului TTTTTTTTTT = 1.048.575.

Identificarea decamerilor intr-o secventa de referinta reala e realizata de **Genome ARTIST** dupa strategia *overlapped*, adica se inainteaza cu pasi de cate o nucleotida pentru a gasi *k*-merii din

referinta. Din acest motiv, **doi decameri localizati succesiv in secventa de referinta au in comun 9 nucleotide**.

Alte euristici (BLAT, SSAHA), fie utilizeaza un grad de suprapunere mai mic, fie nu aplica strategia *k*-merilor suprapusi, astfel incat indexarea genomului se face mai rapid. Exista insa si un dezavantaj: unii *k*-meri care exista efectiv in genomul nu sunt vazuti de o astfel de euristica.

De exemplu, pentru secventa de referinta:

5'TAGACGCGTCTCTCGCGTC3'

Daca **nu** se opereaza cu *k*-meri *overlapped*, doar *k*-merul TAGACGCGTC (care are coordonata 1) este identificat. Sub-secventa urmatoare este TCTCGCGTC (cu coordonata 11) si are doar 9 nucleotide, deci nu este un decamer. Insa, daca sunt utilizati *k*-meri *overlapped* (care au 9 nucleotide comune), sunt identificati si alti *k*-meri, respectiv: AGACGCGTCT (pozitia 2), GACGCGTCTC (pozitia 3), etc.

Observati ca primele 9 nucleotide ale *k*-merului AGACGCGTCT sunt aceleasi cu ultimele 9 nucleotide ale *k*-merului TAGACGCGTC.

Sa presupunem ca avem o secventa *query* TCTCTCGCGC (pentru simplitatea exemplului, am ales o secventa *query* de 10 nucleotide) pe care intentionam sa o aliniem cu secventa de referinta. Aceasta nu va putea fi asociata insa cu nici un *k*-mer din secventa de referinta de mai sus **daca pentru indexarea acesteia au fost utilizati doar *k*-meri non-*overlapped***. In abordarea *non-overlapped*, decamerul TCTCTCGCGC nu are nici o locatie asociata *hash*-ului respectiv in tabelul *hash* al secventei de referinta, chiar daca are o pozitie rezervata in tabel si a fost convertit in valoare *hash*.

Dar daca utilizam varianta *overlapped*, secventa *query* este asociata cu pozitia **9** din secventa de referinta, intrucat *k*-merul **TCTCTCGCGT** este vazut de **Genome ARTIST**, iar *hash*-ul acestui decamer va avea asociata valoarea **9**.

EXEMPLU DE TABEL *HASH*

Construim tabelul *hash* al genomul de referinta ipotetic:

5’GCATTCGTTGCTACTAGGTT3’

Genomul este un text scris cu 4 litere: A, C, G, T. Aceste litere pot fi convertite in numere, fie in sistemul binar, unde exista doar biti: 0 si 1, fie in sistemul cuaternar, unde exista doar 4 cifre: 0, 1, 2 si 3.

Conventii de conversie a nucleotidelor (litere) in numere:

NUCLEOTIDA	BINAR	CUATERNAR
A	00	0
C	01	1
G	10	2
T	11	3

Lucram cu dimeri ($k = 2$) si strategia dimeri *overlapped*.

Tabel *hash* pentru $k = 2$ ($2^4 = 16$ dimeri distincti):

***k*-mer (*HASH* KEY)**	**BINAR**	**CUATERNAR**	**ZECIMAL *HASH* (H)**	**Valoari asociate lui H**	**Pozitia in query a dimerilor**
AA	0000	00	0	-	-
AC	0001	01	1	13	5
AG	0010	02	2	16	-
AT	0011	03	3	3	1, 8
CA	0100	10	4	2	-
CC	0101	11	5	-	-
CG	0110	12	6	6	3, 6
CT	0111	13	7	11, 14	-
GA	1000	20	8	-	4, 7
GC	1001	21	9	1, 10	-
GG	1010	22	10	17	-
GT	1011	23	11	7, 18	-
TA	1100	30	12	12, 15	-
TC	1101	31	13	5	2
TG	1110	32	14	9	-
TT	1111	33	15	4, 8, 19	-

Pentru a converti o cheie *hash* (*k*-mer) in valoare *hash* este utilizata urmatoarea formula:

$$\sum_{i=0}^{n-1} (\text{valoare } i \times \text{baza}^{i})$$

Unde n = *k* ; baza = 2 pentru binar si respectiv 4 pentru cuaternar; *i* = pozitia fiecarei cifre, numaratoarea incepe de la zero, de la dreapta la stanga.

EXEMPLU pentru trimerii ATT si TTT:

	543210	210
ATT =	001111	033
TTT =	111111	333

H = valoarea *hash*

H = ATT (001111) = 0x2^5 + 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 = 0 + 0 + 8 + 4 + 2 + 1 = **15**
H = ATT (033) = 0x4^2 + 3x4^1 + 3x4^0 = 0 + 12 + 3 = **15**

H = TTT = 1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 = 32 + 16 + 8 + 4 + 2 + 1 = **63**
H = TTT = 3x4^2 + 3x4^1 + 3x4^0 = 48 + 12 + 3 = **63**

Se observa ca H are aceeasi valoare pentru un *k*-mer anume, indiferent ca a fost utilizat sistemul binar sau pe cel cuaternar pentru simbolizarea nucleotidelor.

Daca luam in considerare secventa *query* ipotetica:

5’ATCGACGAT3’

Se observa ca dimerii *overlapped* posibili din *query* sunt: **AT, TC, CG, GA, AC, CG, GA si AT.** Adica, pentru un *query* de 9 nucleotide avem 8 dimeri posibili. Daca lungimea *query* este L, atunci numarul de *k*-meri existenti intr-o secventa de lungime L este dat de formula **n = L – *k* +1**.

In cazul nostru, 9 – 2 +1 = 8 dimeri, dar se observa ca **AT, CG si GA** exista de doua ori in secventa *query*. Cei 8 *k*-meri *overlapped* dintr-o secventa *query* de 9 nucleotide nu trebuie sa fie diferiti sau identici ca secventa. Valoarea n = 8 reprezinta mai degraba 8 posibilitati, 8 pasi de cate 2 nucleotide cu care **Genome ARTIST** imparte o secventa *query* in partile ei componente atunci cand *k* = 2.

Coloana marcata cu galben **nu** face parte din tabelul *hash* al acestui genom ipotetic.

Exercitiu:

Care este valoarea *hash* a decamerului TCTCTCGCGC ?