ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

ΟΜΑΔΑ ΑΝΑΠΤΥΞΗΣ:

- ΝΙΚΟΛΑΣ ΠΑΤΕΡΑΣ – **Π17172**
- ΑΝΤΡΕΑΣ ΘΕΟΔΩΡΙΔΗΣ – **Π17164**
- ΒΑΣΙΛΕΙΟΣ ΖΑΡΤΗΛΑΣ ΠΑΠΑΧΑΡΑΛΑΜΠΟΥΣ – **Π17168**

Ιούλιος 2020

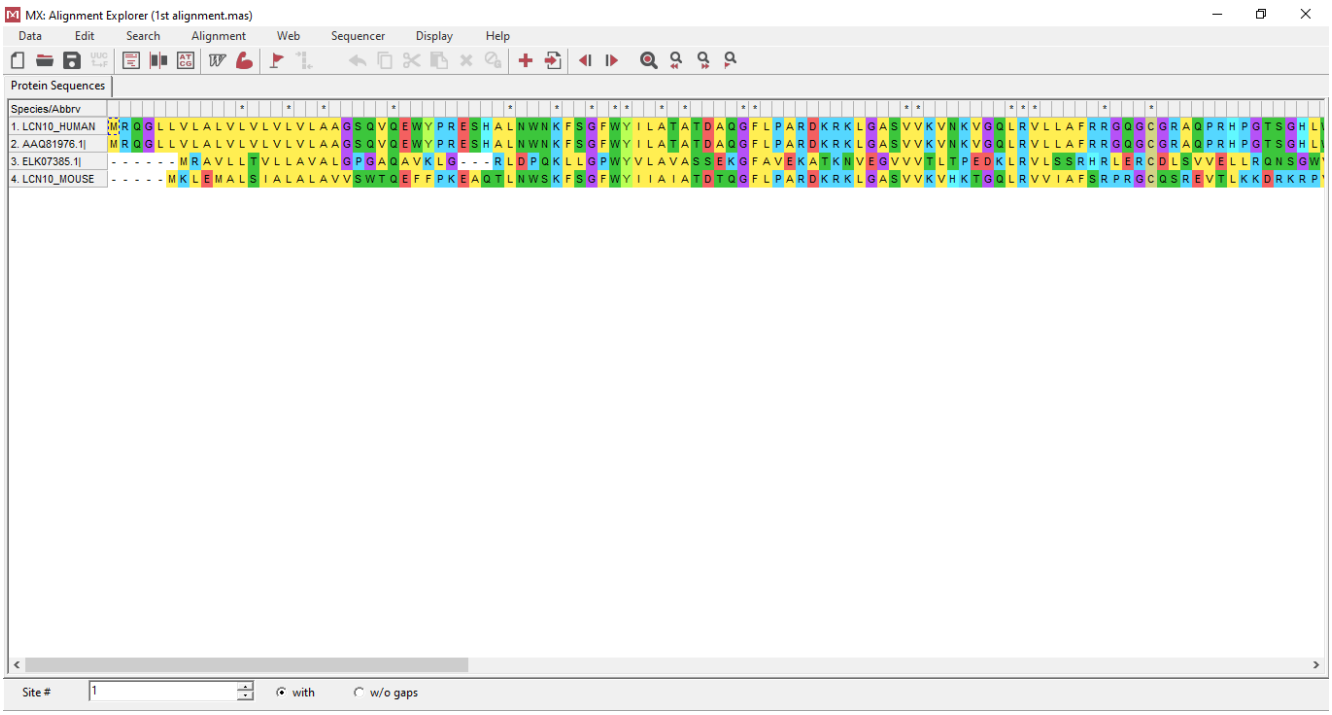
ΠΕΡΙΕΧΟΜΕΝΑ

ΘΕΜΑ 1

ΆΣΚΗΣΗ 7.2

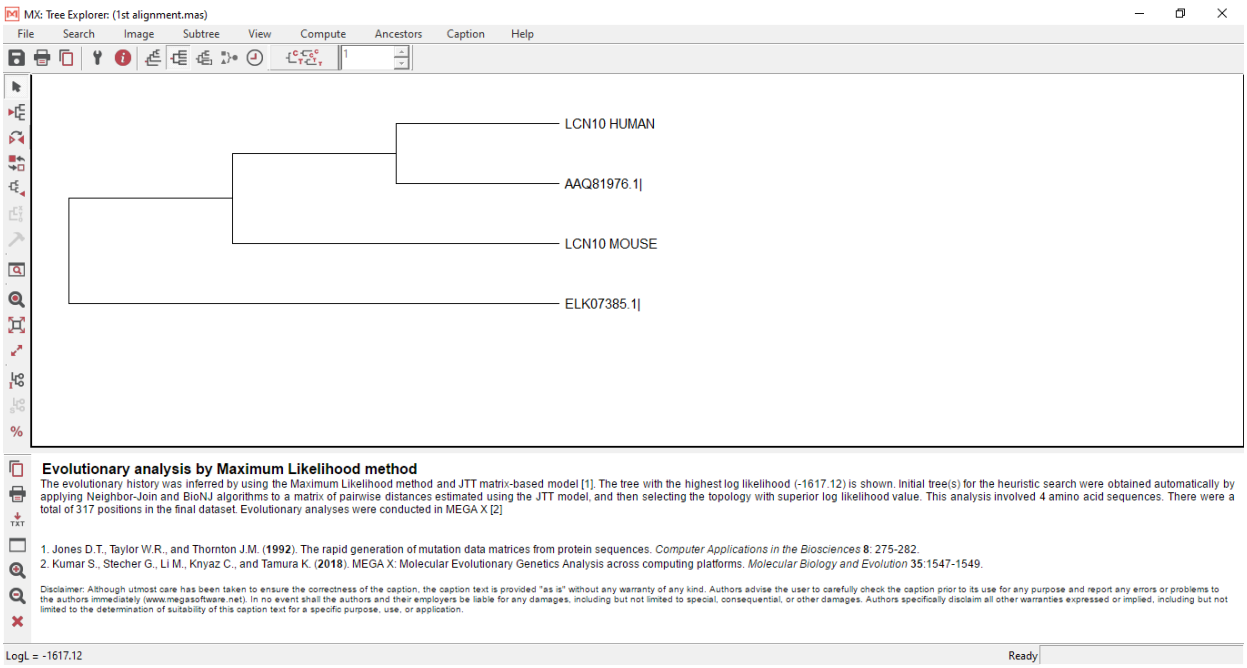
Στοίχιση Αλληλουχιών

Στην πιο κάτω εικόνα βλέπουμε τις στοιχισμένες αλληλουχίες της οικογένειας λιποκαλίνες:



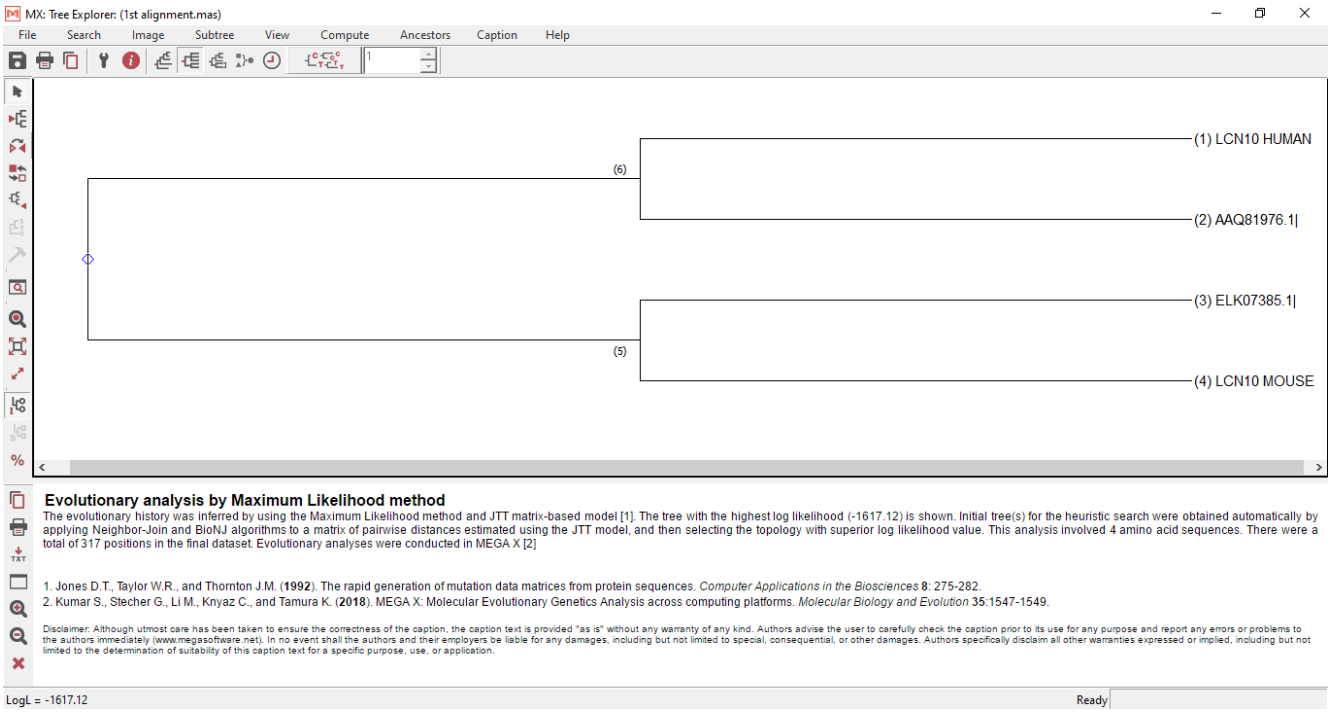
Likelihood Tree

Το δέντρο μέγιστης πιθανοφάνειας εμφανίζεται στην πιο κάτω εικόνα:

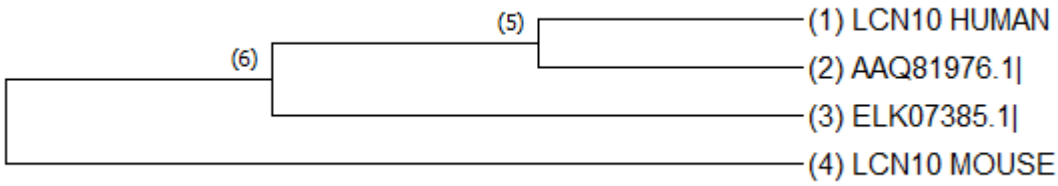


Η εξελικτική ιστορία συνάγεται με τη χρήση της μεθόδου Maximum Likelihood και του μοντέλου JTT που βασίζεται σε μήτρα. Εμφανίζεται το δέντρο με την υψηλότερη πιθανότητα καταγραφής (-1617.12). Τα αρχικά δέντρα για την ευρετική αναζήτηση ελήφθησαν αυτόματα με την εφαρμογή αλγορίθμων Neighbor-Join και BioNJ σε μια μήτρα ζευγών αποστάσεων που υπολογίστηκαν χρησιμοποιώντας το μοντέλο JTT και στη συνέχεια επιλέγοντας την τοπολογία με ανώτερη τιμή πιθανότητας log. Αυτή η ανάλυση περιελάβανε 4 αλληλουχίες αμινοξέων. Υπήρχαν συνολικά 317 θέσεις στο τελικό σύνολο δεδομένων.

Αλλαγή στην ρίζα του δέντρου:



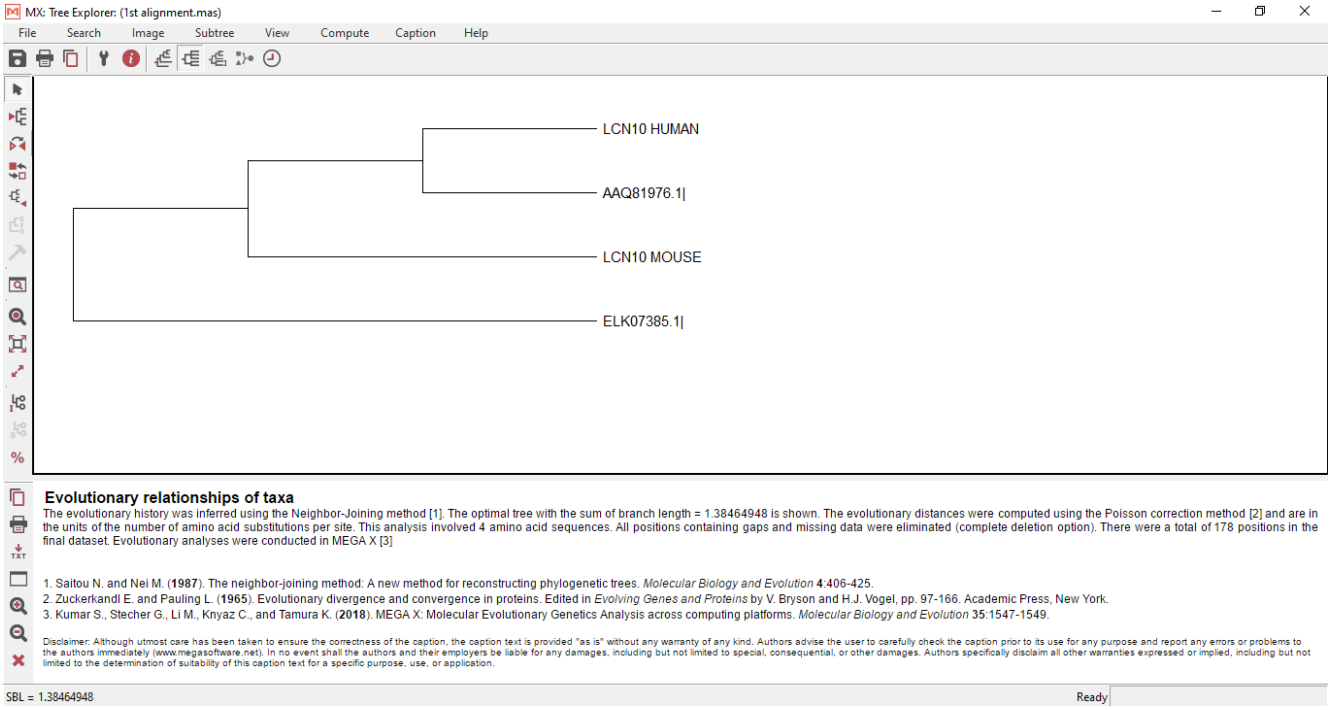
Το δέντρο μέγιστης πιθανοφάνειας με bootstrap(500):



Το δέντρο συναίνεσης bootstrap που συνάγεται από 500 επαναλήψεις θεωρείται ότι αντιπροσωπεύει την εξελικτική ιστορία των ταξινομήσεων που αναλύθηκαν. Οι κλάδοι που αντιστοιχούν σε διαμερίσματα που αναπαράγονται σε λιγότερο από 50% επαναλήψεις bootstrap συμπτύσσονται. Τα αρχικά δέντρα για την ευρετική αναζήτηση ελήφθησαν αυτόματα με την εφαρμογή αλγορίθμων Neighbor-Join και BioNJ σε μια μήτρα ζευγών αποστάσεων που υπολογίστηκαν χρησιμοποιώντας το μοντέλο JTT και στη συνέχεια επιλέγοντας την τοπολογία με ανώτερη τιμή πιθανότητας log.

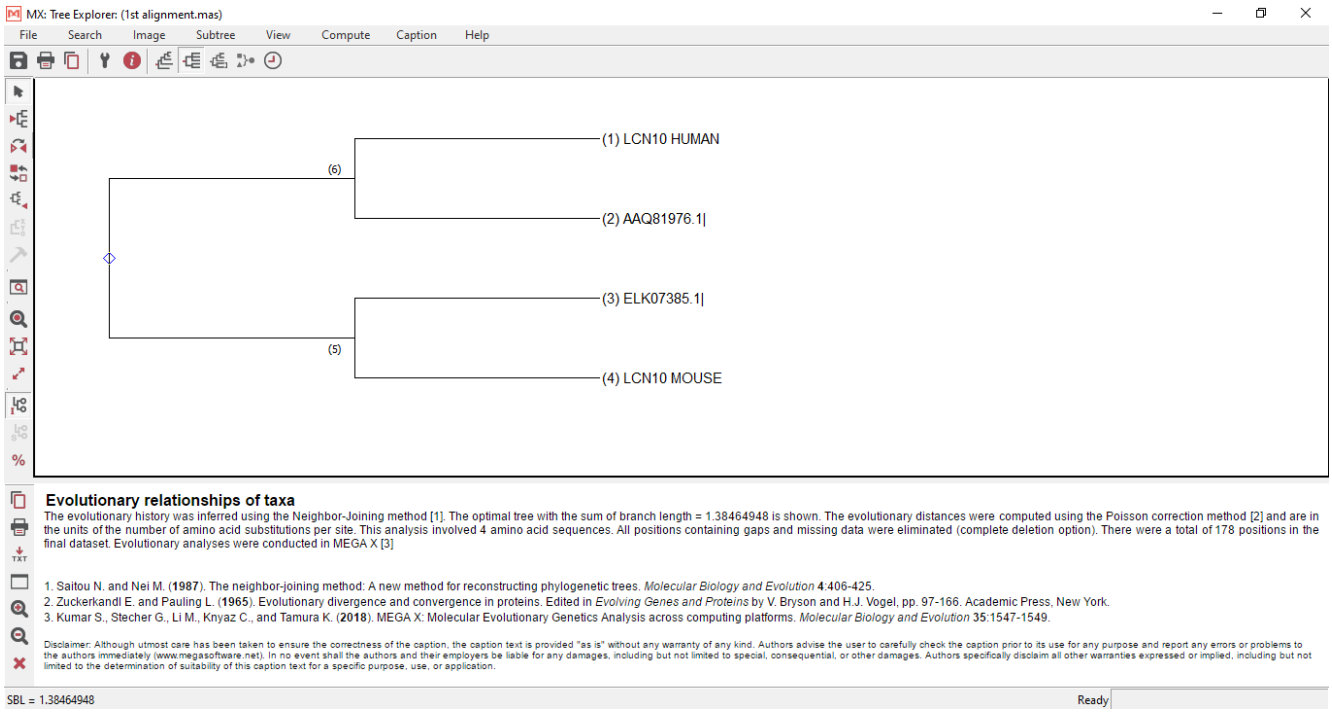
Neighbor-Join Tree

Το δέντρο Neighbor-Joining εμφανίζεται στην πιο κάτω εικόνα:

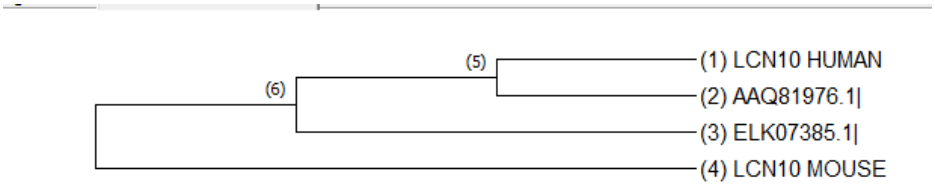


Η εξελικτική ιστορία συνήχθη με τη μέθοδο Neighbor-Joining. Εμφανίζεται το βέλτιστο δέντρο με το άθροισμα του μήκους διακλάδωσης =1.38464948. Οι εξελικτικές αποστάσεις υπολογίστηκαν χρησιμοποιώντας τη μέθοδο διόρθωσης Poisson και είναι στις μονάδες του αριθμού υποκαταστάσεων αμινοξέων ανά τοποθεσία. Αυτή η ανάλυση περιλάμβανε 4 αλληλουχίες αμινοξέων. Όλες οι θέσεις που περιέχουν κενά και ελλείποντα δεδομένα εξαλείφθηκαν (πλήρης επιλογή διαγραφής). Υπήρχαν συνολικά 178 θέσεις στο τελικό σύνολο δεδομένων.

Αλλαγή στην ρίζα του δέντρου:



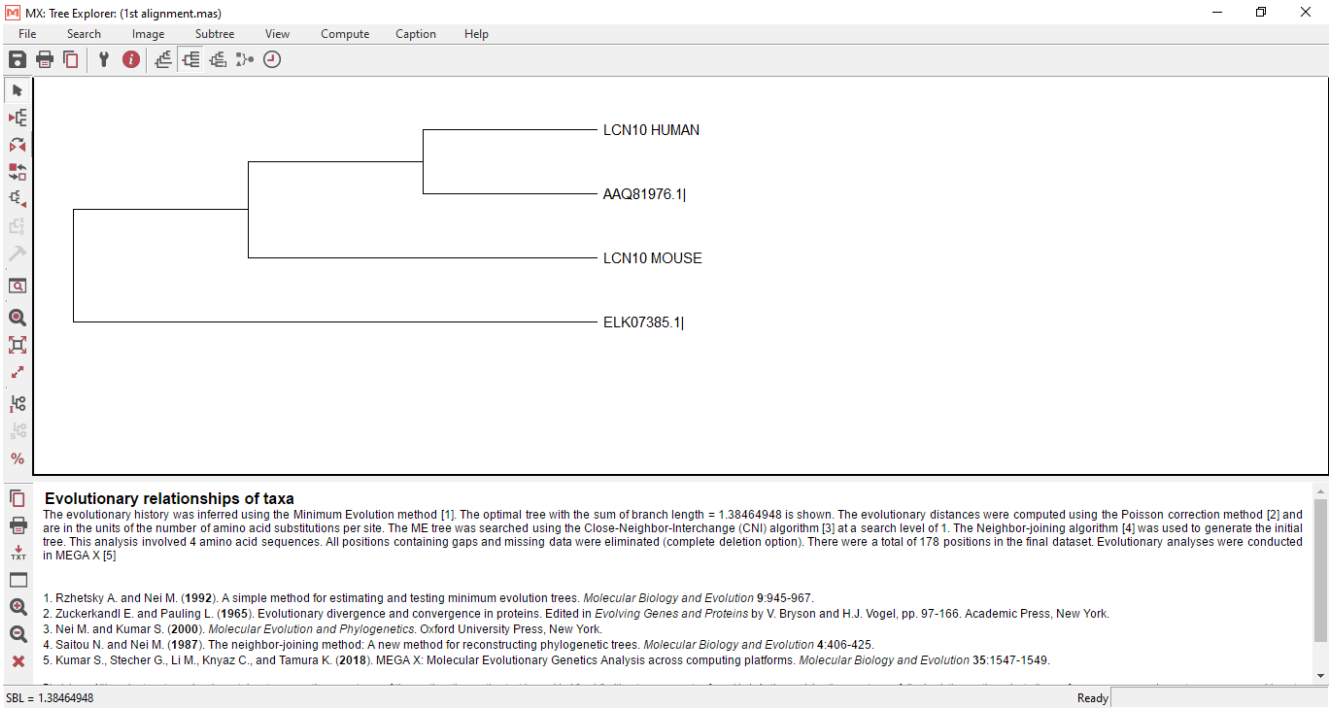
Το δέντρο Neighbor-Joining με bootstrap(500):



Το δέντρο συναίνεσης bootstrap που συνάγεται από 500 επαναλήψεις θεωρείται ότι αντιπροσωπεύει την εξελικτική ιστορία των ταξινομήσεων που αναλύθηκαν. Οι κλάδοι που αντιστοιχούν σε διαμερίσματα που αναπαράγονται σε λιγότερο από 50% επαναλήψεις bootstrap συμπτύσσονται. Το ποσοστό των επαναλαμβανόμενων δέντρων στα οποία ο σχετικός ταξί ομαδοποιήθηκε μαζί στη δοκιμή bootstrap (500 επαναλήψεις) εμφανίζεται δίπλα στους κλάδους.

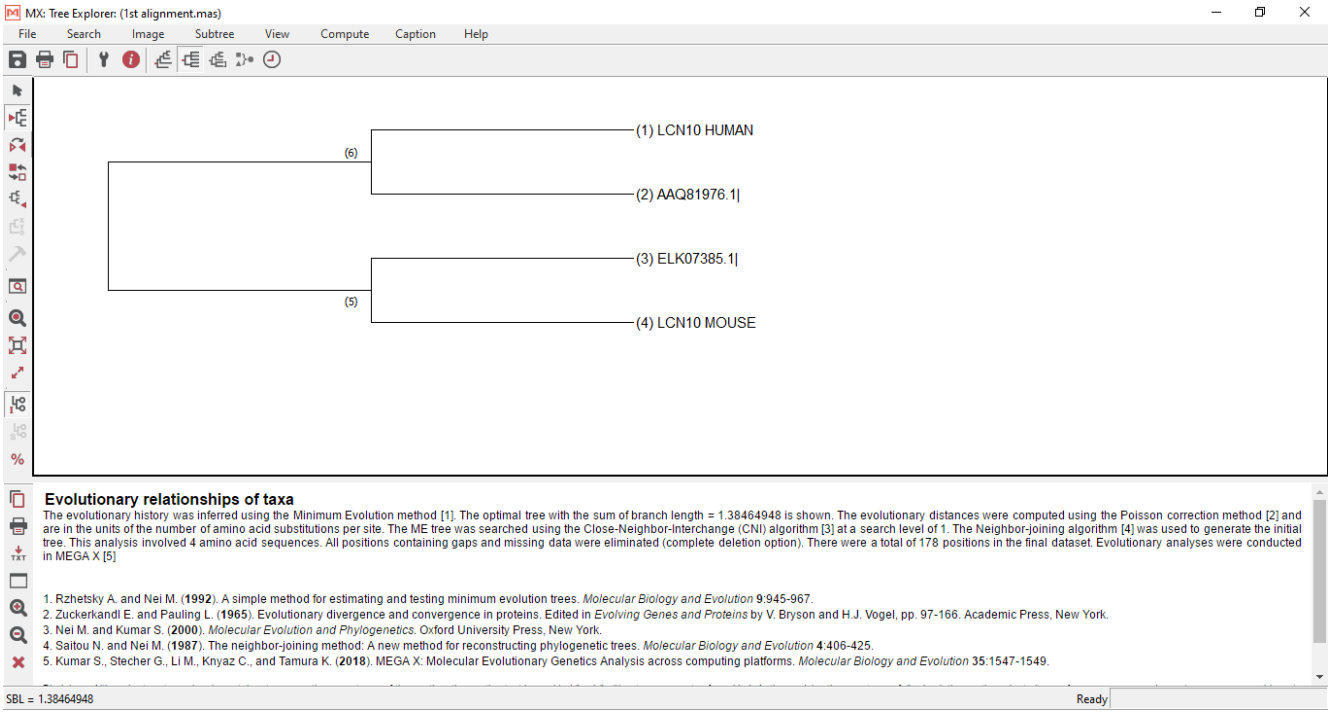
Minimum Evolution Tree

Το δέντρο Minimum Evolution Tree εμφανίζεται στην πιο κάτω εικόνα:

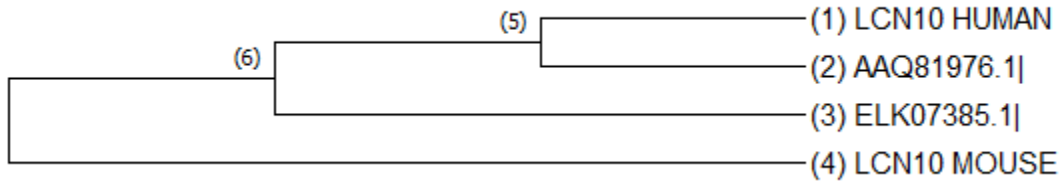


Η εξελικτική ιστορία συνήχθη με τη μέθοδο Minimum Evolution. Εμφανίζεται το βέλτιστο δέντρο με το άθροισμα του μήκους διακλάδωσης =1.38464948. Οι εξελικτικές αποστάσεις υπολογίστηκαν χρησιμοποιώντας τη μέθοδο διόρθωσης Poisson και είναι στις μονάδες του αριθμού υποκαταστάσεων αμινοξέων ανά τοποθεσία. Το δέντρο ME πραγματοποιήθηκε αναζήτηση χρησιμοποιώντας τον αλγόριθμο Close-Neighbor-Interchange (CNI) σε επίπεδο αναζήτησης 1. Ο αλγόριθμος σύνδεσης Neighbor χρησιμοποιήθηκε για τη δημιουργία του αρχικού δέντρου. Αυτή η ανάλυση περιλάμβανε 4 αλληλουχίες αμινοξέων. Όλες οι θέσεις που περιέχουν κενά και ελλείποντα δεδομένα εξαλείφθηκαν (πλήρης επιλογή διαγραφής). Υπήρχαν συνολικά 178 θέσεις στο τελικό σύνολο δεδομένων.

Αλλαγή στην ρίζα του δέντρου:



Το δέντρο Minimum Evolution Tree με bootstrap(500):



Το δέντρο συναίνεσης bootstrap που συνάγεται από 500 επαναλήψεις θεωρείται ότι αντιπροσωπεύει την εξελικτική ιστορία των ταξινομήσεων που αναλύθηκαν. Οι κλάδοι που αντιστοιχούν σε διαμερίσματα που αναπαράγονται σε λιγότερο από 50% επαναλήψεις bootstrap συμπτύσσονται. Το ποσοστό των αναπαραγόμενων δέντρων στα οποία ο σχετικός ταξί ομαδοποιήθηκε μαζί στη δοκιμή bootstrap (500 επαναλήψεις) εμφανίζεται δίπλα στους κλάδους.

ΘΕΜΑ 2

ΆΣΚΗΣΗ 11.4

Αρχικά, έχουμε Q = α,b (Καταστάσεις), από το διάγραμμα καταστάσεων μπορούμε να συμπεράνουμε ότι ο Transition Matrix θα είναι της μορφής,

$$\mathbf{T} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

Στην πρώτη στήλη είναι οι πιθανότητες να μεταβεί στην κατάσταση α, δηλαδή από την α μπορεί να ξαναπάει στην α με p(α) = 0.9 και από την β στην α με p(β -> α) = 0.1. Το ίδιο για την κατάσταση β στην 2^{η} στήλη της μήτρας.

Όσο για τον Emissions Matrix είναι,

$$\mathbf{E} = \begin{bmatrix} 0.4 & 0.1 & 0.4 & 0.1 \\ 0.2 & 0.3 & 0.2 & 0.3 \end{bmatrix}$$

Η αρχική κατανομή πιθανοτήτων για τις καταστάσεις (initial probability distribution) θα είναι,

$$\pi = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

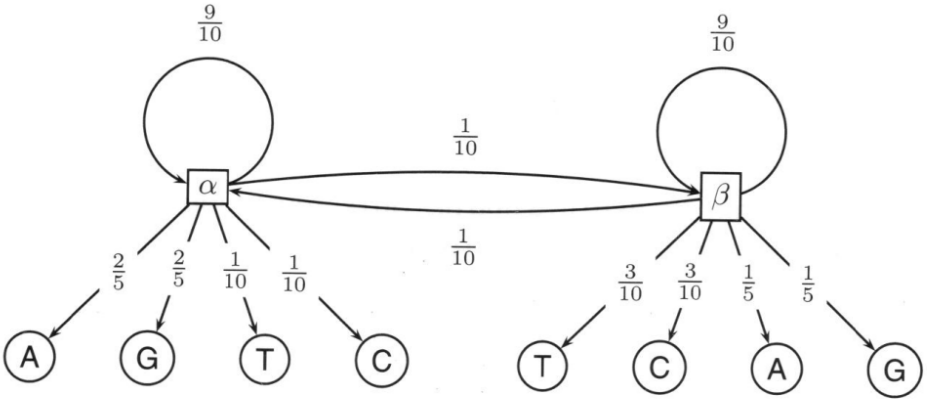
π_i είναι η πιθανότητα που δηλώνει ότι η αλυσίδα Markov θα ξεκινήσει από την κατάσταση i.

Χρησιμοποιείται η συνάρτηση **nt2int** της MATLAB για να κωδικοποιήσουμε την ακολουθία «**GGCT**» σε αριθμούς που τους τοποθετούμε σε ένα διάνυσμα.

Το αποτέλεσμα που παράγεται εξάγεται είναι **-9.7258** και είναι σε μορφή λογαριθμικής πιθανότητας. Επίσης, σαν καλύτερο μονοπάτι συμπεραίνεται ότι είναι το **BBBB** δηλαδή η δεύτερη κατάσταση, για την καλύτερη ακολουθία καταστάσεων.

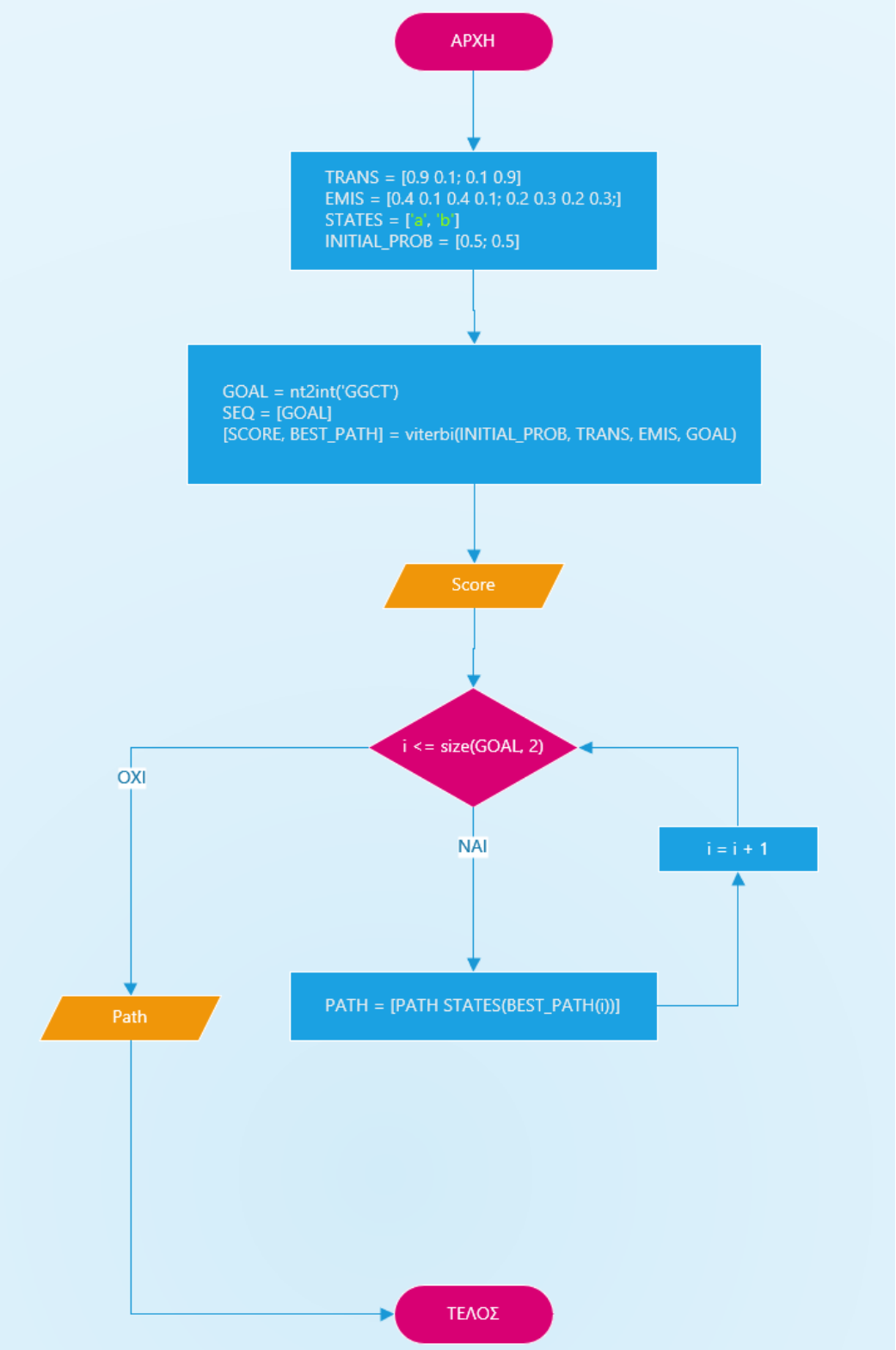
Για τον αλγόριθμο Viterbi (**viterbi.m**) πρώτα μετατρέπουμε τις τιμές των μητρών σε λογαριθμικές πιθανότητες για να μην υπάρχει **underflow**, δηλαδή για να μην προκύψει η περίπτωση που ο υπολογιστής δεν θα μπορεί να αναπαραστήσει μια τιμή επειδή θα είναι πολύ μικρός για αυτόν. Στην συνέχεια, αρχικοποιούμε τον πίνακα **Viterbi** με τις αρχικές πιθανότητες των καταστάσεων και για κάθε σύμβολο της ακολουθίας, υπολογίζουμε την πιθανότητα να εκπεμφθεί δοσμένη της τωρινής κατάστασης και της πιθανότητας της προηγούμενης έχοντας και «βέλη» **οπισθοδρόμησης** για την κατάσταση στην οποία καταλήγουμε κάθε φορά. Τελικά, υπολογίσουμε ολόκληρο το πίνακα **Viterbi** και βρίσκουμε την μέγιστη πιθανότητα στο τέλος ανάμεσα στις δύο καταστάσεις, ξεκινάμε την **οπισθοδρόμηση** (backtracking) με τη βοήθεια των βέλων οπισθοδρόμησης.

Σχήμα:



Σχήμα 11.7 Το HMM που περιγράφεται στο Πρόβλημα 11.4.

Λογικό Διάγραμμα (Flowchart) – ΘΕΜΑΤΟΣ 2:



ΘΕΜΑ 3

Άσκηση 6.13

Εκφώνηση:

Δύο παίκτες παίζουν το ακόλουθο παιχνίδι με δύο αλληλουχίες που έχουν μήκος n και m νουκλεοτίδια αντίστοιχα. Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να αφαιρέσει έναν τυχαίο αριθμό νουκλεοτιδίων από τη μια αλληλουχία ή τον ίδιο (αλλά και πάλι τυχαίο) αριθμό νουκλεοτιδίων και από τις δύο αλληλουχίες. Ο παίκτης που αφαιρεί το τελευταίο νουκλεοτίδιο κερδίζει. Ποιος θα κερδίσει; Περιγράψτε τη νικηφόρα στρατηγική για όλες τις τιμές των n και m.

Υλοποίηση:

Το παιχνίδι ξεκινάει και δίνει την επιλογή στον χρήστη να διαλέξει ποιος παίκτης θα παίξει πρώτος. Αφού επιλεχθεί δείχνει στην έξοδο τις ακολουθίες και ξεκινάει η διαδικασία παιχνιδιού. Όσο οι ακολουθίες m & n είναι διάφορες του μηδενός, δηλαδή δεν έχουν φτάσει στο τέλος του, ο παίκτης1 και ο παίκτης2 συνεχίζουν να παίζουν. Όταν τελειώσει με την κίνηση του ένας από τους δύο παίκτες, τότε η μεταβλητή choice αλλάζει αντίστοιχα για να παίξει ο αντίπαλος παίκτης. Όταν η μεταβλητή choice είναι 1 παίζει ο παίκτης1 και 2 για τον παίκτης2 και καλείται η συνάρτηση Player. Στην συνάρτηση Player, αν οι ακολουθίες m & n ισούνται τότε ο παίκτης1 βρίσκεται σε νικηφόρα θέση διαγράφοντας έναν κάποιον(όχι τυχαίο) αριθμό και από τις δύο

ακολουθίες(move2). Αλλιώς αν η ακολουθία n ισούται με το 0, τότε ο παίκτης1 έχει κερδίσει αφού αφαιρώντας έναν κάποιον(όχι τυχαίο) αριθμό από την ακολουθία n μηδενίστηκε(move1). Επομένως, αφαίρεσε και το τελευταίο νουκλεοτίδιο και νίκησε. Αντίστοιχα ισχύει και για την ακολουθία m όταν αυτή ισούται με 0, για τον παίκτης1. Επίσης, δημιουργήσαμε τη συνάρτηση loser όπου ελέγχει αν η θέση που βρίσκεται ο παίκτης είναι μειονεκτική και μπορεί να ηττηθεί . Αν οι ακολουθίες έχουν το ίδιο μήκος, τότε επιστρέφει τιμή 0. Αν η ακολουθία **m** ισούται με 0 και η **n** είναι μεγαλύτερη από το 0 πάλι θα επιστραφεί η τιμή 0. Αυτό γίνεται διότι από δω και πέρα θα πρέπει να κάνει κινήσεις που αφορούν τη μια ακολουθία, όχι και τις 2.Το ανάποδο, αν η **n** ισούται με 0 και η **m** μεγαλύτερη του μηδενός, η επιστρεφόμενη τιμή θα είναι 0. Τελικά επιστρέφει 1 μόνο όταν οι ακολουθίες είναι διάφορες του μηδενός και όχι ίσες.

Αποτέλεσμα:

```
Select which player will play first, select '1' for Player1 or '2' for Player2 (1/2): 1
Initial sequences :
1st: AAACTTTCGTGCGCGTTGAAAGCTGCTGGCGCGGCGGGGCGGACTCCACCCCTGCCCGGCAGCCCAGCGCCTCCGGCCGCACTTCCAGCTCTCTGCGCAGCCCGCCGCGCAGCCCGC
New sequences:
1st: AATACATGTCTGTATCCCCAGATTCCCAGGACCATATTCAGTATTCTATTCTACACTCAGAGTCTTGCCTTCTTTAATATGCTATGGTTACAAGACACAAACCAAGTGTCCTGATTT
2nd: GGAACCCCTTTCCGGGGAGCTGACCACAGCTCTGTCCTGGCAGGTACAATGCCCAGGAGTACTACGATCGCATTCCTGAGCTTCGGCAGGTCATTGAGCAGCTGAGCAGTGGCTTCT
```

ΘΕΜΑ 4

Άσκηση 6.22

Εκφώνηση:

Διατυπώστε έναν αλγόριθμο που υπολογίζει τη βέλτιστη στοίχιση επικάλυψης και εκτελείται σε χρόνο O(nm).

Υλοποίηση:

Φορτώσαμε τις ακολουθίες και σχηματίσαμε τον πίνακα ομοιότητας όπου έχει 1 μόνο στην κύρια διαγώνιο τού (για τα γράμματα που αντιστοιχούνται μεταξύ τους είναι +1 δηλαδή). Όλα τα υπόλοιπα στοιχεία ισούνται με -1. Δημιουργήσαμε μία μήτρα μηδενικών στον οποίο θα αποθηκεύσουμε τα ταιριάσματα που θα βρούμε για κάθε γράμμα της μία ακολουθίας για όλα τα γράμματα της άλλης με την βοήθεια του πίνακα ομοιότητας. Αυτό γίνεται στο πρώτο διπλό loop όπου εξετάζουμε το ταίριασμα για κάθε στοιχείο. Έπειτα, δημιουργούμε τους πίνακες scores, Seq1, Seq2 όπου θα αποθηκεύσουμε όλα τα score όλων των επικαλύψεων που βρήκαμε, καθώς και όλες τις επικαλύψεις που βρήκαμε. Μετά, δημιουργήσαμε την μήτρα F με τις βαθμολογίες στοίχισης κάθε γράμματος των δύο ακολουθιών παρατηρούμε ότι για να υπολογίσουμε τα σκορ στοίχισης όλων των προθεμάτων της πρώτης ακολουθίας και όλων των επιθεμάτων της δεύτερης αρκεί να πάρουμε τα στοιχεία της αριστερής διαγώνιου της μήτρας με συγκεκριμένη σειρά ώστε να μην προσπελάσουμε το ίδιο στοιχείο δεύτερη φορά. Επειδή όμως για να γίνει αυτό πρέπει ο πίνακας να είναι τετράγωνος, δηλαδή n = m πρέπει να ισχύει και για n>m.

Άρα αυτό που κάνουμε είναι στην αρχή να θέτουμε το n ως n+(m-n), π.χ αν m = 9 και n = 6 τότε το n για την αρχικοποίησης της μήτρας F θα γίνει 6+3 = 9.

Οι επιπλέον στήλες που θα προστεθούν θα γεμίσουν με -1 καθώς θα σημαίνει ότι δεν θα υπάρχει αντιστοιχία με κανένα γράμμα αφού θα είναι ο χαρακτήρας του κενού «-».

Αυτό γίνεται με την βοήθεια της μήτρας F, όπου μέσα στο loop, υπολογίζει όλα τα score στοίχισης από τα νουκλεοτίδια των ακολουθιών. Στις μεταβλητές sequence1 και sequence2 αποθηκεύονται αναδρομικά τα γράμματα της κάθε επικάλυψης κάθε φορά για να αποθηκευτούν στο τέλος της επανάληψης στον αντίστοιχο πίνακα. Μόλις τελειώσει το loop το score που υπολογίστηκε ακριβώς από πάνω, αποθηκεύεται στον πίνακα scores, και τα sequence1 και sequence2 μετατρέπονται σε strings για να μπορέσουμε να τα αποθηκεύσουμε ως ένα στοιχείο και όχι ως αντικείμενο χαρακτήρων. Ακολούθως, πριν την εμφάνιση, βρίσκουμε το μέγιστο score όλων των scores και το αποθηκεύουμε στην μεταβλητή maximum. Αυτό γίνεται για να βρούμε τη καλύτερη στοίχιση επικάλυψης των δύο ακολουθιών. Χρησιμοποιούμε το μέγιστο σκορ που βρήκαμε, για να βρούμε αν υπάρχουν και άλλες επικαλύψεις με το ίδιο σκορ στο πίνακα τον σκορ, και αν υπάρχουν αποθηκεύουμε την θέση τους σε ένα πίνακα indexes. Στο τελικό loop χρησιμοποιούμε όλα τα indexes των επικαλύψεων που πέτυχαν το μεγαλύτερο σκορ επικάλυψης για να τις απεικονίσουμε. Τέλος εμφανίζουμε τις καλύτερες στοιχίσεις επικάλυψης μεταξύ των δύο ακολουθιών μαζί με το σκορ.

Αποτέλεσμα:

```
Elapsed time is 3465.865818 seconds.


The best sequences with the highest score of global-alignment are:
    "C"    "ACGCC"
    "A"    "ATTTC"

With maximum score of: -1
>>
```

Άσκηση 6.23

Εκφώνηση:

Διατυπώστε έναν αλγόριθμο που υπολογίζει τη βέλτιστη στοίχιση προσαρμογής. Εξηγήστε πώς συμπληρώνεται η πρώτη γραμμή και η πρώτη στήλη του πίνακα δυναμικού προγραμματισμού και γράψτε μια σχέση επανάληψης για τη συμπλήρωση του υπόλοιπου πίνακα. Παρουσιάστε μια μέθοδο που βρίσκει την καλύτερη στοίχιση συμπληρωθεί ο πίνακας. Ο αλγόριθμος θα πρέπει να εκτελείται σε χρόνο O(nm).

Υλοποίηση:

Φορτώσαμε τις ακολουθίες και σχηματίσαμε τον πίνακα ομοιότητας όπου έχει 1 μόνο στην κύρια διαγώνιο τού, για τα γράμματα που αντιστοιχούνται μεταξύ τους είναι +1. Όλα τα υπόλοιπα στοιχεία ισούνται με -1. Δημιουργούμε μία μήτρα μηδενικών F διαστάσεων mXn όπου m=μήκος πρώτης ακολουθίας +1 και n=μήκος δεύτερης ακολουθίας + 1. Για την υλοποίηση της άσκησης θα πρέπει στην ουσία να υλοποιήσουμε μια παραλλαγή του αλγόριθμου **Needleman–Wunsch** ώστε να κάνει την στοίχιση χωρίς όμως να αφαιρεί στοιχεία της δεύτερης ακολουθίας από το τελικό αποτέλεσμα.

Για να γεμίσουμε την μήτρα F κάναμε το εξής. Σε όλη την πρώτή γραμμή της F θα υπάρχει το σκορ της πρώτης ακολουθίας με το «κενό» (άρα για i από 1 μέχρι N θα είναι -i), το ίδιο και για τα στοιχεία της πρώτης στήλης για τη δεύτερη ακολουθία. Στο παρακάτω διπλό βρόχο επανάληψης για κάθε κελί θα συγκρίνουμε τη βαθμολογία των γειτονικών κελιών του (δηλαδή πάνω, πάνω-αριστερά και αριστερά κελιά) παίρνοντας κάθε φορά το μέγιστο από αυτά και θέτοντας το στο κελί που είμαστε.

Να σημειωθεί εδώ ότι η σύγκριση των βαθμολογίων των γειτονικών κελιών γίνεται ως εξής:

α) Αν το πάνω-αριστερά κελί έχει αντιστοιχία χαρακτήρων (δηλαδή ο πίνακα ομοιότητας επιστρέφει 1), τότε η βαθμολογία του κελιού αυτού θα είναι η ήδη υπάρχουσα +1 για τη σύγκριση, αλλιώς -1,

β) Για τις περιπτώσεις των αριστερών και πάνω κελιών, που σημαίνουν διαγραφή ή προσθήκη τότε η βαθμολογία του εκάστοτε κελιού θα είναι η ήδη υπάρχουσα -1 για τη σύγκριση. Έχοντας τη μέγιστη τιμή από αυτά τα τρία κελιά γεμίζουμε κάθε φορά και ένα κελί και μέσω αυτής της διαδικασίας γεμίζει ολόκληρη η μήτρα F.

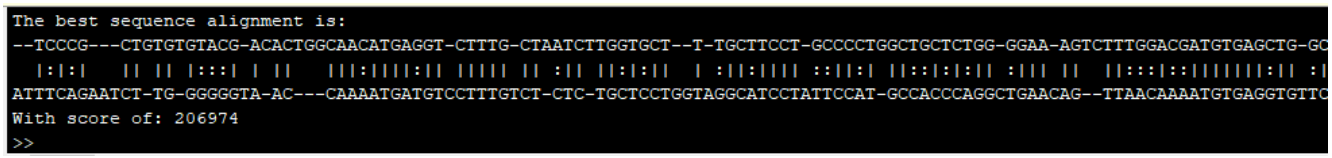
Για την εύρεση καλύτερης στοίχισης προσαρμογής δημιουργούμε 3 πίνακες AlignmentA, AlignmentB, Alignment όπου θα χρησιμοποιηθούν για την εμφάνιση των τελικών ακολουθιών προσαρμογής. Ο πίνακας Alignment είναι για την εμφάνιση των συμβόλων «|,:» που είναι για την αντιστοίχιση συμβόλων. Τώρα για την εύρεση της καλύτερης στοίχισης προσαρμογής από τη μήτρα F θα ακολουθήσουμε την εξής ιδέα :

Αρχικά, θα βρούμε τη καλύτερη βαθμολογία σε όλη την τελευταία στήλη. Αυτό θα το κάνουμε ώστε να ξεκινήσουμε την διαδικασία του backtracking από την αντιστοιχία που δίνει τη μέγιστη συνολική βαθμολογία μεταξύ του τελευταίου γράμματος της δεύτερης ακολουθίας με το εκάστοτε γράμμα της πρώτης ακολουθίας. Άρα ξεκινώντας από αυτή τη θέση στη μήτρα F κάνουμε backtracking εξασφαλίζοντας ότι η ακολουθία προσαρμογής θα μείνει όπως είναι και απλά θα έχουμε πια την πιο όμοια υπο-ακολουθία της πρώτης ακολουθίας για να τη συγκρίνουμε. Τέλος, στη διαδικασία του backtracking, ελέγχουμε κάθε φορά αν με βάση τους κανόνες σύγκρισης των κελιών που αναφέραμε πριν, τα κελιά έχουν τις αντίστοιχες τιμές και ανάλογα συγκρίνουμε αν τα γράμματα των ακολουθιών που συναντάμε στο «μονοπάτι» είναι ίδια η διαφορετικά και ανάλογα αν κάνουμε

αντικατάσταση, προσθήκη ή αφαίρεση προσθέτουμε τους κατάλληλους χαρακτήρες ή σύμβολα στους αντίστοιχους πίνακες Alignment που αναφέραμε πριν.

Σημαντικό είναι να αναφέρουμε ότι λόγω της φύσης του αλγόριθμου Needleman-Wunch μπορεί να υπάρχουν παραπάνω από μία βέλτιστες στοιχίσεις ακολουθιών προσαρμογής επειδή στο backtracking γίνεται διακλάδωση σε διαφορετικά μονοπάτια αν υπάρχει ισότητα στις τιμές των γειτονικών κελιών.

Αποτέλεσμα:



ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ

Π1. ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΗ ΓΟΝΙΔΙΩΜΑΤΙΚΗ: JONATHAN PEVSNER

Π2. https://www.mathworks.com/help/stats/hidden-markov-models-hmm.html

Π3. https://www.mathworks.com/help/stats/hmmdecode.html

Π4. BIOINFORMATICS FOR BEGINNERS: GENES, GENOMES, MOLECULAR EVOLUTION, DATABASES AND ANALYTICAL TOOLS: SUPRATIM CHOUDHURI

Π5. ESSENTIAL BIOINFORMATICS: JIN XIONG

Π6. BIOINFORMATICS: SEQUENCE AND GENOME ANALYSIS: DAVID W. MOUNT

Π7. https://www.youtube.com/watch?v=6JVqutwtzmo

Π8. http://www.scholarpedia.org/article/Viterbi_algorithm

Π9. https://www.youtube.com/watch?v=bQ7kRW6zo9Y

Π10. https://www.youtube.com/watch?v=aD4Cc4L3qW0

Π11. https://vlab.amrita.edu/?sub=3&brch=274&sim=1431&cnt=1

Π12. http://sepwww.stanford.edu/public/docs/sep112/bob2/paper_html/node3.html

Π13. https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch&PROG_DEF=blastn&BLAST_PROG_DEF=blastn&BLAST_SPEC=GlobalAln&LINK_LOC=BlastHomeLink

Π14. https://www.geeksforgeeks.org/backtracking-introduction/

Π15. https://blast.ncbi.nlm.nih.gov/Blast.cgi

Π16. https://www.youtube.com/watch?v=LlnMtI2Sg4g

ΕΠΕΚΤΑΣΕΙΣ/ΕΡΓΑΛΕΙΑ

- **MEGA-X** v.10.1.8
- **MATLAB** v.R2019b
- **BIOINFORMATICS TOOLBOX FOR MATLAB**