ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ ΙΙ

Δυναμικός Προγραμματισμός

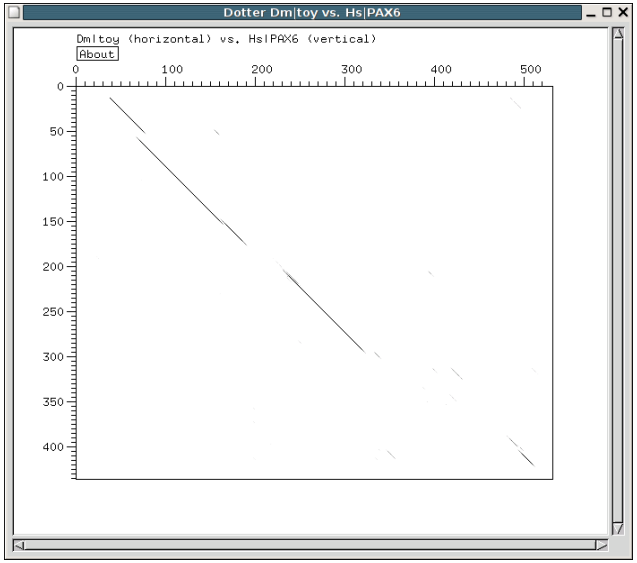
Παντελής Μπάγκος

Δυναμικός Προγραμματισμός

- Στοίχιση (τοπική-ολική)
- RNA secondary structure prediction
- Διαμεμβρανικά τμήματα
- Hidden Markov Models
- Άλλες εφαρμογές

Στοίχιση

- Ολική
- Τοπική
- Ειδικές περιπτώσεις

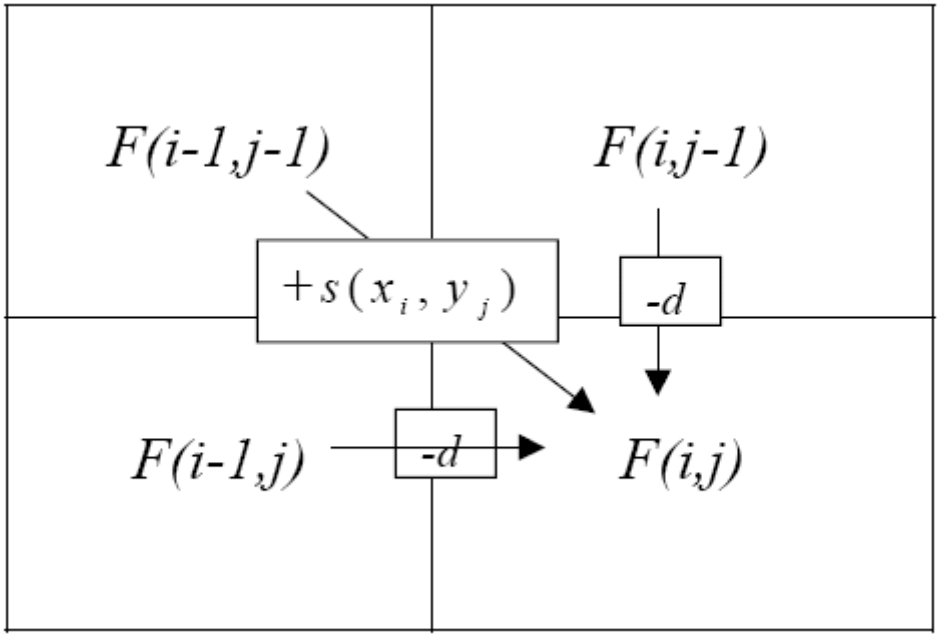



```
α)
>P01922|HBA_HUMAN  GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL
                   G+ +VK HGKKV  A ++ +AH+D++    + LS+LH  KL
>P02023|HBB_HUMAN  GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL

β)
>P01922|HBA_HUMAN  GSAQVKGHGKKVADALTNA-----VAHVDDMPNALSALSDLHAHKL
                    + +++ H  KV   +  A     V  V      L  L  +H  K
>P02240|LGB2_LUPLU NNPELQAHAGKVFKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSKG

γ)
>P01922|HBA_HUMAN  GSAQVKGHGKKVADALT----NAVAHVDDMPNALSALSD----LHAHKL
                         G G  V D+LT        H  D+  A +AL D      AH+
>P91253|GTS7_CAEEL ------GSGYLVGDSLTFVDLLVAQHTADLLAANAALLDEFPQFKAHQE
```

Δυναμικός προγραμματισμός



Δυο περιπτώσεις στοιχίσεων

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j),\\F(i-1,j)-d,\\F(i,j-1)-d\end{cases}$$

$F(i,0)=-id,$
$F(0,j)=-jd$

$$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j),\\F(i-1,j)-d,\\F(i,j-1)-d,\\0\end{cases}$$

$F(i,0)=0,$
$F(0,j)=0$

Ποινές για τα κενά (gap penalties)

Απλή ποινή για τα κενά:

$$\gamma(g) = -gd$$

Σύνθετη ποινή για τα κενά:

$$\gamma(g) = -d - (g-1)e$$

Παράδειγμα

Έστω δυο ακολουθίες:

$$\mathbf{x} = AAGTTAGCAG$$

$$\mathbf{y} = CAGTATCGCA$$

Αν έχουμε για τα κενά:

$$s(x_i, y_i) = \begin{cases} 1, & \text{αν} \quad x_i = y_i \\ -1, & \text{αν} \quad x_i \neq y_i \end{cases}$$

$$d = 1$$

Τότε η καλύτερη ολική στοίχιση θα είναι:

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

Ολική στοίχιση…

	-	*A*	*A*	*G*	*T*	*T*	*A*	*G*	*C*	*A*	*G*
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
C	-1	-1	-2	-3	-4	-5	-6	-7	-6	-7	-8
A	-2	0	0	-1	-2	-3	-4	-5	-6	-5	-6
G	-3	-1	-1	1	0	-1	-2	-3	-4	-5	-4
T	-4	-2	-2	0	2	1	0	-1	-2	-3	-4
A	-5	-3	-1	-1	1	1	2	1	0	-1	-2
T	-6	-4	-2	-2	0	2	1	1	0	-1	-2
C	-7	-5	-3	-3	-1	1	1	0	2	1	0
G	-8	-6	-4	-2	-2	0	0	2	1	1	2
C	-9	-7	-5	-3	-3	-1	-1	1	3	2	1
A	-10	-8	-6	-4	-4	-2	0	0	2	4	3

A A G T – T A G C A G

C A G T A T C G C A –

Τοπική στοίχιση…

	-	A	A	G	T	T	A	G	C	A	G
-	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	1	0	0
A	0	1	1	0	0	0	1	0	0	2	1
G	0	0	0	2	1	0	0	2	1	1	3
T	0	0	0	1	3	2	1	1	1	0	2
A	0	1	1	0	2	2	3	2	1	2	1
T	0	0	0	0	1	3	2	2	1	1	1
C	0	0	0	0	0	2	2	1	3	2	1
G	0	0	0	1	0	1	1	3	2	2	3
C	0	0	0	0	0	0	0	2	4	3	2
A	0	1	1	0	0	0	1	1	3	5	4

A G T – T A G C A

A G T A T C G C A

Άλλοι αλγόριθμοι

- Υπάρχουν επίσης ειδικές περιπτώσεις στοίχισης (π.χ. προσαρμογή)
- Θέλουμε δηλαδή να εντοπίσουμε, μια μικρή ακολουθία αν συναντάται σε μια μεγαλύτερη

  Έστω ότι θέλουμε να ανιχνεύσουμε αν στην αλληλουχία του γονιδίου *lacI* της *E.coli* υπάρχει η γνωστή αλληλουχία του υποκινητή (promoter). Έστω ακόμα ότι το τμήμα του γονιδίου έχει αλληλουχία:

  $$\mathbf{x} = TCGCGGTATGGCATGATAGCGCCCGGAA$$

  και η αλληλουχία του υποκινητή είναι

  $$\mathbf{y} = TATAAT$$

συνέχεια…

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j), \\ F(i-1, j) - d, \\ F(i, j-1) - d \end{cases}$$

$F(i,0) = -id$

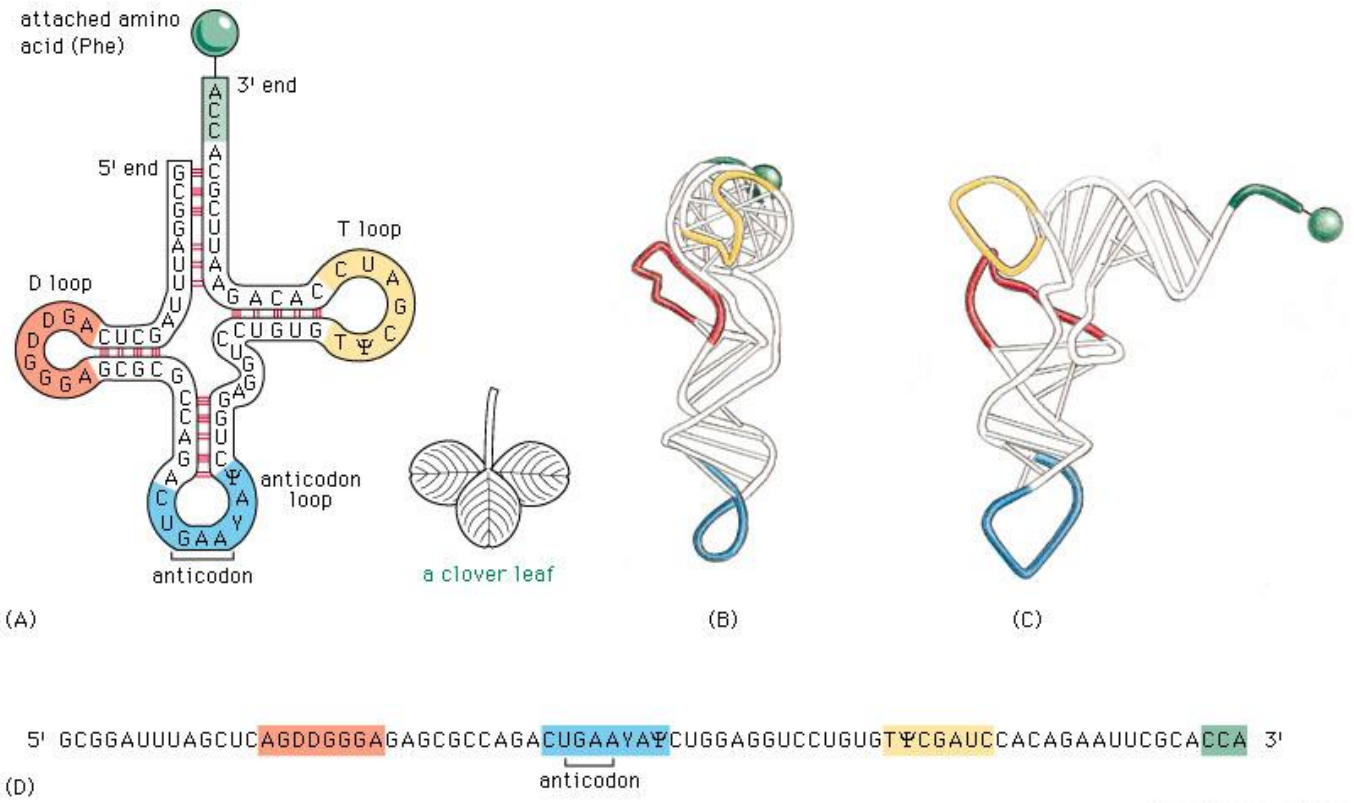
$F(0,j) = 0$.

	T	C	G	C	G	G	T	A	T	G	G	C	A	T	G	A	T	A	G	C	G	C	C	C	G	G	A	A
T	1	-1	-1	-1	-1	-1	1	-1	1	-1	-1	-1	-1	1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	0	0	-2	-2	-2	-2	-1	2	0	0	-2	-2	0	-1	0	0	-1	2	0	-2	-2	-2	-2	-2	-2	-2	0	0
T	1	-1	-1	-3	-3	-3	-1	0	3	1	-1	-3	-2	1	-1	-1	1	0	1	-1	-3	-3	-3	-3	-3	-3	-2	-1
A	-1	0	-2	-2	-4	-4	-3	0	1	2	0	-2	-2	-1	0	0	-1	2	0	0	-2	-4	-4	-4	-4	-4	-2	-1
A	-3	-2	-1	-3	-3	-5	-5	-2	-1	0	1	-1	-1	-3	-2	1	-1	0	1	-1	-1	-3	-5	-5	-5	-5	-3	-1
T	-3	-4	-3	-2	-4	-4	-4	-4	-1	-2	-1	0	-2	0	-2	-1	2	0	-1	0	-2	-2	-4	-6	-6	-6	-5	-3

Και η ακολουθία του πιθανού υποκινητή είναι:

C A T G A T

RNA secondary structure prediction



Nussinov

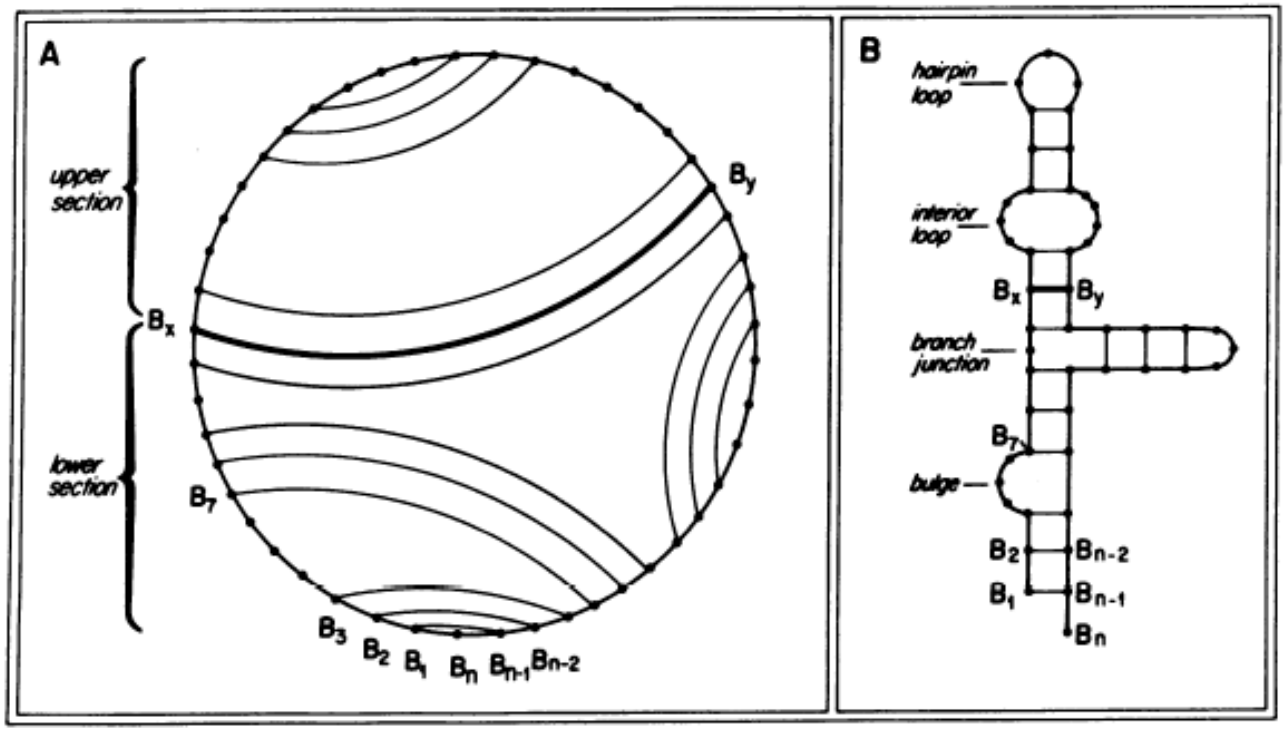


FIG. 1. Base pairing in a simple planar structure. (*A*) Extended form. Base pairs are represented by arcs which join nucleotides located along the circumference of the circle. (*B*) Condensed form. The individual bonds from the structure shown in *A* were shortened and set to a fixed length, resulting in a more conventional representation of a base-paired structure with one branch. Single-stranded loops are labeled according to the nomenclature of Tinoco *et al.* (2).

of the entire molecule is obtained. To illustrate the approach we will consider the sequence $B_1 \ldots B_n$ containing the subsection $B_i \ldots B_j$ of length p. The method used to find the maximal number of base pairs occurring in $B_i \ldots B_j$ is shown in Fig. 2. The variable k is allowed to assume each position from B_i to B_{j-1}. At each point the algorithm tests the ability of B_k to pair with B_j. It further determines the total number of base pairs in the section by checking for the number of base pairs present in the subsections $B_i \ldots B_{k-1}$ and $B_{k+1} \ldots B_{j-1}$. After all positions of k have been tested, the best value obtained is saved and stored in the matrix $M(i,j)$. If B_j cannot pair with any k in $B_i \ldots B_j$, then $M(i,j) = M(i,j-1)$. Information on base pairing for additional sections of length p is obtained by incrementing i and j successively. The maximum number of base pairs that can form in sections of increasing length is obtained by incrementing p to $p+1$ and repeating the search. Thus, for each interval $B_i \ldots B_j$ within the sequence:

$$M(i,j) = \max \begin{cases} M(i,k-1) + M(k+1,j-1) + 1 \\ M(i,j-1) \end{cases} \quad i \le k < j = i + p \qquad [1]$$

Because $M(i,j)$ is filled with sections of increasing length, the values $M(i,k-1)$, $M(k+1,j-1)$, and $M(i,j-1)$ are known and can be read directly from the matrix. This results in an extremely efficient search procedure. The value 1 corresponds to the base pair B_kB_j. The last value in $M(i,j)$ is obtained when the section $B_i \ldots B_j$ corresponds to $B_1 \ldots B_n$. This value of $M(i,j)$ specifies the maximum number of base pairs that can be found for the entire sequence.

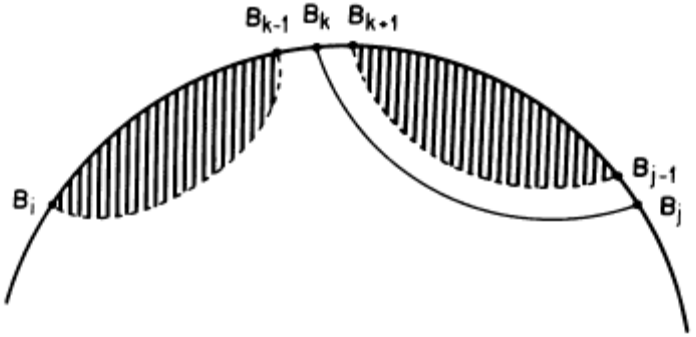


FIG. 2. Various possibilities for B_kB_j bonds as k sweeps across the sequence $B_i \ldots B_{j-1}$ and the separate simple planar structures $(i, k-1)$ and $(k+1, j-1)$, obtained this way.

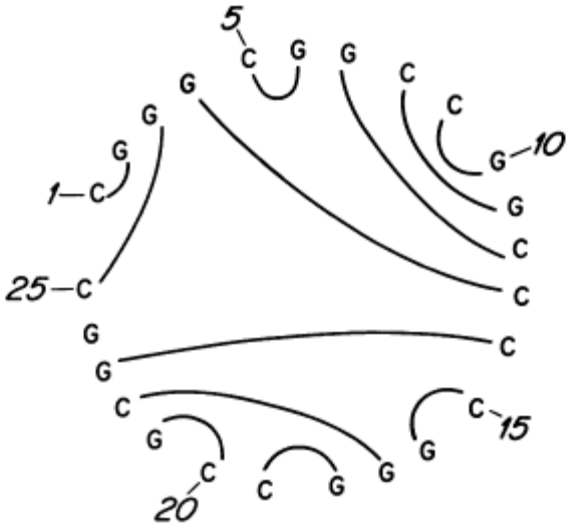
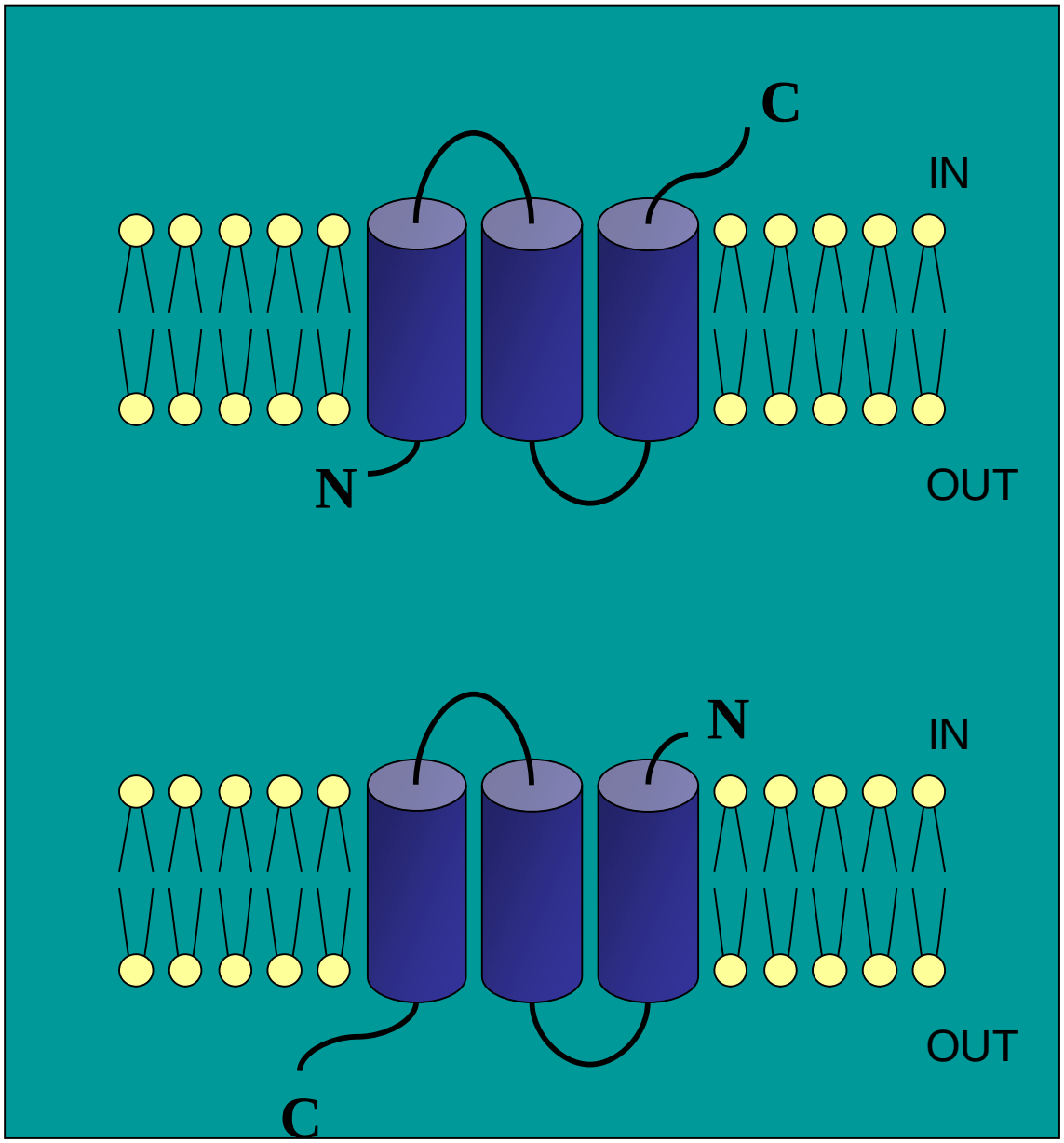


FIG. 4. Secondary structure for the 25-nucleotide chain with maximal pairings as deduced from the algorithm for maximal matching.

Διαμεμβρανικά τμήματα



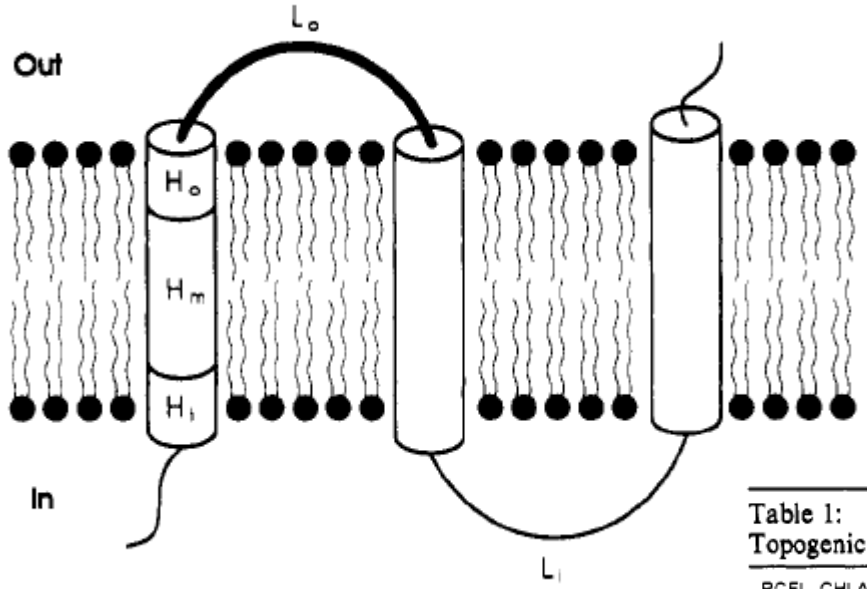


Table 1: Multispanning Protein Sequences Used To Calculate Topogenic Parameters[a]

RCEL_CHLAU	REACTION CENTER PROTEIN L CHAIN. 5/92
RCEM_CHLAU	REACTION CENTER PROTEIN M CHAIN. 5/92
5HT2_CRIGR	5-HYDROXYTRYPTAMINE 2 RECEPTOR (5-HT-2). 5/92
5HT3_MOUSE	5-HYDROXYTRYPTAMINE 3 RECEPTOR PRECURSOR (5-HT-3). 3/92
5HTA_HUMAN	5-HYDROXYTRYPTAMINE 1A RECEPTOR (5-HT-1A). 5/92
A1AA_HUMAN	ALPHA-1A ADRENERGIC RECEPTOR. 5/92
A2AA_HUMAN	ALPHA-2A ADRENERGIC RECEPTOR (SUBTYPE C10). 5/92
EDG1_HUMAN	PROBABLE G PROTEIN-COUPLED RECEPTOR EDG-1. 8/92
MOTA_ECOLI	CHEMOTAXIS MOTA PROTEIN. 11/91
MALF_ECOLI	MALTOSE TRANSPORT INNER MEMBRANE MALF PROTEIN. 11/90
SECY_BACSU	SECY PROTEIN. 3/92
OPS1_CALVI	OPSIN RH1 (OUTER R1-R6 PHOTORECEPTOR CELLS OPSIN). 8/91
OPSB_HUMAN	BLUE-SENSITIVE OPSIN (BLUE CONE PHOTORECEPTOR PIGMENT). 8/91
OPSD_BOVIN	RHODOPSIN. 8/91
AA1R_CANFA	ADENOSINE A1 RECEPTOR. 8/92
AA2R_CANFA	ADENOSINE A2 RECEPTOR. 8/92
C561_BOVIN	CYTOCHROME B561. 7/89
ADT_RICPR	ADP,ATP CARRIER PROTEIN (ADP/ATP TRANSLOCASE). 8/91
CYOB_ECOLI	CYTOCHROME O UBIQUINOL OXIDASE SUBUNIT I (EC 1.10.3.-). 11/90
CYOC_ECOLI	CYTOCHROME O UBIQUINOL OXIDASE SUBUNIT III (EC 1.10.3.-). 11/90
CYOD_ECOLI	CYTOCHROME O UBIQUINOL OXIDASE OPERON PROTEIN CYOD. 11/90
SECE_ECOLI	INNER MEMBRANE PROTEIN SECE. 8/91
GAPB_HUMAN	GAP JUNCTION BETA-1 PROTEIN (CONNEXIN 32). 3/92
LEP_ECOLI	SIGNAL PEPTIDASE I (EC 3.4.-.-) (SPASE I). 8/92
PT2M_ECOLI	PHOSPHOTRANSFERASE ENZYME II, MANNITOL-SPECIFIC. 3/92
ATHP_NEUCR	PLASMA MEMBRANE ATPASE (EC 3.6.1.35). 12/92
LACY_ECOLI	LACTOSE PERMEASE (LACTOSE-PROTON SYMPORT). 12/92
OPPB_SALTY	OLIGOPEPTIDE PERMEASE PROTEIN OPPB. 12/92
TAPA_HUMAN	CELL SURFACE PROTEIN TAPA-1. 8/92
DHSC_BACSU	SUCCINATE DEHYDROGENASE CYTOCHROME B-558. 7/89
LSPA_ECOLI	LIPOPROTEIN SIGNAL PEPTIDASE. 5/91
IMM1_ECOLI	IMMUNITY PROTEIN FOR COLICIN E1. 11/90
IMMA_CITFR	IMMUNITY PROTEIN FOR COLICIN A. 8/91
TCR1_ECOLI	TETRACYCLINE RESISTANCE PROTEIN. 2/91
UHPT_ECOLI	HEXOSE PHOSPHATE TRANSPORT PROTEIN. 8/92

PROPENSITIES

For each of the 5 structural classes, log likelihood ratios for each of the 20 amino acids were calculated:

$$s_i = \ln(q_i/p_i)$$

where p_i is the relative frequency of occurrence (or fraction) of amino acid i in all the sequences in the data set, and q_i is the relative frequency of occurrence of amino acid i in a particular structural class. A positive score indicates a higher than expected frequency for a given amino acid to be found in a particular structural class and a negative score a lower than expected frequency, and a score close to zero indicates that the frequency of occurrence of the given amino acid in a particular class is no different from that expected from chance alone. To circumvent the previously mentioned problem of classifying globular domains as loops, loops longer than 100 residues are *not* classified as loops, and are ignored in the calculation of q_i values. These oversized loops are, however, included in the calculation of the overall relative frequencies of occurrence p_i. The scores calculated for the previously described set of sequences are shown for both single-spanning (Table I in the supplementary material and Figure 2) and multispanning segments (Table II in the supplementary material and Figure 3).

The overall problem of determining the optimal position and length of n transmembranal helices in a sequence of length m is divided into n subproblems: namely, determining the optimal position and length of a single transmembranal helix along with its associated C-terminal coil segment. Let s^{il} be the score associated with a transmembranal helix of length l at position i in the given sequence. This score is calculated according to the diagram shown in Figure 1, where the helix is divided into three sections (two caps of length 4, and a center region of length $l - 8$). Whether the cap and its associated loop are inside or outside depends on the initially specified membrane topology. In order to find the best set of s_j^{il}, we use a recursive algorithm almost identical to the algorithms used for pairwise sequence alignment (Needleman & Wunsch, 1970; Sellers, 1974). A score matrix $\mathbf{S}_j^i(i{:}1...n, j{:}1...m)$ is thus defined (see Figure 4):

$$\mathbf{S}_j^i = \max_{l=17\rightarrow 25}\{s_j^{il} + \max_{k=i+l+A\rightarrow n}\{\mathbf{S}_{j-1}^k\}\}$$

where A is the minimum length of a loop segment.

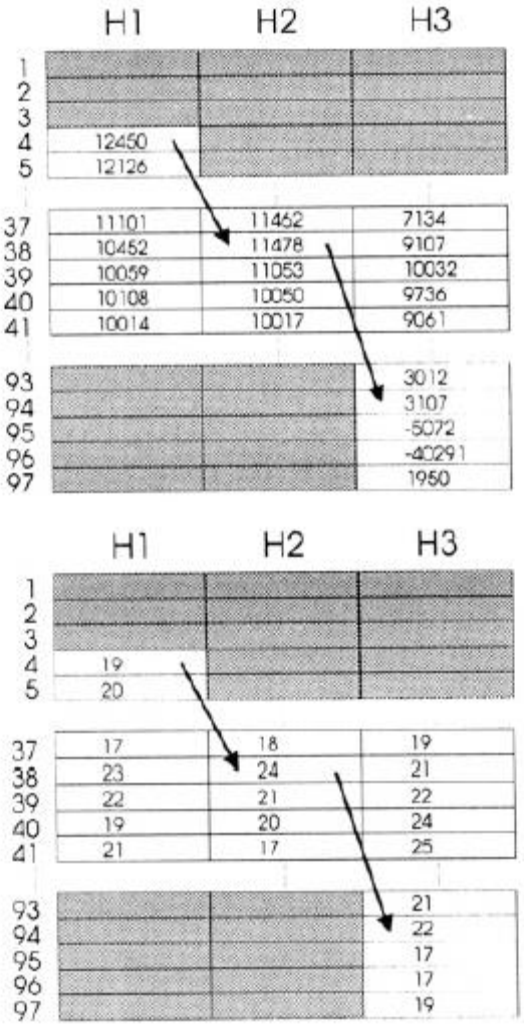


FIGURE 4: A hypothetical score matrix for three transmembrane helices. The upper matrix holds the highest achievable path score for each cell, and the lower matrix stores the helix length which permits this score.

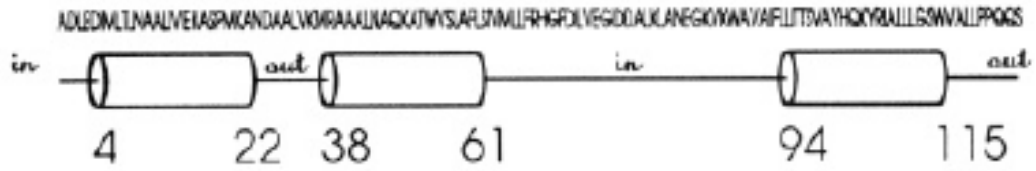


FIGURE 5: Predicted structure and topology relating to the optimal path shown in Figure 4.

Τα 3 βασικά ερωτήματα σε ένα HMM ...

Εκτίμηση

- Δεδομένου του μοντέλου, πως θα υπολογίσουμε την ολική πιθανότητα μιας ακολουθίας συμβόλων. $P(\mathbf{x}|\theta)$

Αποκωδικοποίηση

- Πως θα βρούμε την πιο πιθανή αλληλουχία καταστάσεων (path) από την οποία έχει διέλθει το μοντέλο, για να δώσει την συγκεκριμμένη ακολουθία συμβόλων.

Εκπαίδευση

- Πως θα τροποποιήσουμε τις παραμέτρους του μοντέλου, έτσι ώστε να μεγιστοποιηθεί η συνολική πιθανοφάνεια των ακολουθιών

$$\pi^* = \arg\max_{\pi} P(x,\pi)$$

$$\theta_{ML} = \text{argmax} P(\mathbf{x}|\theta)$$

… και οι απαντήσεις τους

Εκτίμηση

- Αλγόριθμος FORWARD, αλγόριθμος δυναμικού προγραμματισμού, που υπολογίζει την συνολική πιθανότητα της ακολουθίας, χωρίς να διέλθει από όλα τα δυνατά μονοπάτια (αλληλουχίες καταστάσεων).

Αποκωδικοποίηση

- Αλγόριθμος του VITERBI, αλγόριθμος δυναμικού προγραμματισμού, που μέσω αναδρομής (recursion) υπολογίζει την πιο πιθανή αλληλουχία καταστάσεων για τη δεδομένη ακολουθία και το δεδομένο μοντέλο. (Εναλλακτικά NBEST).

Εκπαίδευση

- Αλγόριθμος των BAUM-WELCH (η αλλιώς FORWARD-BACKWARD), ειδική περίπτωση του αλγόριθμου EM (Expectation-Maximization), ο οποίος χειρίζεται τα δεδομένα σαν δεδομένα με ελλειπής τιμές (missing values) και υπολογίζει Ε.Μ.Π. για τις παραμέτρους του μοντέλου (Εναλλακτικά Gradient Descent).

Αλγόριθμος Forward

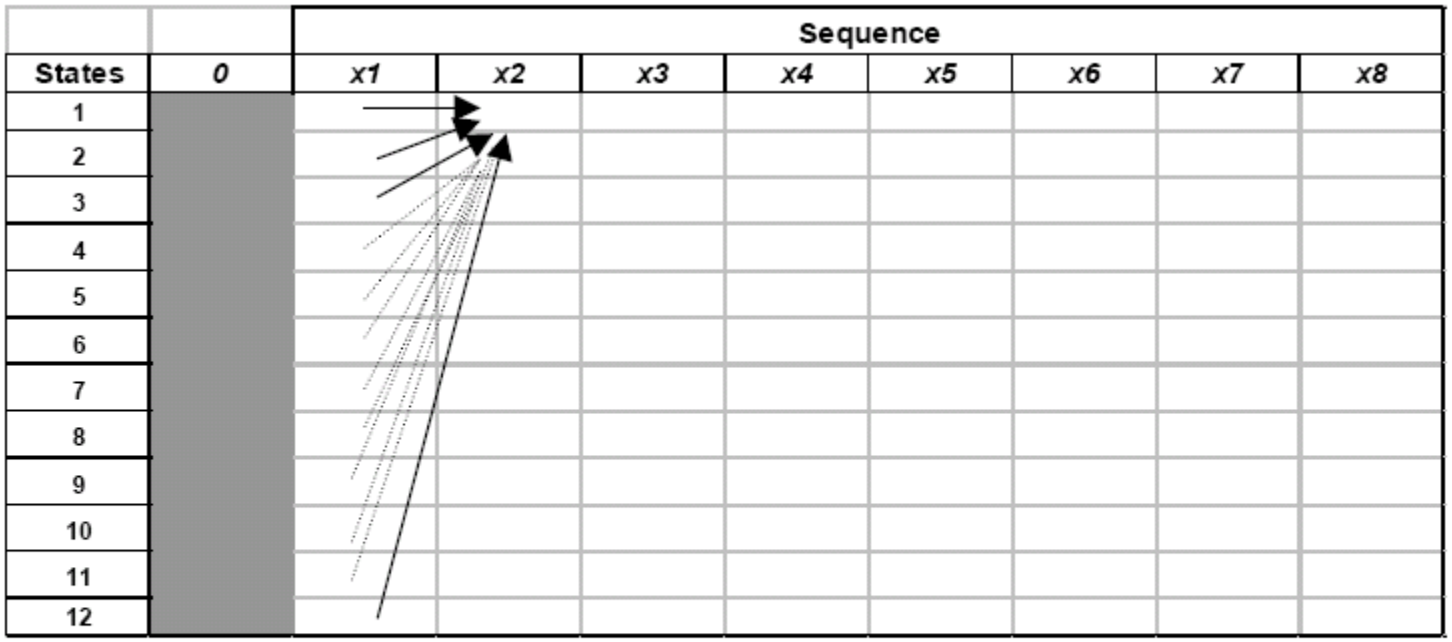
$$\forall k \neq B,\ i = 0 : f_B(0) = 1,\ f_k(0) = 0$$

$$\forall 1 \leq i \leq L : f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl}$$

$$P(\mathbf{x}|\theta) = \sum_k f_k(L) a_{kE}$$

Ο αλγόριθμος αυτός, κατασκευάζει έναν πίνακα με διαστάσεις *N(L+1)*, όπου *N* ο αριθμός των καταστάσεων και *L* το μήκος της ακολουθίας, και θεωρεί μια ενδιάμεση μεταβλητή $f_k(i)$ για κάθε θέση *i* και κατάσταση *k* της ακολουθίας. Η ποσότητα αυτή, πρακτικά, είναι ίση με την από κοινού πιθανότητα της αλληλουχίας έως το κατάλοιπο *i*, και του μονοπατιού που αντιστοιχεί στην κατάσταση *k*. Δηλαδή:

$$f_k(i) = P(x_1, x_2, ..., x_i, \pi_i = k)$$



Εικόνα 2.6 Διαγραμματική απεικόνιση του πίνακα Forward, για ένα υποθετικό μοντέλο με 12 καταστάσεις (states), και μια ακολουθία από 8 κατάλοιπα. Για τον υπολογισμό της τιμής ενός κελιού (π.χ. του $f_1(2)$), υπολογίζονται οι συνεισφορές όλων των προηγούμενων κελιών στη θέση 1 της ακολουθίας (βέλη).

Εντελώς ανάλογος είναι ο αλγόριθμος Backward (Durbin et al., 1998; Rabiner, 1989), ο οποίος διαφέρει μόνο ως προς την κατεύθυνση προς την οποία διατρέχει την αλληλουχία. Η ενδιάμεση μεταβλητή που χρησιμοποιείται, ονομάζεται πλέον $b_k(i)$, και ορίζεται για κάθε i ως η πιθανότητα της ακολουθίας από την θέση i+1 έως το τέλος, δεδομένου ότι στη την θέση i συναντάμε την κατάσταση k. Δηλαδή:

$$b_k(i) = P(x_{i+1},...,x_L \mid \pi_i = k)$$

Άρα ο αλγόριθμος, διατυπώνεται ως εξής:

Αλγόριθμος Backward

$$\forall k,\ i = L:\ b_k(L) = a_{kE}$$

$$\forall\ 1 \le i < L:\ b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$

$$P(\mathbf{x}|\theta) = \sum_l a_{Bl} e_l(x_1) b_l(1)$$

Όμοια, αν δεν υπάρχουν καταστάσεις λήξεως, στην αρχικοποίηση, οι αντίστοιχες πιθανότητες τίθενται ίσες με 1. Το τελικό αποτέλεσμα του αλγορίθμου, είναι ακριβώς όμοιο με αυτό του Forward.

Αλγόριθμος Viterbi

Αλγόριθμος Viterbi

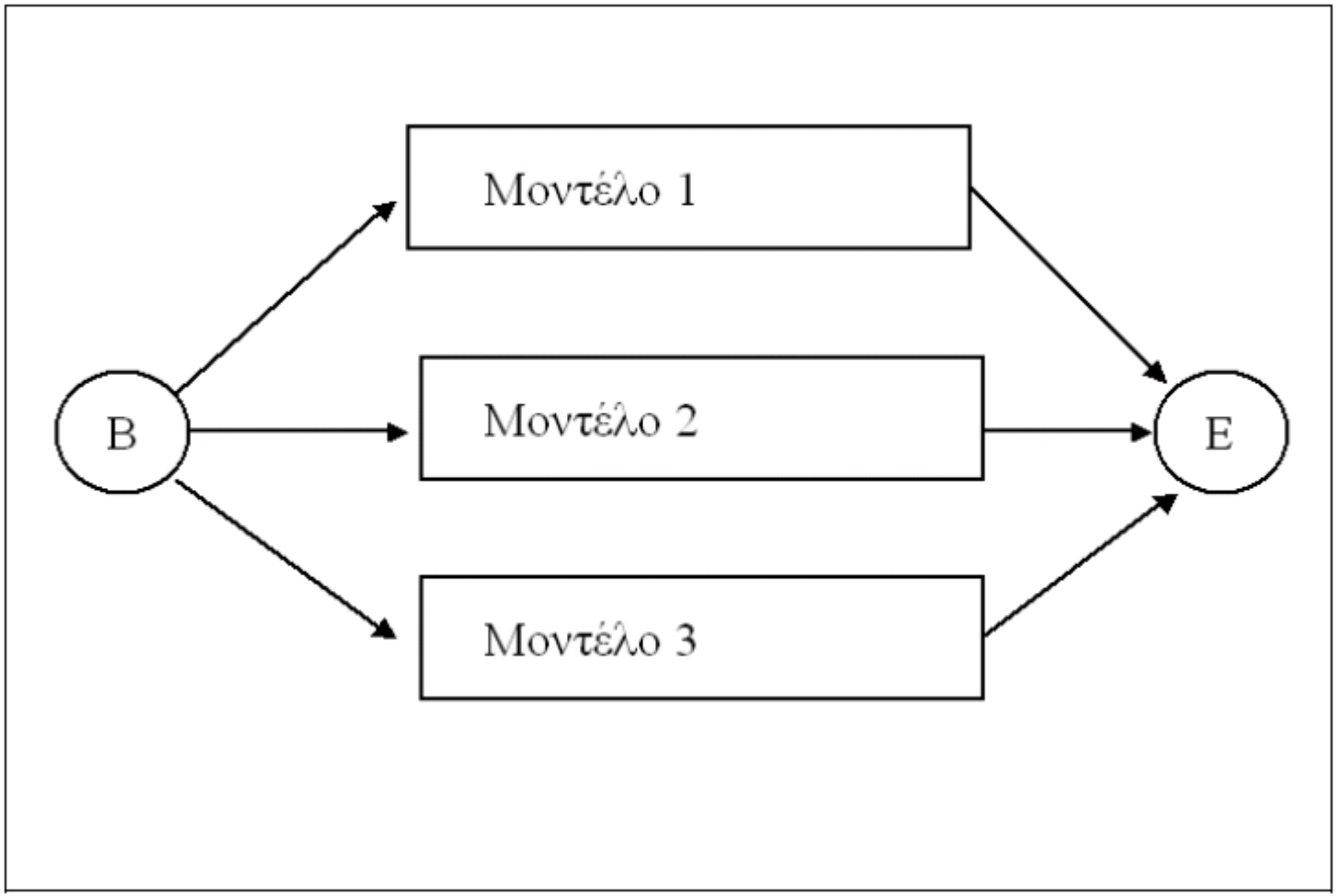
$$\forall k \neq B,\ i = 0:\ u_B(0) = 1,\ u_k(0) = 0$$

$$\forall\ 1 \leq i \leq L:\ u_l(i) = e_l(x_i) \max_k \{u_k(i-1)a_{kl}\}$$

$$P(\mathbf{x}, \pi^{\max} \mid \theta) = \max_k \{u_k(L)a_{kE}\}$$

Ο αλγόριθμος του Viterbi, είναι στην ουσία όμοιος με τον Forward, με τη μόνη διαφορά να βρίσκεται στο ότι τα διαδοχικά αθροίσματα αντικαθίστανται από μεγιστοποιήσεις. Σε αυτή την περίπτωση με π^{max}, συμβολίζουμε το μονοπάτι με τη μεγαλύτερη πιθανότητα και η πιθανότητα αυτή συμβολίζεται με $P(\mathbf{x}, \pi^{max}|\theta)$. Προφανώς, ισχύει ότι $P\left(\mathbf{x}, \pi^{\max} \mid \theta\right) \leq P\left(\mathbf{x} \mid \theta\right)$. Ένα επιπλέον χαρακτηριστικό του αλγόριθμου αυτού, είναι το ότι απαιτεί την ύπαρξη ενός ξεχωριστού πίνακα στον οποίο θα κρατούνται δείκτες (pointers), για την καλύτερη (πιθανότερη) κατάσταση σε κάθε θέση της ακολουθίας. Με αναδρομή (back-tracking), σε αυτόν τον πίνακα, ανακτά κανείς στο τέλος, το ίδιο το πιθανότερο μονοπάτι.

Αποκωδικοποίηση forward



“Εκ των υστέρων” αποκωδικοποίηση

Εναλλακτικά μπορεί να υπολογισθεί η πιθανότητα: $P(\pi_i = k \mid \mathbf{x})$
δηλαδή, η εκ των υστέρων πιθανότητα το συγκεκριμμένο νουκλεοτίδιο να προήλθε απο μια κατάσταση

$$P(\mathbf{x}, \pi_i = k) = P(x_1, x_2, ..., x_i, \pi_i = k) P(x_{i+1}, ..., x_n \mid x_1, ..., x_i, \pi_i = k)$$

$$= P(x_1, x_2, ..., x_i, \pi_i = k) P(x_{i+1}, ..., x_n \mid \pi_i = k)$$

Κάνοντας χρήση των Forward και Backward:

$$P(\mathbf{x}, \pi_i = k) = f_k(i) b_k(i)$$

Τέλος, σύμφωνα με το θεώρημα Bayes θα έχουμε:

$$P(\pi_i = k \mid \mathbf{x}) = \frac{f_k(i) b_k(i)}{P(\mathbf{x})}$$

Με τον τύπο αυτό, μπορούμε να υπολογίσουμε την πιθανότητα μια παρατήρηση να προέρχεται από μια συγκεκριμένη κατάσταση. Μπορούμε επίσης να ορίσουμε μια άλλη αλληλουχία καταστάσεων, για την οποία ισχύει:

$$\hat{\pi}_i = \arg\max_k P(\pi_i = k \mid \mathbf{x})$$

Πλεονεκτήματα:

• στις περιπτώσεις που τα εναλλακτικά μονοπάτια έχουν πολύ μικρές διαφορές στις προβλεπόμενες πιθανότητες.

• όταν μια κατάσταση έχει πολύ μικρή πιθανότητα και το μονοπάτι με την μέγιστη πιθανότητα, δεν την «επισκέπτεται» ποτέ.

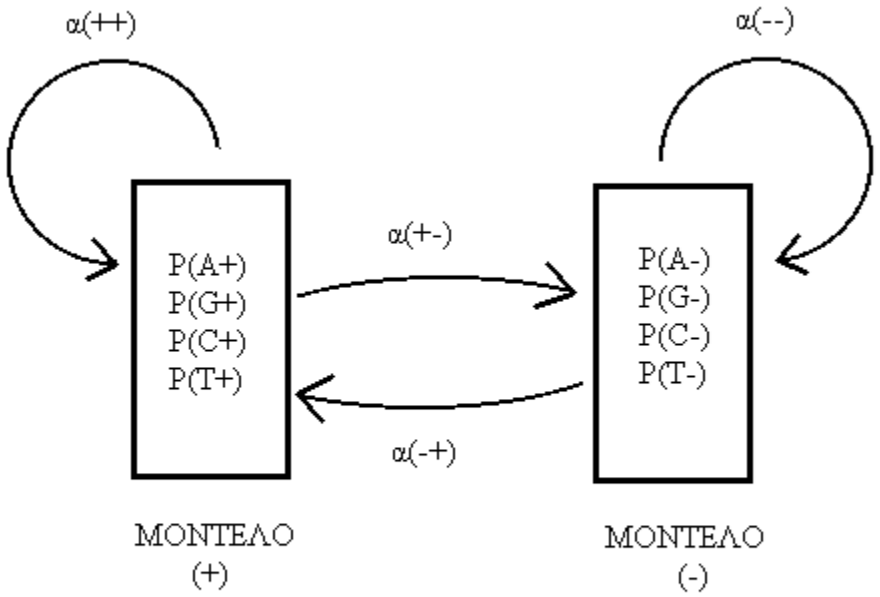
Μειονεκτήματα:

• Μπορεί να προβλεφθεί μια πιθανότητα η οποία δεν είναι έγκυρη για το μοντέλο (μια μη επιτρεπτή μετάβαση).

Συνοπτικά ο αλγόριθμος

- Υπολογισμός των Α και Ε
- Υπολογισμός των ΕΜΠ
- Επανάληψη μέχρι να συγκλίνει

Ένα παράδειγμα...



συνέχεια…

Πιθανότητες μεταβάσεως:

	1	0
1	0.90	0.10
0	0.10	0.90

Πιθανότητες γεννήσεως :

	A	T	G	C
1	0.70	0.10	0.10	0.10
0	0.25	0.25	0.25	0.25

συνέχεια...

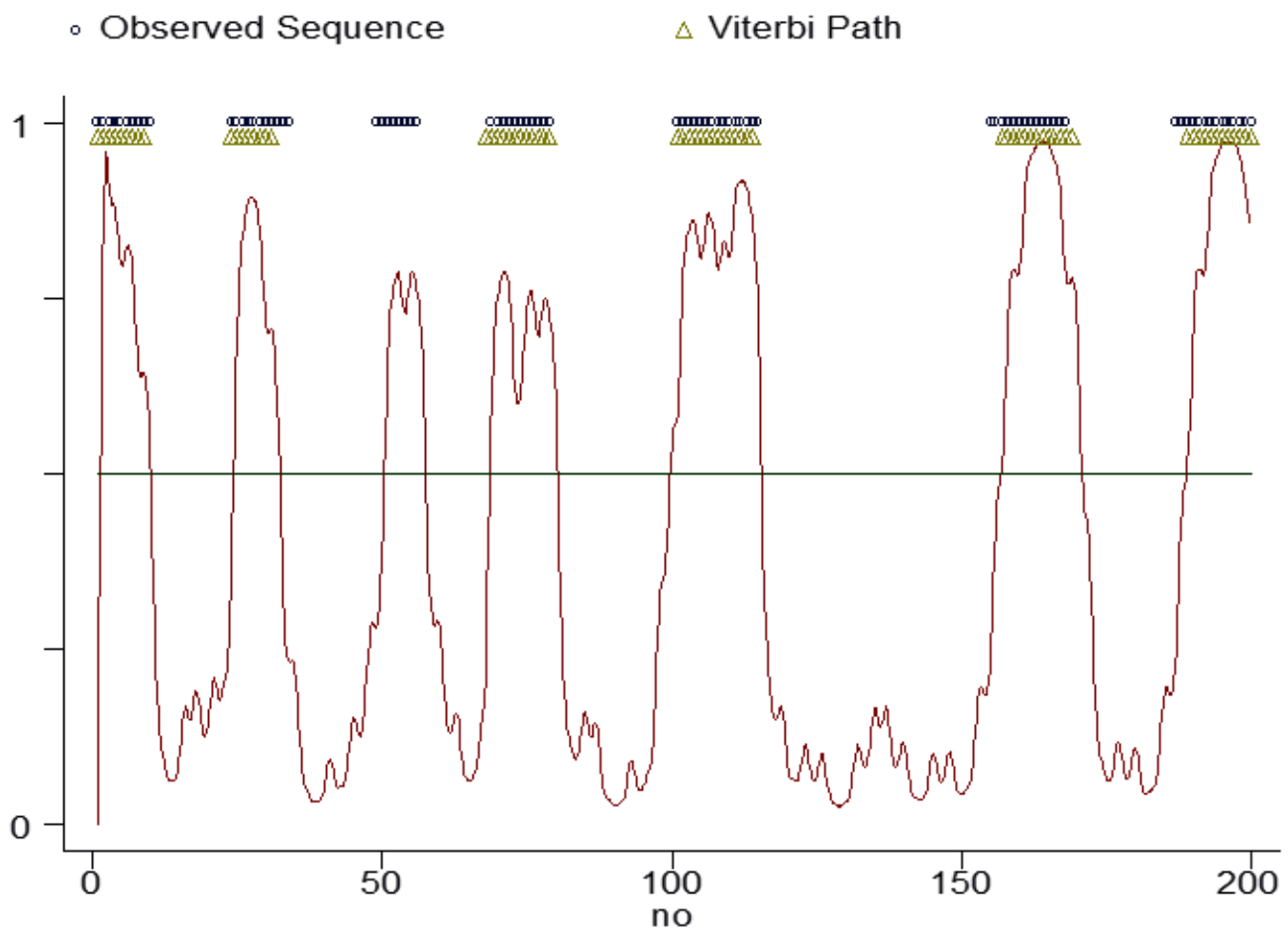
Έστω μια ακολουθία DNA, η οποία προέρχεται από το παραπάνω μοντέλο:

```
AAACAAGAATGCGCACACTACGCAAAAACAATTAGTCGCACTCACGATGAAACAAATTACCACGGTGAA
1111111111000000000000011111111110000000000000011111110000000000001

AACGAATAAACCTCAGAGGCCCAGCGTATATAAACAAGATAAAAACCTAGTCAGCACTCTGACCAGACG
1111111111000000000000000000000011111111111111100000000000000000000000

AGCTCACGACTTGAGGATAAGAAAAAAACAACAGCTCACGACTTGAGGATAAGAAAAAAACA
0000000000000000111111111111110000000000000000011111111111111
```

συνέχεια...



συνέχεια…

Αν όμως οι πιθανότητες μεταβάσεως άλλαζαν:

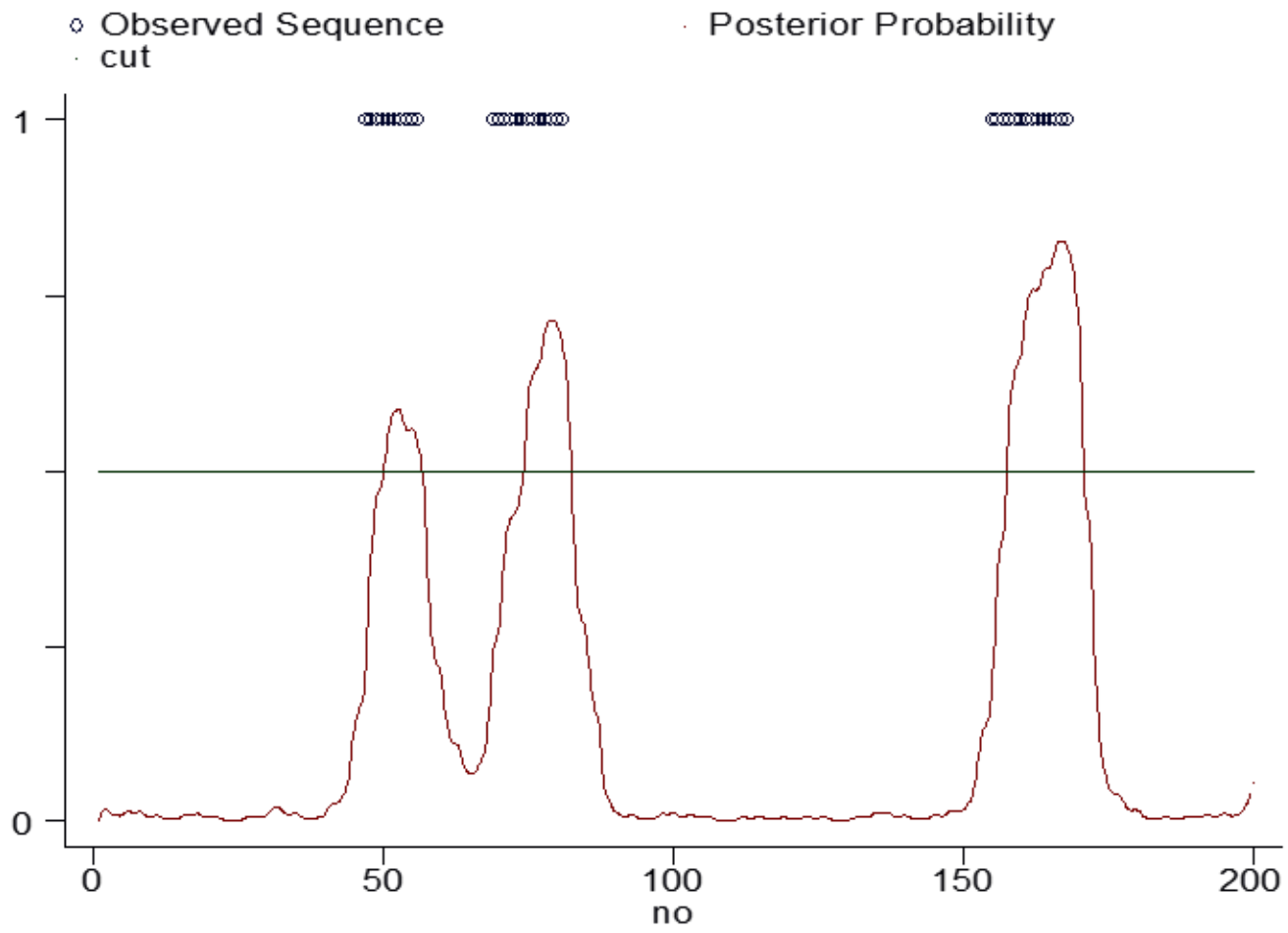
Πιθανότητες μεταβάσεως:

	1	0
1	0.98	0.02
0	0.03	0.97

Πιθανότητες γεννήσεως :

	A	T	G	C
1	0.60	0.10	0.10	0.10
0	0.25	0.25	0.25	0.25

συνέχεια...


Observed Sequence
cut
Posterior Probability
1
0
0
50
100
150
200
no

Posterior-Viterbi decoding

Ορίζονται οι επιτρεπτές μεταβάσεις:

$$\delta(k,l)=\begin{cases}1, \text{ if } a_{kl}>0\\ 0, \text{ otherwise}\end{cases}$$

Τελικά, το βέλτιστο επιτρεπτό εκ των υστέρων μονοπάτι π^{PV}, δίνεται από τη σχέση:

$$\pi^{PV}=\arg\max_{\pi}\prod_{i=1}^{L}\delta(\pi_i,\pi_{i+1})P(\pi_i\,|\,\mathbf{x})$$

Ο συνολικός αλγόριθμος, ο οποίος παρουσιάζεται παρακάτω, είναι στην ουσία μια παραλλαγή του αλγορίθμου Viterbi, στην οποία οι πιθανότητες γεννήσεως αντικαθίστανται από τις εκ των υστέρων πιθανότητες και οι πιθανότητες μετάβασης από την δίτιμη συνάρτηση που είδαμε παραπάνω.

Αλγόριθμος Posterior-Viterbi

$$\forall k\neq B,\ i=0:\ u_B(0)=1,\ u_k(0)=0$$

$$\forall\ 1\le i\le L:\ u_l(i)=P(\pi_i=l\,|\,\mathrm{x})\max_k\{u_k(i-1)\delta(k,l)\}$$

$$P(\mathbf{x},\pi^{PV}\,|\,\theta)=\max_k\{u_k(L)\delta(k,E)\}$$

Optimal Accuracy Posterior Decoding

Παραλλαγή του Posterior-Viterbi, η οποία υπολογίζει το μονοπάτι:

$$\pi^{OAPD} = \arg\max_{\pi} \sum_{i=1}^{L} \left\{ \delta(\pi_i, \pi_{i+1}) \left(\sum_k P(\pi_i \mid \mathbf{x}) \lambda_k(c) \right) \right\}$$

Συνολικά:

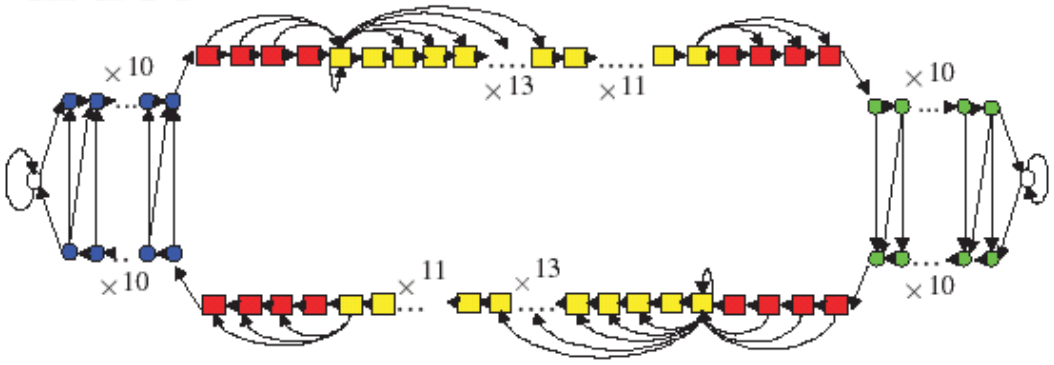
Optimal Accuracy Posterior Decoder algorithm

$$\forall k \neq B,\ i = 0:\ A_B(0) = 0,\ A_k(0) = -\infty$$

$$\forall\ 1 \leq i \leq L:\ A_l(i) = P\left(y_i = c^l \mid \mathbf{x}, \theta\right) + \max_k \left\{ A_k(i-1)\delta(k,l) \right\}$$

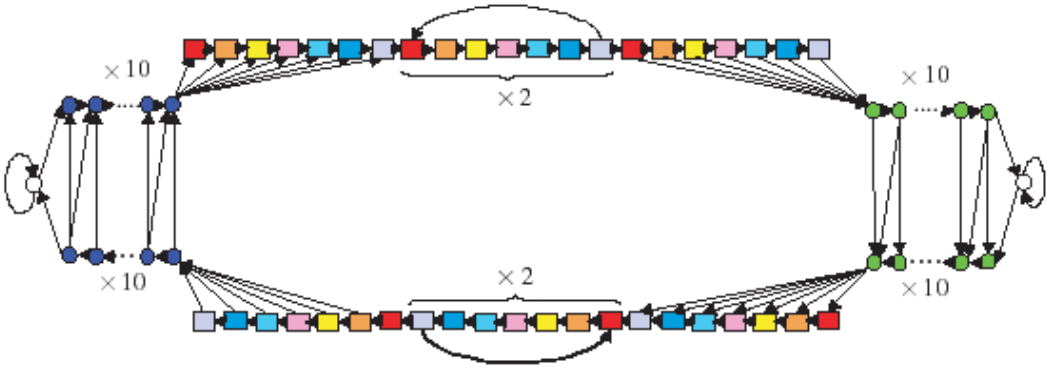
$$P(\mathbf{x}, \pi^{OAPD} \mid \theta) = \max_k \left\{ A_k(L)\delta(k,E) \right\}$$

HMM 1

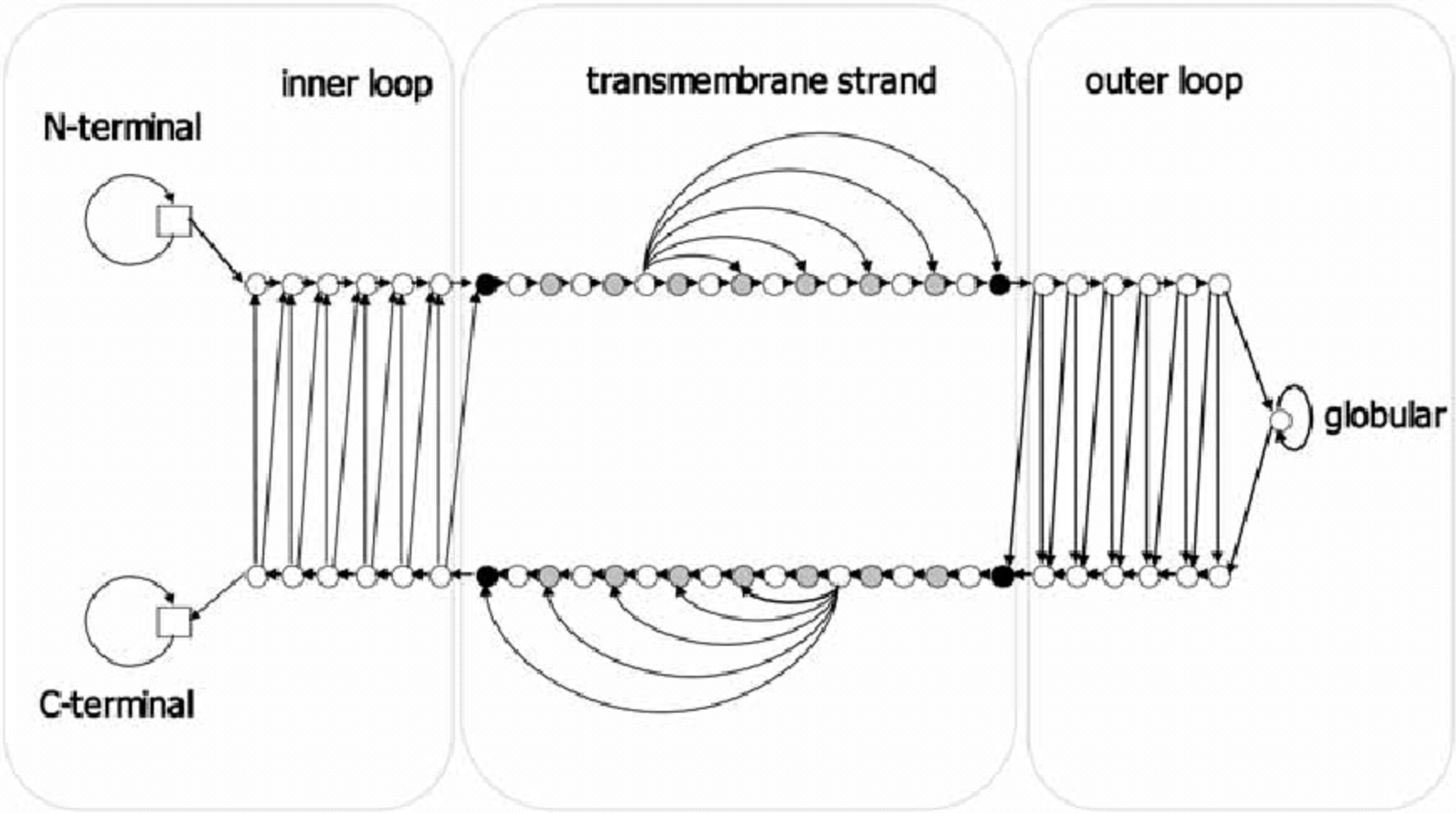


Outer Side Transmembrane Inner Side

HMM 2



Outer Side Transmembrane Inner Side


inner loop
transmembrane strand
outer loop
N-terminal
globular
C-terminal

Άλλες εφαρμογές

- Fold recognition
- Threading
- Domain recognition

Fold recognition

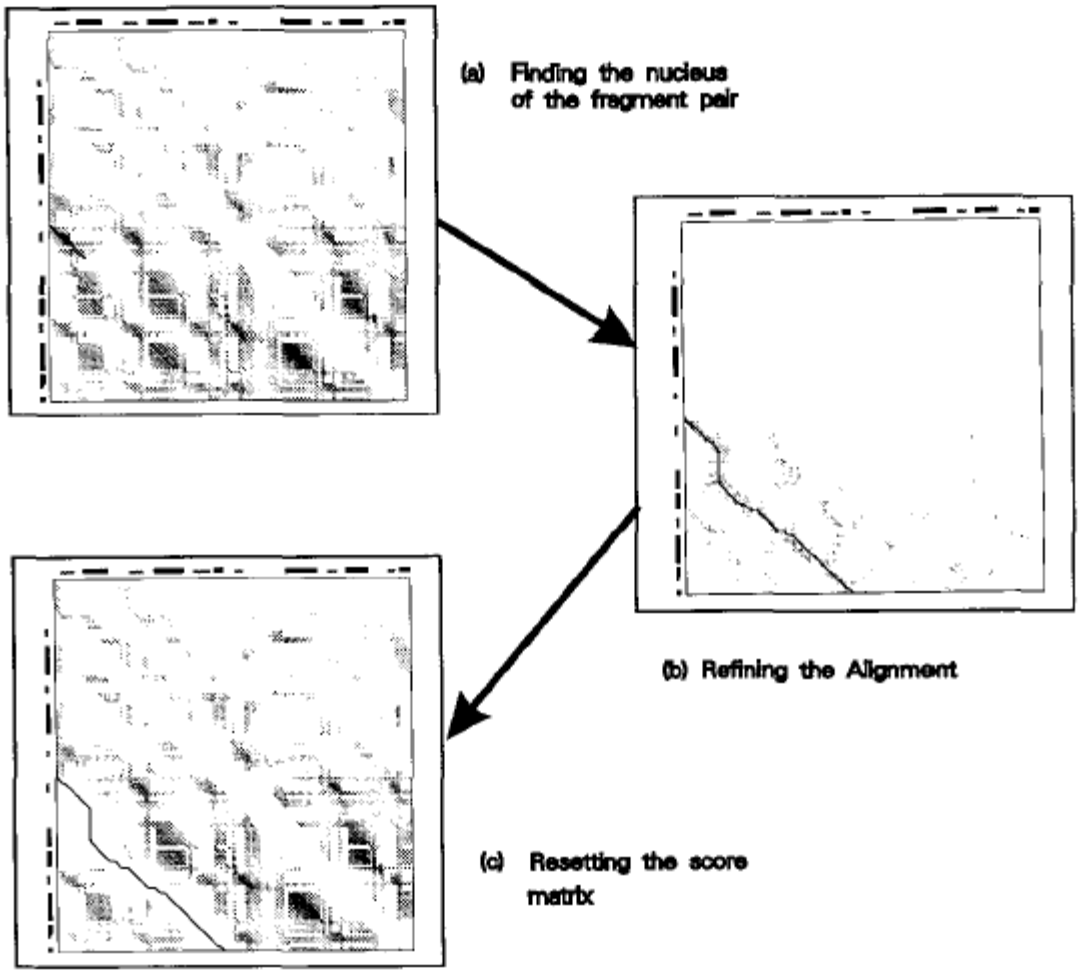


Figure 1. Flow chart of the local structural alignment method. The axes show secondary structure regions, where filled rectangles represent alpha helical regions and narrower rectangles denote beta strands. After comparing *all* residue pairs in similar structural environments and summing scores in the upper similarity matrix (U), the pathway associated with the highest score in the matrix is identified. This path is traced back until the score falls below the threshold, revealing the most conserved region (nucleus) of a fragment pair (a). In the next step, at most 20 of the aligned residues in the fragment nucleus are re-compared and scores re-summed in the temporary matrix (U') allowing a refined alignment path to be determined through the complete fragment (b). Before searching the upper matrix for the next path, all scores along and in secondary structures aligned by the previous path are reset to zero (c).

Threading

- Protein threading is the problem of aligning a protein sequence whose structure we want to elucidate (the target protein) with a protein sequence whose structure is known (the template protein) in such a way that mapping residues of the target onto a template according to the alignment affords an accurate model of the backbone structure of the target.

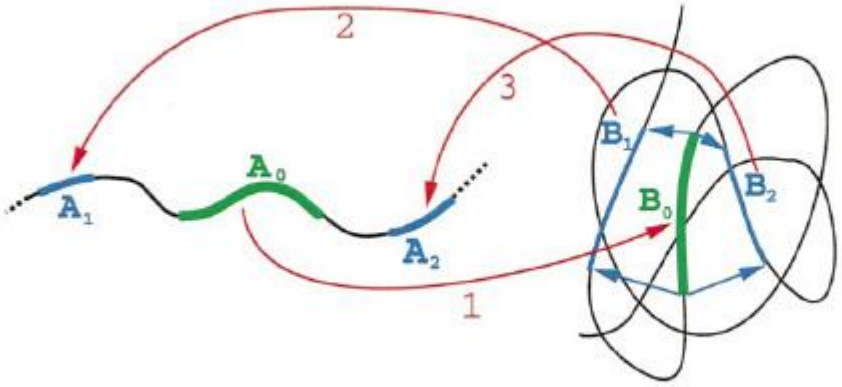


Figure 13. Sketch of the RDP procedure: recursively mapping the target sequence onto the template structure.

Domain recognition

CRX_HUMAN
cone-rod homeobox protein

homeobox 40-96
Context: TF_Otx **164-250**

SR68_HUMAN
signal recognition particle 68 kda protein (srp68)

Context: TPR **180-213**
Context: TPR **455-488**

E2BG_CAEEL
putative translation initiation factor eif-2b gamma subunit

Context: NTP_transferase **5-280**
Context: hexapep **305-322**
Context: hexapep **339-356**
Context: hexapep **362-379**

Dynamic Programming Algorithm. The space of all potential domain assignments for a particular protein is large. Hence we need to design algorithms that concentrate on searching through probable domain assignments. Our approach is to first run HMMER against the protein for each Pfam family. We keep those hits that have a HMMER e value $<1{,}000$. In this way, we obtain a list $\mathbf{d} = d_1 \ldots d_m$ of potential domains, ordered by end position, with corresponding amino acid sequences $a_1 \ldots a_m$. Our search space is now all possible combinations of domains in this list. We optimize the search through this reduced space by using a dynamic programming technique.

We want to find the domain sentence $D = D_1 \ldots D_n$, a sublist of $\mathbf{d}$ with corresponding amino acid sequences $A_1 \ldots A_n$, which maximizes the protein log-odds score $S(D)$, where

$$S(D) = \sum_{i=1}^{i=n+1} H(D_i) + C(D_i|D_{i-1}) \qquad [8]$$

$$H(D_i) = \log_2\left(\frac{P(A_i|D_i)}{P(A_i|R)}\right) - T_{D_i} \qquad [9]$$

$$C(D_i|D_{i-1}) = \log_2\left(\frac{P(D_i|D_{i-1})}{P(D_i)}\right). \qquad [10]$$

Note that $H(D_i)$ is just the HMMER score for the domain, and that $C(D_i|D_{i-1})$ is the transition score. We denote the begin and end states as D_0, D_{n+1}, respectively, so that $C(D_1|D_0)$ is the transition from begin state and $C(D_{n+1}|D_n)$ is the transition to end state. We set $H(D_{n+1}) = 0$ as the end state contributes no sequence-based score. We use the curated Pfam "gathering" threshold for T_{D_i}.

We define D^i to be the highest scoring domain sentence that ends in domain d_i without overlaps. The following recursion relation then applies:

$$S(D^i) = H(d_i) + \max_{j<i:a_j\cap a_i=\phi} (S(D^j) + C(d_i|d_j)), \qquad [11]$$

where the condition $a_j \cap a_i = \phi$ ensures that the maximizing sentence does not contain domain overlaps, which requires tracking the protein coordinates of domains. We then set

$$D^i = \{D^j, d_i\}, \qquad [12]$$

where D^j maximizes Eq. **11**. Repeated application of Eqs. **11** and **12** for $i = 1 \ldots m + 1$ gives the maximizing sentence $D = D^{m+1}$ required by Eq. **8** (again, we use the convention that d_{m+1} is the end state, so that D^{m+1} is interpreted as the maximizing sentence ending with the end state).