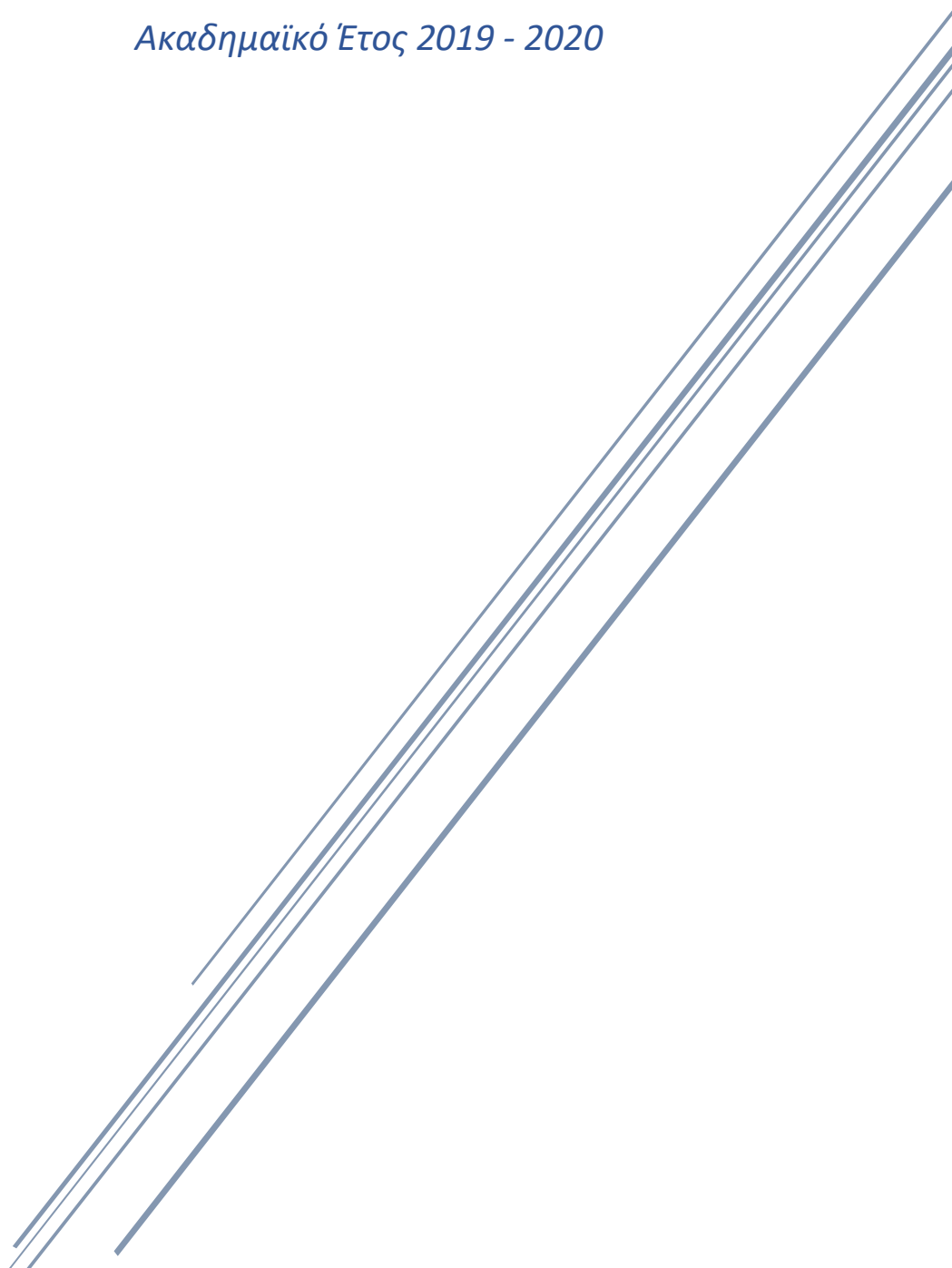


ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ

Απαλλακτικής Εργασία

Ακαδημαϊκό Έτος 2019 - 2020



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Δημήτρης Ματσαγγάνης, Π17068
Αλέξανδρος Σκαρπέλος, Π17122

Περιεχόμενα

Άσκηση 6.13.....	4
1. Εκφώνηση	4
2. Κεντρική Ιδέα Υλοποίησης	4
2.1. Κεντρικό Μενού	4
2.2. Εύρεση ακολουθιών νουκλεοτιδίων	5
2.3. Κύριο Μέρος	5
2.4. Συναρτήσεις Υλοποίησης	6
2.5. Παραδείγματα Στρατηγικής	6
3. Αποτελέσματα	8
Άσκηση 6.23.....	9
1. Εκφώνηση	9
2. Κεντρική Ιδέα Υλοποίησης	9
2.1. Εύρεση ακολουθιών νουκλεοτιδίων	10
2.2. Γέμισμα πίνακα F	10
2.3. Κύριο Μέρος – Εύρεση καλύτερης στοίχισης	11
3. Αποτελέσματα	12
Άσκηση 6.37.....	13
1. Εκφώνηση	13
2. Κεντρική Ιδέα Υλοποίησης	13
2.1. Εύρεση Νουκλεοτιδίων μέσω της ιστοσελίδας RCSB	14
2.2. Κύριο Μέρος	14
2.3. Συνάρτηση checkNext	14
2.4. Συνάρτηση resetCounts.....	15
3. Αποτελέσματα	15
Άσκηση 7.2.....	17
1. Εκφώνηση	17
2. Κεντρική Ιδέα Υλοποίησης	18
3. Ζητούμενα	18
3.1. Ζητούμενο 1	18
3.2. Ζητούμενο 2	18
3.3. Ζητούμενο 3	20



3.4. Ζητούμενο 4	21
3.5. Ζητούμενο 5	24
3.6. Ζητούμενο 6	27
3.7. Ζητούμενο 7	28
Άσκηση 11.4.....	30
1.Εκφώνηση	30
2. Κεντρική Ιδέα Υλοποίησης	30
2.1. Δηλώσεις Μεταβλητών	30
2.2. Αλγόριθμος Viterbi.....	31
2.3. Κύριο Μέρος	31
2.4. Τελικό Στάδιο	31
3. Αποτελέσματα	32
Απαραίτητα Εργαλεία	33
Βιβλιογραφία.....	33
Περιεχόμενα Απεσταλμένου Αρχείου	33

Άσκηση 6.13

1. Εκφώνηση

Δυο παίκτες παίζουν το παρακάτω παιχνίδι με δύο χρωμοσώματα που έχουν μήκος n και m νουκλεοτίδια αντίστοιχα.

Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να αφαιρέσει έναν τυχαίο αριθμό νουκλεοτιδίων από τη μία 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 την νικηφόρα στρατηγική για όλες τις τιμές n και m .

2. Κεντρική Ιδέα Υλοποίησης

Το πρόγραμμα μας αποτελείται από ένα αρχείο το οποίο πρέπει και να εκτελέσουμε ώστε να δούμε τα αποτελέσματα.

Σε αυτήν την άσκηση προσπαθούμε να ανακαλύψουμε ποιος χρήστης θα κερδίσει για αυτό εκτελώντας το πρόγραμμα υπάρχει κεντρικό μενού στο οποίο επιλέγουμε ποιος παίκτης θα ξεκινήσει αλλά και ποια αρχεία θα χρησιμοποιηθούν.

2.1. Κεντρικό Μενού

Υλοποιήσαμε την άσκηση αυτή και προσπαθήσαμε να την κάνουμε πιο ευχάριστη για τον χρήστη με την ύπαρξη κεντρικού μενού μέσω του οποίου μπορεί να επιλεγεί ποιος από τους δύο παίκτες θα ξεκινήσει καθώς και ποια δύο αρχεία θα επιλεγούν (από τα τρία διαθέσιμα: liver, brain, muscle).

Ακολουθεί ενδεικτική εικόνα:

```
1. PYGL glycogen phosphorylase L (liber.txt)
2. PYGM glycogen phosphorylase, muscle associated (muscle.txt)
3. PYGB glycogen phosphorylase B (brain.txt)
Select the first out of the 3 following Isoenzymes to implement it's sequence (ex. 3):1
Select the second out of the 3 following Isoenzymes to implement it's sequence (ex. 2):2
Game starts with /n Player 1 (Random Player) and Player 2 (Tactic Player)
Please select who plays first, please type the player number (1/2): 1|
```

Εικόνα Αποτελεσμάτων 1

2.2. Εύρεση ακολουθιών νουκλεοτιδίων

Αρχικά, κατεβάζουμε από την ιστοσελίδα της NCBI (όπως μας το συστήσατε) όπου θα κατεβάσουμε και την ακολουθία FASTA και θα τη μετατρέψουμε σε αρχείο txt.

Έπειτα στο Matlab χρησιμοποιώντας την εντολή *fastaread* θα δημιουργηθεί ένα αντικείμενο με τον header και την ακολουθία από το αρχείο και εμείς θα επιλέξουμε την ακολουθία για να υλοποιήσουμε την υπόθεση της άσκησης.

2.3. Κύριο Μέρος

Έπειτα από το κεντρικό μενού αναγράφονται οι αλληλουχίες των νουκλεοτιδίων που επιλέξαμε (τα δύο από τα τρία αρχεία που επιλέγει ο χρήστης παραπάνω).

Στην συνέχεια κάνει κίνηση δηλαδή αφαιρεί νουκλεοτίδια ο παίχτης που επιλέξαμε εμείς στο κεντρικό μενού.

Για αυτόν τον λόγο δημιουργήσαμε συναρτήσεις αναφορικά με τους παίχτες (τις Player1, Player2 και την losing_position).

2.4. Συναρτήσεις Υλοποίησης

Για την επιλογή του τυχαίου αριθμού, ο παίκτης1 θα επιλέξει τυχαία αναμεσά στις κινήσεις `move1` & `move2`, για να αφαιρέσει οποιονδήποτε τυχαίο αριθμό από ολόκληρη/ρες την ακολουθία/ες.

Στο τέλος της συνάρτησης, επιστρέφονται οι νέες ακολουθίες μετά την αφαίρεση νουκλεοτιδίων.

Για την συνάρτηση `Player2`, όπου σε αυτή την περίπτωση ο παίκτης δεν παίζει τυχαία, ακολουθεί η ίδια λογική στην περίπτωση της αφαίρεσης ενός (όχι τυχαίου) αριθμού.

Στην περίπτωση όπου ο παίκτης1 παίζει τυχαία, εδώ δημιουργήσαμε τη συνάρτηση `losePosi` όπου ελέγχει αν η θέση που βρίσκεται ο παίκτης2 είναι μειονεκτική (`losing position`).

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.

Αντιστρόφως ανάλογα, 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.

2.5. Παραδείγματα Στρατηγικής

Σε αυτή την ενότητα θα παρουσιαστεί ένα σύντομο παράδειγμα που θα αναδεικνύει τη σειρά αφαίρεσης νουκλεοτιδίων την οποία ακολούθησε ο κάθε παίκτης για να νικήσει, καθώς και τον νικητή στη κάθε περίπτωση.

Για παράδειγμα: Η στρατηγική παιξίματος κάθε φορά έχει να κάνει με το ποιος παίκτης θα φτάσει πρώτος στα 3 νουκλεοτίδια, αν φτάσει ο παίκτης1 πρώτος στα 3 νουκλεοτίδια και είναι η σειρά του παίκτης2, τότε όποιο αριθμό νουκλεοτιδίων αφαιρέσει, ο παίκτης1 θα κερδίσει. Υποθετικά βγάζει 2 νουκλεοτίδια ο παίκτης2. Τώρα ο παίκτης1 μπορεί να αφαιρέσει το 1 νουκλεοτίδιο και να κερδίσει. Αν από την άλλη ο παίκτης2 αφαιρέσει 1



νουκλεοτίδιο, τότε ο παίκτης2 πάλι μπορεί να κερδίσει αφαιρώντας και τα 2 νουκλεοτίδια.

Επεκτείνοντας τη στρατηγική που προαναφέρθηκε για η αριθμό νουκλεοτιδίων έχουμε τα ακόλουθα.

Αν πάρουμε για παράδειγμα 10 νουκλεοτίδια, σημασία έχει ποιος θα φτάσει πρώτος στα 9 νουκλεοτίδια, για να μπορεί να οδηγεί το παιχνίδι μέχρι το 3 όπου και είναι ο απώτερος σκοπός για να κερδίσει.

Ο παίκτης1 παίζει και αφαιρεί 1, άρα έχουμε 9 νουκλεοτίδια και είναι η σειρά του παίκτης2. Αν και εκείνος αφαιρέσει 2 τότε θα πάμε στα 7.

Η επόμενη κίνηση του παίκτη1 είναι να αφαιρέσει 1 νουκλεοτίδιο για να πάει στα 6.

Αν ο παίκτης2 αν αφαιρέσει 2, θα πάμε στα 4 νουκλεοτίδια και ο παίκτης1 θα αφαιρέσει 1 νουκλεοτίδιο οπότε θα είναι και αυτός που θα κερδίσει γιατί έφτασε πρώτος στα 3.

Τώρα αν πάλι ο παίκτης2 αφαιρέσει 1, θα πάμε στα 5 νουκλεοτίδια και ο παίκτης1 θα αφαιρέσει 2 αυτή τη φορά για να φτάσει πρώτος στα 3 και θα κερδίσει το παιχνίδι.

Παρατηρούμε δηλαδή πως η στρατηγική επεκτείνεται στους περιττούς αριθμούς κάθε φορά και σημασία έχει ποιος θα φτάσει στο 9, μετά στο 7, 5 και τελικά στο 3.

Αν γίνει κάποιο “λάθος” και φτάσει ο παίκτης2 πρώτος στο 7 ή στο 9 (όπως στο παραπάνω παράδειγμα) τότε ο παίκτης1 θα πρέπει να κάνει κίνηση η οποία δεν θα επιτρέπει στον παίκτης2 να φτάσει πρώτος στον ζητούμενο περιττό αριθμό, 3.

Στην περίπτωση όπου ο αριθμός των νουκλεοτιδίων είναι περιττός πχ: $n=11$ τότε ο παίκτης ο οποίος παίζει πρώτος, έχει από την αρχή το προβάδισμα διότι ο αριθμός είναι ήδη περιττός. Άρα θα πρέπει να κρατήσει αυτή τη λογική και να στέλνει στον αντίπαλο παίκτη άρτιους.

Οποιοσδήποτε και να είναι ο αριθμός του n , αρκεί κάθε φορά ο παίκτης που θα επιχειρήσει την νικηφόρα στρατηγική να έχει κατά νου ότι πρέπει να φτάσει πρώτος στους περιττούς αριθμούς και να στέλνει στον αντίπαλο τους άρτιους.

3. Αποτελέσματα

Ακολουθούν τα αποτελέσματα της παραπάνω υλοποίησης (ενδεικτικά από την αρχή και το τέλος των αποτελεσμάτων):

```
Select the first out of the 3 following Isoenzymes to implement it's sequence (ex. 3):1
Select the second out of the 3 following Isoenzymes to implement it's sequence (ex. 2):2
Game starts with /n Player 1 (Random Player) and Player 2 (Tactic Player)
Please select who plays first, please type the player number (1/2): 1
Initial sequences :
AAACTTTTCGTGCGCGTTGAAAGCTGCTGGCGCGGCGGGGCGGACTCCACCCCTGCCCGGCAGCCAGCGCCTCCGGCCGCACTTCCAGCTCTCTGCGCAGCCCGCCGCGCAG
New sequences:
CTTACTGTCATTAAACCAGTATGATGATGGAGTGGGGAATGCCTCAGTTTCTCCACCGCCGAGGGCCTATAGGACTACAGTTCCCAGGGCCCCGTGCTGCAGGCTAGCGGC
ACTTACTGTCATTAAACCAGTATGATGATGGAGTGGGGAATGCCTCAGTTTCTCCACCGCCGAGGGCCTATAGGACTACAGTTCCCAGGGCCCCGTGCTGCAGGCTAGCGGC

Player 2 made a move
New sequences:
GGGAGAAAAGAGCTTGTTTGGGGGCGCTTGGCTCTAACACTTAGCTTTCTGTATCCTCCGGGCGCTCAGTTTCTTCATTACAAAGGAGGGGATGAAAAGCTGAGCAGAGAAGGG
TGGGAGAAAAGAGCTTGTTTGGGGGCGCTTGGCTCTAACACTTAGCTTTCTGTATCCTCCGGGCGCTCAGTTTCTTCATTACAAAGGAGGGATGAAAAGCTGAGCAGAGAAGG

Player 1 made a move
New sequences:
GGAGGCCCTGCAGGGGCGATGTGGCAGCCCTGGGGAGCAGCAGGCTGGACCTGGGTTTTGACCCTGGGGCATGGGACTTCTCAGCTTTTCTCTGGAAGAGGAGCCAGGAAC
TGGGAGAAAAGAGCTTGTTTGGGGGCGCTTGGCTCTAACACTTAGCTTTCTGTATCCTCCGGGCGCTCAGTTTCTTCATTACAAAGGAGGGATGAAAAGCTGAGCAGAGAAGG

Player 2 made a move
New sequences:
GCAGGGCTTTGGTGGCCCTGGTTGGGATGAGGCAGAAAGGTGGAGATCCTGCCAGCAGAGTGAACCAGAGCTTCCCTTTGACCTGCAGAACCCAGAGAGTGGACGCGGATGG
AATAGCACATGCCTATCCTTTCCCTCCAGGTTTAAAGTCTTCGCAGATTATGAAGACTACATTAATGCCAGGAGAAAGTCAGCGCCTTGTAACAAGGTGAGGGGTCTTGGG
```

Εικόνα Αποτελεσμάτων 2

```
Player 1 made a move
New sequences:
AGCCTCCTTA
GAGCCTCCTTA

Player 2 made a move
New sequences:
TTA
GAGCCTCCTTA

Player 1 made a move
New sequences:
CCTCCTTA

Player 1 made the last move and won the game!
```

Εικόνα Αποτελεσμάτων 3

Άσκηση 6.23

1. 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 $O(nm)$.

Επεξήγηση: Γενικότερα οι δύο βασικές μέθοδοι στοίχισης είναι δύο: η καθολική και η τοπική. Υπάρχει και μια μέση κατάσταση, η οποία είναι η ημικαθολική και μοιάζει στην προσαρμοστική αλλά δεν υπολογίζονται τα 'κοστοβόρα' κενά που βρίσκονται στην άκρη. Με βάση της ημικαθολικής στοίχισης θα υλοποιηθεί η άσκηση.

2. Κεντρική Ιδέα Υλοποίησης

Φορτώνουμε τις ακολουθίες και σχηματίζουμε τον πίνακα ομοιότητας (scoring_matrix) όπου έχει 1 μόνο στην κύρια διαγώνιο τού (για τα γράμματα που αντιστοιχούνται μεταξύ τους είναι +1 δηλαδή, Α με Α, Τ με Τ κλπ.).

Όλα τα υπόλοιπα στοιχεία ισούνται με -1. Δημιουργούμε μία μήτρα μηδενικών F, διαστάσεων $m \times n$, όπου m = μήκος πρώτης ακολουθίας + 1, και n = μήκος δεύτερης ακολουθίας + 1.

Για την υλοποίηση της άσκησης θα πρέπει στην ουσία να υλοποιήσουμε τον αλγόριθμο Needleman–Wunsch με κάποιες αλλαγές ώστε να κάνει την στοίχιση χωρίς όμως να αφαιρεί στοιχεία της δεύτερης ακολουθίας από το τελικό αποτέλεσμα.

2.1. Εύρεση ακολουθιών νουκλεοτιδίων

Αρχικά, κατεβάζουμε από την ιστοσελίδα της NCBI (όπως μας το συστήσατε) όπου θα κατεβάσουμε και την ακολουθία FASTA και θα τη μετατρέψουμε σε αρχείο txt.

Έπειτα στο Matlab χρησιμοποιώντας την εντολή *fastaread* θα δημιουργηθεί ένα αντικείμενο με τον header και την ακολουθία από το αρχείο και εμείς θα επιλέξουμε την ακολουθία για να υλοποιήσουμε την υπόθεση της άσκησης.

2.2. Γέμισμα πίνακα F

Σε όλη την πρώτη γραμμή της F θα μέχρι στιγμής 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 του κελιών (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 βαθμολογιών 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 γειτονικών 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 ολόκληρος ο πίνακας F.

2.3 Κύριο Μέρος – Εύρεση καλύτερης στοίχισης

Δημιουργούμε 3 πίνακες, αρχικά κενούς και στη συνέχεια θα γεμίσουν ως εξής :

- MatrixA: Πίνακας εμφάνισης της τελικής 1^{ης} ακολουθίας.
- MatrixB: Πίνακας εμφάνισης της τελικής 1ης ακολουθίας.
- MatrixC: Πίνακας εμφάνισης των συμβόλων “|, :” που είναι απαραίτητα για την αντιστοίχιση συμβόλων.

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 (ακολουθίας προσαρμογής) 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 υπό-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 κελία έχουν τις αντίστοιχες τιμές και ανάλογα συγκρίνουμε αν τα γράμματα των ακολουθιών που συναντάμε στο μονοπάτι είναι ίδια ή διαφορετικά, και ανάλογα αν κάνουμε αντικατάσταση, προσθήκη ή αφαίρεση προσθέτουμε τους κατάλληλους χαρακτήρες ή σύμβολα στους αντίστοιχους πίνακες Matrix που αναφέραμε παραπάνω.

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

Άσκηση 6.37

1. Εκφώνηση

Επινοήστε έναν αποδοτικό αλγόριθμο για το πρόβλημα της Χιμαιρικής Στοιχείωσης.

Κανόνες:

Ένας ιός μολύνει ένα βακτήριο και τροποποιεί μια διεργασία αναδιπλασιασμού στο βακτήριο προσθέτοντας:

- Σε κάθε A, ένα πολύ A με μήκος από 1 έως 5.
- Σε κάθε C, ένα πολύ C με μήκος από 1 έως 10.
- Σε κάθε G, ένα πολύ G με τυχαίο μήκος $\Rightarrow 1$.
- Σε κάθε T, ένα πολύ T με τυχαίο μήκος $\Rightarrow 1$.

Δεν επιτρέπονται κενά ή άλλες προσθήκες στο DNA που έχει τροποποιηθεί από τον ιό.

Παράδειγμα: Η αλληλουχία “AAATAAGGGGCCCCCTTTTTTCC” αποτελεί μολυσμένη έκδοση της “ATAGCTC”.

2. Κεντρική Ιδέα Υλοποίησης

Αρχικά, θα πρέπει να ελέγχουμε κάθε φορά ποιο νουκλεοτίδιο βρίσκεται προς έλεγχο (ξεκινώντας προφανώς από το πρώτο αριστερά) και να λάβουμε υπόψιν μας τι γίνεται μετά από την εμφάνιση του κάθε νουκλεοτιδίου (βλέπε κανόνες από την εκφώνηση).

Ακόμη, θα ορίσουμε έναν μετρητή που θα μετράει κάθε σύμβολο και θα του ορίζει το μέγιστο δυνατό αριθμό επαναλήψεων. Μέσω αυτού θα γίνεται και ο πολλαπλασιασμός των συμβόλων.

2.1. Εύρεση Νουκλεοτιδίων μέσω της ιστοσελίδας RCSB

Ψάχνοντας τον κωδικό 1BBT για το καψίδιο βρίσκουμε στην ιστοσελίδα του NCIB τις 4 ακολουθίες αμινοξέων για τις 4 μορφές του FOOT AND MOUTH DISEASE VIRUS.

Έχοντας τις 4 ακολουθίες αμινοξέων για κάθε μία χρησιμοποιούμε την εντολή του Matlab *aa2nt* (όπου μετατρέπει ακολουθίες αμινοξέων σε ακολουθίες νουκλεοτιδίων).

Με αυτό τον τρόπο, επιτυγχάνουμε να αποτυπώσουμε την νουκλεοτιδική ακολουθία του ιού.

2.2 Κύριο Μέρος

Κατά την εκκίνηση του προγράμματος, φορτώνουμε την ακολουθία που είναι προς επεξεργασία στη μεταβλητή *infected_seq*

Δημιουργούμε τον πίνακα *original_seq* αρχικά ως έναν κενό πίνακα, όπου θα αποθηκεύσει την νέα ακολουθία, και το πίνακα *counter* όπου είναι ο μετρητής για τις συνεχόμενες εμφανίσεις του κάθε γράμματος στην ακολουθία.

Στο δεύτερο loop ελέγχεται αν το επόμενο στοιχείο της ακολουθίας είναι ίδιο με το αυτό στο οποίο βρίσκεται τώρα, όπου στην αρχή αρχικοποιούμε ως πρώτο στοιχείο το πρώτο γράμμα της ακολουθίας.

Αν δεν είναι τότε προσθέτει στο πίνακα *original_seq* το παρόν στοιχείο και καλεί την συνάρτηση *resetCounts* όπου θα επαναφέρει το μετρητή στο 0 με το επόμενο στοιχείο γίνεται το παρόν και το loop συνεχίζει.

2.3 Συνάρτηση checkNext

Αρχικά, έχουμε την συνάρτηση checkNext που ελέγχει κάθε φορά ποιο στοιχείο διαβάζει γιατί από αυτό εξαρτάται και οι φορές που μπορεί να εμφανιστεί το αντίστοιχο νουκλεοτίδιο στην νέα ακολουθία. Επιπλέον, ελέγχεται αν το τωρινό στοιχείο έχει ελεγχθεί ήδη.

Ακόμη, μέσω αυτής της συνάρτησης ορίζονται και τα όρια (limits) της δυνατής επανάληψης του κάθε στοιχείου.

Έτσι το γράμμα A μπορεί να έχει μέχρι και 6 συνεχόμενα (max 5 νέα + 1 που προ υπήρχε), το γράμμα C μέχρι και 11 (10 + 1), ενώ για τα γράμμα G και T, δεν δίνεται τερματικός αριθμός εμφανίσεων (άπειρο).

Η συνάρτηση επιστρέφει την νέα ακολουθία (τη μη μολυσμένη), την λογική τιμή αν έχει υπερβεί κάποιο όριο και τον αριθμό των εμφανίσεων του κάθε γράμματος.

2.4. Συνάρτηση resetCounts

Η συνάρτηση resetCounts δέχεται ως όρισμα το επόμενο στοιχείο της ακολουθίας, τα πλήθη των στοιχείων την ακολουθία και το τωρινό στοιχείο της ακολουθίας.

Έπειτα, επαναφέρει τον αριθμό εμφανίσεων όλων των στοιχείων σε 0 και απλά προσθέτει 1 στο στοιχείο που είναι το επόμενο στην ακολουθία. Τέλος, επιστρέφει τον αριθμό εμφανίσεων του κάθε στοιχείου.

3. Αποτελέσματα

Ακολουθούν τα αποτελέσματα της παρούσας υλοποίησης :

Εάν εκτελέσουμε τη παρούσα υλοποίηση για τις 4 ακολουθίες αμινοξέων για τις 4 μορφές του FOOT AND MOUTH DISEASE VIRUS, έχουμε τα παρακάτω ενδεικτικά αποτελέσματα:

Below follows the bacterial DNA sequence after being infected by the virus :

GGTGCAGGTCAATCCAGTCCCGCCACAGGTAGCCAGAACAGTCCGGGAACACCGGGAGTATAATCAATAATTATTATATGCAGCAATATCAAAACAGTATGGATACCAACTAGGGAACGACGCAATCAGCGGTGGTAGCAATGAGGGT

Below follows the bacterial DNA sequence before being infected by the virus :

GTGCAGTCATCAGTCCGACAGTAGCAGACAGTCCGACAGGATATCATATATATATGCAGCATATCACAGTATGATACACTAGACGACGCATCAGCGTGTAGCATGAGCAGCAGGATACACAGTACTCAGACTACTATACGCAGACAT

Εικόνα Αποτελεσμάτων 8



Εάν εκτελέσουμε το υλοποιημένο πρόγραμμα με την ακολουθία που ορίζει το παράδειγμα της εκφώνησής, τότε λαμβάνουμε τα παρακάτω αποτελέσματα (βρίσκονται στο πρόγραμμα υπό τη μορφή σχολίου):

```
% Exercise's Example.  
infected_seq = ('AAATAAGGGGGCCCCCTTTTTTCC');
```

Εικόνα Αποτελεσμάτων 9

Below follows the bacterial DNA sequence after being infected by the virus :

AAATAAGGGGGCCCCCTTTTTTCC

Below follows the bacterial DNA sequence before being infected by the virus :

ATAGCTC

Εικόνα Αποτελεσμάτων 10

Άσκηση 7.2

1.Εκφώνηση

Πραγματοποιήστε φυλογενετικές αναλύσεις χρησιμοποιώντας το λογισμικό MEGA.

1. Μεταβείτε στη βάση δεδομένων των συντηρημένων δομικών επικρατειών (<https://www.ncbi.nlm.nih.gov/cdd>) στο NCBI.
2. Εισάγετε τον όρο λιποκαλίνες (lipocalins) ή άλλο όνομα οικογένειας της επιλογής σας. Εναλλακτικά, μπορείτε να ξεκινήσετε από την Ensembl, τη HomoloGene ή την Pfam.
3. Επιλέξτε τη μορφή αρχείου mFASTA και στη συνέχεια κάντε κλικ στην επιλογή «Reformat». Το αποτέλεσμα είναι μια πολλαπλή στοίχιση αλληλουχιών. Αντιγράψτε το αποτέλεσμα σε έναν επεξεργαστή κειμένου (π.χ. NotePad++) και απλοποιήστε τα ονόματα των αλληλουχιών.
4. Εισάγετε το αρχείο(ή επικολλήστε τις αλληλουχίες) στο MEGA, όπως φαίνεται στην **εικόνα 7.9**. Στοιχίστε τις αλληλουχίες και αποθηκεύστε τις σε μορφή .mas και .meg.
5. Επιλέξτε Phylogeny>Construct/Test για να δημιουργήσετε δέντρα με τις μεθόδους ένωσης γειτόνων, μέγιστης πιθανοφάνειας ή άλλες.
6. Για κάθε δέντρο που δημιουργείτε, διαβάστε την αντίστοιχη λεζάντα. Δοκιμάστε τα εργαλεία δέντρων (π.χ. τοποθετήστε μια ρίζα, αναστρέψτε κόμβους, εμφανίστε ή αποκρύψτε τα μήκη των κλάδων, αλλάξτε μορφές απεικόνισης).
7. Πραγματοποιήστε bootstrapping. Προσδιορίστε τις συστάδες κλάδων που έχουν χαμηλά επίπεδα στήριξης. Γιατί συμβαίνει αυτό;

2. Κεντρική Ιδέα Υλοποίησης

Στη παρούσα άσκηση θα υλοποιήσουμε μια σειρά από εντολές και ζητούμενα με τη βοήθεια του λογισμικού MEGA και της βάσης δεδομένων του NCBI.

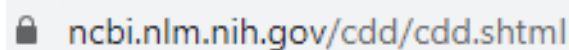
Τα υλοποιημένα θα αναλυθούν εκτενώς μέσω ενδεικτικών εικόνων από τη διαδικασία και απαραίτητων επεξηγήσεων.

3. Ζητούμενα

Στην παρούσα ενότητα θα επιλύσουμε και θα αναλύσουμε σε βάθος το κάθε ζητούμενο ξεχωριστά.

3.1. Ζητούμενο 1

Για το πρώτο ζητούμενο της άσκησης αρκεί να μετάβουμε στον ιστότοπο των συντηρημένων δομικών επικρατειών του NCBI, που μας παραθέτει η εκφώνηση.




Εικόνα Αποτελεσμάτων 11

3.2. Ζητούμενο 2

Στο δεύτερο ζητούμενο μας ζητείται να αναζητήσουμε, (στον ιστότοπο που ήδη έχουμε μεταβεί, βλέπε Ζητούμενο 1) τον όρο λιποκαλίνες, όπως μας προτρέπει.

Ακολουθεί σχετική εικόνα από τα αποτελέσματα που μας εμφανίστηκαν (από τη πηγή Pfam) καθώς και από την εύρεση της ζητούμενης ακολουθίας (τα σχετικά αρχεία είναι τα *sequence.txt* & *sequence_original.fasta*).



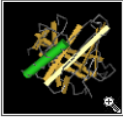
Conserved Protein Domain Family

Lipocalin

HOME | SEARCH | SITE MAP | Entrez | CDD | Structure | Protein | Help

COVID-19 is an emerging, rapidly evolving situation.
Get the latest public health information from CDC: <https://www.coronavirus.gov>.
Get the latest research from NIH: <https://www.nih.gov/coronavirus>.
Find NCBI SARS-CoV-2 literature, sequence, and clinical content: <https://www.ncbi.nlm.nih.gov/sars-cov-2/>.

pfam00061: Lipocalin [Download alignment](#) [?](#)



Lipocalin / cytosolic fatty-acid binding protein family
Lipocalins are transporters for small hydrophobic molecules, such as lipids, steroid hormones, bilins, and retinoids. The family also encompasses the enzyme prostaglandin D synthase (EC:5.3.99.2). Alignment subsumes both the lipocalin and fatty acid binding protein signatures from PROSITE. This is supported on structural and functional grounds. The structure is an eight-stranded beta barrel.

Links [?](#)
Source: pfam
Taxonomy: Bilateria
Protein: [Representatives](#)
[Specific Protein](#)
[Related Protein](#)
[Related Structure](#)
[Architectures](#)
Superfamily: [cl21528](#)

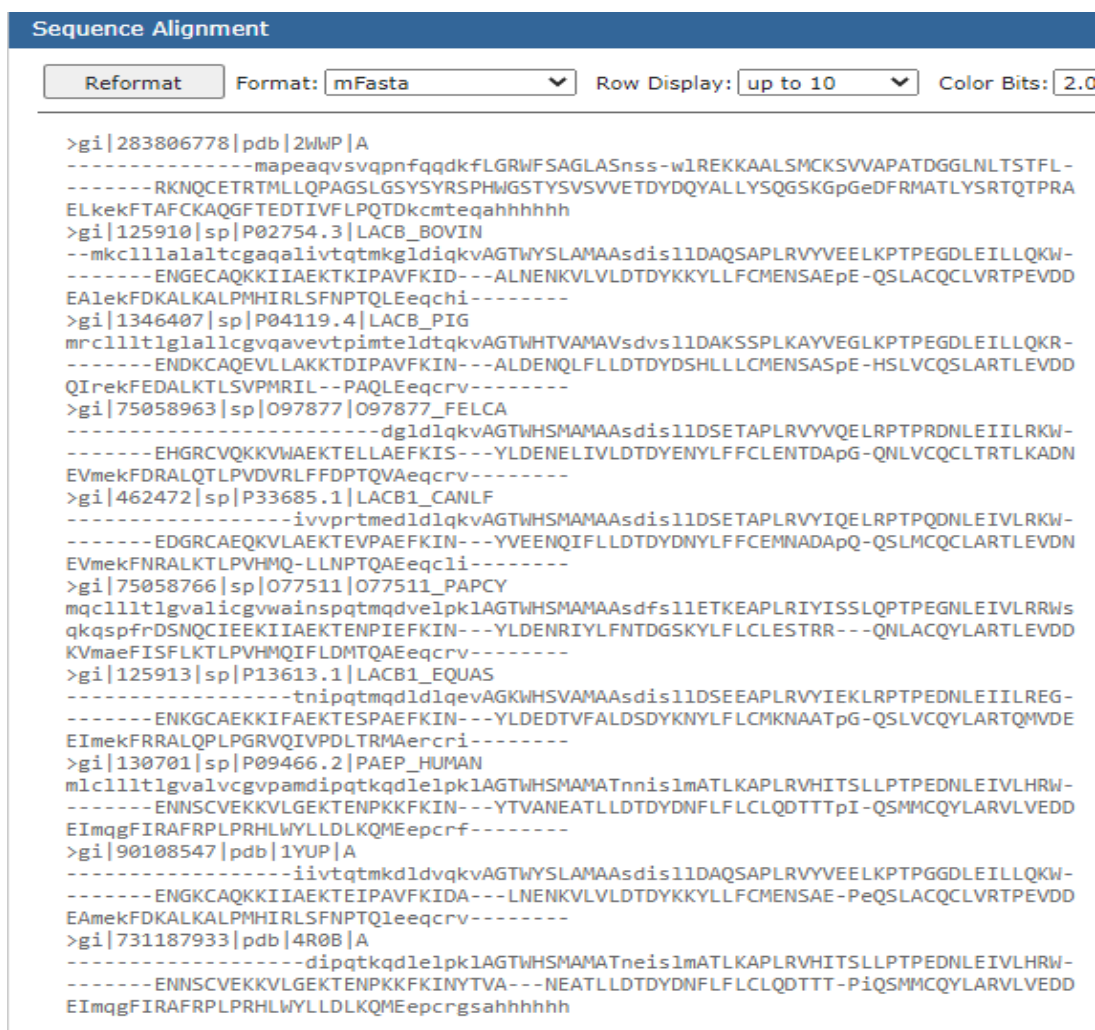
pfam00061 is a member of the superfamily [cl21528](#).

Εικόνα Αποτελεσμάτων 12

3.3. Ζητούμενο 3

Έπειτα, επιλέγουμε – όπως αναφέρεται – τη μορφή αρχείου mFASTA στη συνέχεια κάνουμε κλικ στην επιλογή “Reformat”.

Τέλος, αφού αντιγράψουμε τα αποτελέσματα της στοίχισης στο κειμενογράφο NotePad++ λαμβάνουμε την παρακάτω εικόνα και απλοποιούμε τα ονόματα των αλληλουχιών.



```

Sequence Alignment

Reformat  Format: mFasta  Row Display: up to 10  Color Bits: 2.0

>gi|283806778|pdb|2WNP|A
-----mapeaqvsvqpnfqgdkfLGRWFSAGLASnss-wlREKKAALSMCKSVVAPATDGGNLNSTFL-
-----RKNQCETRTMLLPAGSLGSYSYSPHWGSTYSVSVVETDQYALLYSQGSKGpGeDFRMTLYSRTQTPRA
ELkekFTAFCQAQGFTEdTIVFLPQTDkcmteqahhhhhh
>gi|125910|sp|P02754.3|LACB_BOVIN
--mkc111a1altcgaqalivtqtmkgldiqkvAGTWYSLAMAAsdis11DAQSAPLRVYVEELKPTPEGDLEILLQKW-
-----ENGCEAQQKIIAEKTKIPAVFKID---ALNENKVLVLDTDYKKYLLFCMENSAEPe-QSLACQCLVRTPEVDD
EAlekFDKALKALPMHIRLSFNPTQLEeqchi-----
>gi|1346407|sp|P04119.4|LACB_PIG
mrc111tlglallcgvqavevtpimteltdqkvAGTWHTVAMAVsdvs11DAKSSPLKAYVEGLKPTPEGDLEILLQKR-
-----ENDKCAQEVLLAKKTDIPAVFKIN---ALDENQLFLDQDYSHLLLCMENSApE-HSLVCQSLARTLEVDD
QIrekFEDALKTLVPMRIL--PAQLEeqcrv-----
>gi|75058963|sp|O97877|O97877_FELCA
-----dgldlqkvAGTWHSMAAAsdis11DSETAPLRVYVQELRPTPRDNLEIILRKW-
-----EHGRCVQKKVWAEKTELLAEFKIS---YLDENELIVLDTDYENYLFCELENTDAPg-QNLVCQCLTRTLKADN
EVmekFDRALQTLVQVRLFFDPTQVAeqcrv-----
>gi|462472|sp|P33685.1|LACB1_CANLF
-----ivvprtmedldlqkvAGTWHSMAAAsdis11DSETAPLRVYIQLRPTPDQNLIEIVLRKW-
-----EDGRCAEQKVLAEKTEVPAEFKIN---YVEENQIFLDQDYDNLYFFCEMNADApQ-QSLMCQCLARTLEVDD
EVmekFNRALKTLVPMQ-LLNPTQAEeqcli-----
>gi|75058766|sp|O77511|O77511_PAPCY
mqc111tlgvalicgvwainspqtmqdvlpk1AGTWHSMAAAsdfs11ETKEAPLRIYISSLQPTPEGNLEIVLRRWs
qkqspfrDSNQCIEEKIIAEKTENPIEFKIN---YLDENRIYLFNTDGSKYFLCCESTRR---QNLACQYLARTLEVDD
KVmaeFISFLKTLVPMQIFLDMTQAEeqcrv-----
>gi|125913|sp|P13613.1|LACB1_EQUAS
-----tnipqtmqdlldlqevAGKWHSVAMAAsdis11DSEEAPLRVYIEKLRTPEDNLEIILREG-
-----ENKGCAEKKIFAECTESPAEFKIN---YLDQDQVFDSDYKNYLLFCMKNAATpG-QSLVCQYLARTQMWDE
EImekFRRALQPLPGRVQIVPDLTRMAercr-----
>gi|130701|sp|P09466.2|PAEP_HUMAN
mlc111tlgvalvcgvpmamdiqtkqdlpklAGTWHSMAAMATnnis1mATLKAPLRVHITSLLPTPEDNLEIVLHRW-
-----ENNSCVEKKVLGEKTENPKFKIN---YTVANEATLLDQDYDNFLFLCLQDTTTPi-QSMMCYLARVLVEDD
EImqgFIRAFRPLPRHLWYLLDLKQMEepcrf-----
>gi|90108547|pdb|1YUP|A
-----iivtqtmkdldvqkvAGTWYSLAMAAsdis11DAQSAPLRVYVEELKPTPGDLEILLQKW-
-----ENGKCAQKKIIAEKTEIPAVFKIDA---LNENKVLVLDTDYKKYLLFCMENSAE-PeQSLACQCLVRTPEVDD
EAlekFDKALKALPMHIRLSFNPTQLEeqcrv-----
>gi|731187933|pdb|4R0B|A
-----dipatkqdlpklAGTWHSMAAMATneis1mATLKAPLRVHITSLLPTPEDNLEIVLHRW-
-----ENNSCVEKKVLGEKTENPKFKINITYVA---NEATLLDQDYDNFLFLCLQDTTTPi-QSMMCYLARVLVEDD
EImqgFIRAFRPLPRHLWYLLDLKQMEepcrghhhhhh

```

Εικόνα Αποτελεσμάτων 13

```
sequence_original.fasta x
1 >gi|283806778|pdb|2WWP|A
2 -----mapeaqvsvqpnfqgdkfLGRWFSAGLASnss-wlREKKAALSMCKSVVAPATDGGGLNLTSTFL-
3 -----RKNQCETRTMLLQPAAGSLGSSYSRSPHWGSTYSVSVVETDQYALLYSQGSKGpGeDFRMTLYSRTQTTPRA
4 ELkekFTAFCQAQGFTEDTIVFLPQTDkcmteqahhhhhh
5 >gi|125910|sp|P02754.3|LACB_BOVIN
6 --mkclllalaltcgagaltvtgmkglldiqkvAGTWYSLAMAAsdsl1DAQSAPLRVYVEELKPTPEGDLLEILLQKW-
7 -----ENGECACQKKIIAEKTKIPAVFKID---ALNENKVLVLDTDYKYLFLCMENSAEpe-QSLACQCLVRTPEVDD
8 EALekFDKALKALPMHIRLSFNPTQLEeqchi-----
9 >gi|1346407|sp|P04119.4|LACB_PIG
10 mrclll1tlglallcgvqavevtptimteltdtqkvAGTWHTVAMAVsdvs11DAKSSPLKAYVEGLKPTPEGDLLEILLQKR-
11 -----ENDKCAQEVLLAKKTIDIPAVFKIN---ALDENQVFLLDTDYSHLLCMENSApe-HSLVCQSLARTLEVDD
12 QIrekFEDALKTILSVPMRIL--PAQLEeqcrv-----
13 >gi|75058963|sp|O97877|O97877_FELCA
14 -----dgldlqkvAGTWHSMAAAsdis11DSETAPLRVYVQELRPTPRDNLEIILRKW-
15 -----EHGRCVQKKVWAEKTELLAEFKIS---YLDENELIVLDTDYENYLFCELENTDApG-QNLVCQCLRTTLKADN
16 EVmekFDRALQTLFVDVRLFFDPTQVAeqcrv-----
17 >gi|462472|sp|P33685.1|LACB1_CANLF
18 -----ivvprtmedldlqkvAGTWHSMAAAsdis11DSETAPLRVYVQELRPTPDQNLLEIVLRKW-
19 -----EDGRCAEQKVLAEKTEVPAEFKIN---YVEENQIFLLDTDYDNFLFCLENTDApG-QSLVCQCLARTLEVDD
20 EVmekFNRALKTLFVHMQ-LLNPTQAEeqcli-----
21 >gi|75058766|sp|O77511|O77511_PAPCY
22 mqclll1tlgvalicgvwainspqtmqdvlpklAGTWHSMAAAsdfs11ETKEAPLRIYISSLQPTPEGNLEIVLRW-
23 qkqspfrDSNQCIEEKIIAEKTENPIEFKIN---YLDENRIYLFNTDGSKYFLCLESTR---QNLACQYLARTLEVDD
24 KVmaeFISFLKTLFVHMQIFLDMTQAEeqcrv-----
25 >gi|125913|sp|P13613.1|LACB1_EQUAS
26 -----tnipqtmqdlldlqevAGKWHVMAAAsdis11DSEAPLRVYIEKLRTPEDNLEIILREG-
27 -----ENKGCACQKKIIAEKTESPAEFKIN---YLDENLIVLDSDYKYLFLCMKNAATpG-QSLVCQYLARTQMVDE
28 EImekFRRALQPLPGRVQIVPDLTRMAercr-----
29 >gi|130701|sp|P09466.2|PAEP_HUMAN
30 mlclll1tlgvalvcgvpamdipgtkqdlelpklAGTWHSMAATnnislATLKAPLRVHITSLLEPTPEDNLEIVLHRW-
31 -----ENNSCQVEKKVLGEKTENPKFKIN---YTVANEATLLDTDYDNFLFLCLQDITTTpI-QSMQCQYLARVLVEDD
32 EImqgFIRAFRPLRHLWYLLDLKQMEepcrf-----
33 >gi|90108547|pdb|1YUP|A
34 -----iivtqtmkdldvqkvAGTWYSLAMAAsdsl1DAQSAPLRVYVEELKPTPGGDLLEILLQKW-
35 -----ENGKCAQKKIIAEKTEIPAVFKIDA---LNENKVLVLDTDYKYLFLCMENSAE-PeQSLACQCLVRTPEVDD
36 EAmekFDKALKALPMHIRLSFNPTQLEeqcrv-----
37 >gi|731187933|pdb|4R0B|A
38 -----dipgtkqdlelpklAGTWHSMAATneislATLKAPLRVHITSLLEPTPEDNLEIVLHRW-
39 -----ENNSCQVEKKVLGEKTENPKFKINITYVA---NEATLLDTDYDNFLFLCLQDITTT-PiQSMQCQYLARVLVEDD
40 EImqgFIRAFRPLRHLWYLLDLKQMEepcrghhhhhh
```

Εικόνα Αποτελεσμάτων 14

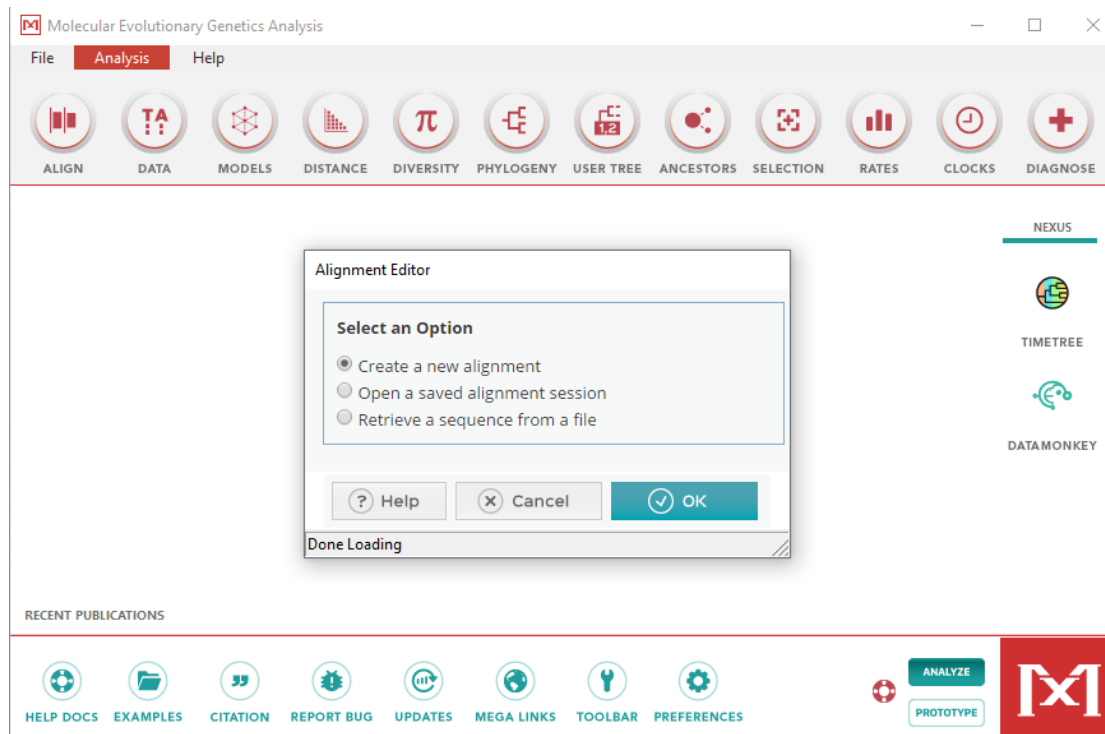
3.4. Ζητούμενο 4

Στο 4^ο ζητούμενο, αρχικά εισάγουμε το αρχείο στο λογισμικό του MEGA (με το τρόπο που μας συστήνει η εικόνα 7.9).

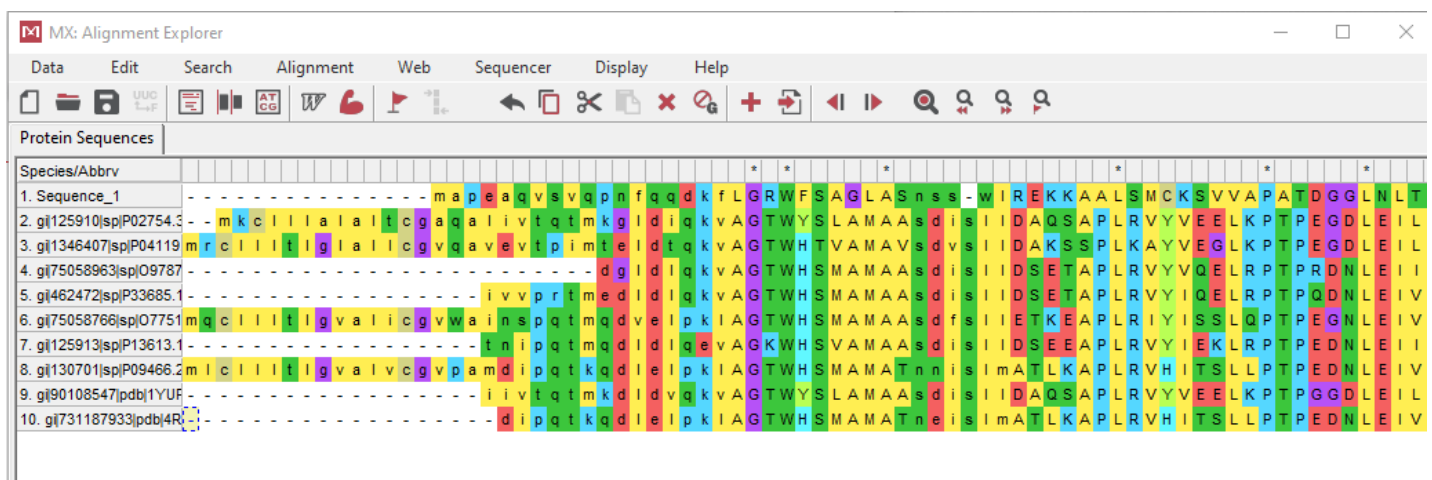
Στη συνέχεια, στοιχίζουμε τις ακολουθίες και τις αποθηκεύουμε σε αρχεία της μορφής .mas και .meg (*sequence.mas* & *sequence.meg*), αφού πρώτα έχουμε ονοματίσει κατάλληλα τις ακολουθίες.

Επιπλέον, ξανά αποθηκεύουμε το αρχείο .fasta, ώστε να είναι πλέον στοιχισμένο, με όνομα *sequence.fas*.

Θα ακολουθήσουν ενδεικτικές εικόνες από τη διαδικασία και τα παραπάνω αρχεία.



Εικόνα Αποτελεσμάτων 15



Εικόνα Αποτελεσμάτων 16



Εικόνα Αποτελεσμάτων 17

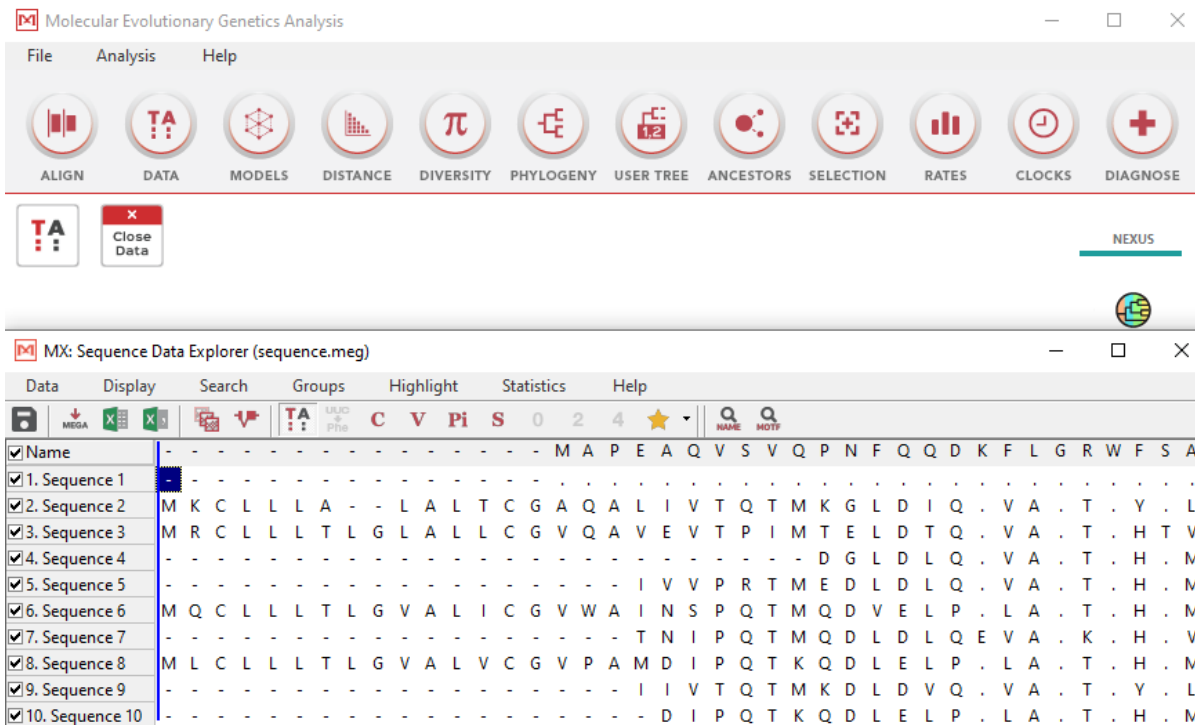
sequence_original.fasta sequence.fasta

```
1 >Sequence_1
2 -----MAPEAQVSVQPNFQDKFLGRWFSAGLASNS-SWLREKKAALSCKSVVAPATDGGNLSTFL-----RKNQCETRIMLLQPAGSLGYSYSRSPHWGSTYSVSUVVETDYDQYALLYSQSGKGPEDFRMATLYSRTQTPRAELKEKFTAFCKAAGFTEDTIVFL
3 >Sequence_2
4 MKCLLLA--LALTQGAQALIVTQTMKGLDIOQKAVGTWYSLAMAAASDISLLDAQSAPLVRVVEELKPTPEGDEILLQKN-----ENGKCAQKKIIAETKIIPAVFKI---DALNENKVLVLDTDYKKYLLFCMENSAPPEQSL-ACQCLVRTPEVDDAEKFKALKALPMHIRLSFNP
5 >Sequence_3
6 MKCLLTGLGALLQGVQVAEVPITMELTDIOKAVGHTWTVAMAVSDVSLDAKSSPLKAVYEGLKPTPEGDEILLQKR-----ENDKCAQEVLLAKKTDIIPAVFKI---NALDENQLFLDITDYDSHLLFCMENSAPPEHSL-VQCSLARTLEVDDQIREKFDALKTSLVPMHRI--P
7 >Sequence_4
8 -----DGLDLQKQVATGWSMAMAASDISLLDSEAPLVRVYQELRPTPRNLEIILRW-----EHRGCVQKKVMAEKTELLAEFKI---SYLDENELIVLDDIYENLYFFCLENIDAPQQNL-VQCLTRTLKADNEVMEKFRALQTLPVVRLFFDP
9 >Sequence_5
10 -----IIVPRTMEDLQKQVATGWSMAMAASDISLLDSEAPLVRVYQELRPTPQNLEIVLRW-----EDGRCAEQKVLAEKTEVPAEFKI---NYVEENQIFLLDITDYDNYLFFCEMNADAPQSL-MQCLARTLEVDDNEVMEKFRALQTLVPMHRI--NP
11 >Sequence_6
12 MQCLLTGLGALLQGVWAINSPQTMQDVELPKLAGTWSMAMAASDFSLETKEAPLRIYISSLQPTPEGNLEIVLRWSQKQSPFRDSSNQICEEKIIAETENPIEFKI---NYLDENRIYLFNTDGSKYFLCLESTR---RQNL-ACQYLARTLEVDDKVMAEFISFLKTLVPMHRIQIFLDM
13 >Sequence_7
14 -----TNIPTQMQLDLQEVAGKWSVAMAASDISLLDSEAPLVRVYIEKLAPTPEDNLEIILREG-----ENKGCAGEKKIIPAEKTESPAEFKI---NYLDEDVTFALDSDYKNYFLCMKNAATPGQSL-VQCYLARTQMVDEIIMEKFRALQTLPGRVQIVPDL
15 >Sequence_8
16 MLCLLTGLGALLQGVQVPMADIPTQKQDELPKLAGTWSMAMATNNISMATLKAPLRVHTISLLTPEDNLEIVLRW-----ENNSCEKKVLGEKTENPKKFKI---NYTVANEATLLDITDYNFLFCLQDITPTIQSM-MQCYLARVLVEDDEIMQGFIRAFRLPAHLWYLLD
17 >Sequence_9
18 -----IIVTQTMKDLQKQVAGTWYSLAMAAASDISLLDAQSAPLVRVVEELKPTPGGDEILLQKN-----ENGKCAQKKIIAETKIIPAVFKI---DALNENKVLVLDTDYKKYLLFCMENSAPPEQSL-ACQCLVRTPEVDDAEKFKALKALPMHIRLSFNP
19 >Sequence_10
20 -----DIPQTKQDELPLKLAGTWSMAMATNEISMATLKAPLRVHTISLLTPEDNLEIVLRW-----ENNSCEKKVLGEKTENPKKFKI---NYTVANEATLLDITDYNFLFCLQDITPTIQSM-MQCYLARVLVEDDEIMQGFIRAFRLPAHLWYLLD
```

Εικόνα Αποτελεσμάτων 18

[illegible]

Εικόνα Αποτελεσμάτων 19



Εικόνα Αποτελεσμάτων 20

```
sequence_original.fasta x sequence.fas x sequence.mas x sequence.meg x
1 #mega
2 !Title sequence;
3 !Format DataType=Protein indel=-;
4
5 #Sequence_1
6 -----MAPEAQVSVQPNFQDDKFLGRWFSAGLASNS-SWLREKKAALSMC
7 KSVVAPATDGGNLNLTSTFL-----RKNQCETRTMLLPAGSLGSSYSRSPHWGSTYS
8 VSVVETDQYALLYSQSGKPGEDFRMATLYSRTQTPRAELKEKFTAFCKAQGFTEDTI
9 VFLPQTDKCMTEQAHHHHHH
10
11 #Sequence_2
12 MKCLLLA--LALTCGAQALIVTQTMKGLDIQKVAGTWYSLAMAASDISLLDAQSAPLRVY
13 VEELKPTPEGDLLEILLQKW-----ENGECQKKIIAEKTIKIPAVFKI---DALNENK
14 VLVLDTDYKYLFCMENSAPPEQSL-ACQCLVRTPEVDDEALEKFDKALKALPMHIRLS
15 FNPTQLEEQCHI-----
16
17 #Sequence_3
18 MRCLLLTLGLALLCGVQAVEVTPIMTELDQKVAGTWHTVAMAVSDVSLDAKSSPLKAY
19 VEGLKPTPEGDLLEILLQKR-----ENDKCAQEVLLAKKTDIPAVFKI---NALDENQ
20 LFLLDTDYDNYLFFCEMNASPEHSL-VCQSLARTLEVDDQIREKFEDALKTLSPVPMRIL
21 --PAQLEEQCRV-----
22
23 #Sequence_4
24 -----DGLDLQKVAGTWHSMAAASDISLLDSETAPLRVY
25 VQELRPTPRDNLEIILRKW-----EHGRCVQKKVWAEKTELLAEFKI---SYLDENE
26 LIVLDTDYENYLFCELENTDAPGQNL-VCQCLRTLKADNEVMEKFDRLQTLFPVDVRLF
27 FDPTQVAEQCRV-----
28
29 #Sequence_5
30 -----IVVPRTMEDLDLQKVAGTWHSMAAASDISLLDSETAPLRVY
31 IQELRPTPDNLEIVLRKW-----EDGRCAEQKVLAEKTEVPAEFKI---NYVEENQ
32 IFLLDTDYDNYLFFCEMNADAPQQL-MCQCLARTLEVDDNEVMEKFNRLKTLFPVHMQLL
33 -NPTQAEQCLI-----
34
35 #Sequence_6
36 MQCLLLTLGVALICGVWAINSPQTMQDVELPKLAGTWHSMAAASDFSLLKETKEAPLRIY
37 ISSLQPTPEGNLEIVLRRWSQKQSPFRDSNQCIEEKIIAEKTENPIEFKI---NYLDENR
38 IYLFNTDGSKYLFLCLESTR--RQNL-ACQYLARTLEVDDKVMAEFISFLKTLFPVHMQIF
39 LDMTQAEQCRV-----
```

Εικόνα Αποτελεσμάτων 21

3.5. Ζητούμενο 5

Στο παρόν ζητούμενο και αφού ακολουθήσουμε τις οδηγίες της άσκησης, δημιουργούμε δέντρα με τις μεθόδους ένωσης γειτόνων (Neighbor Joining Method) και μέγιστης πιθανοφάνειας (Maximum Likelihood Method).

Τα αρχεία των δένδρων έχουν αποθηκευτεί στον κατάλογο των αρχείων της άσκησης 7.2 .

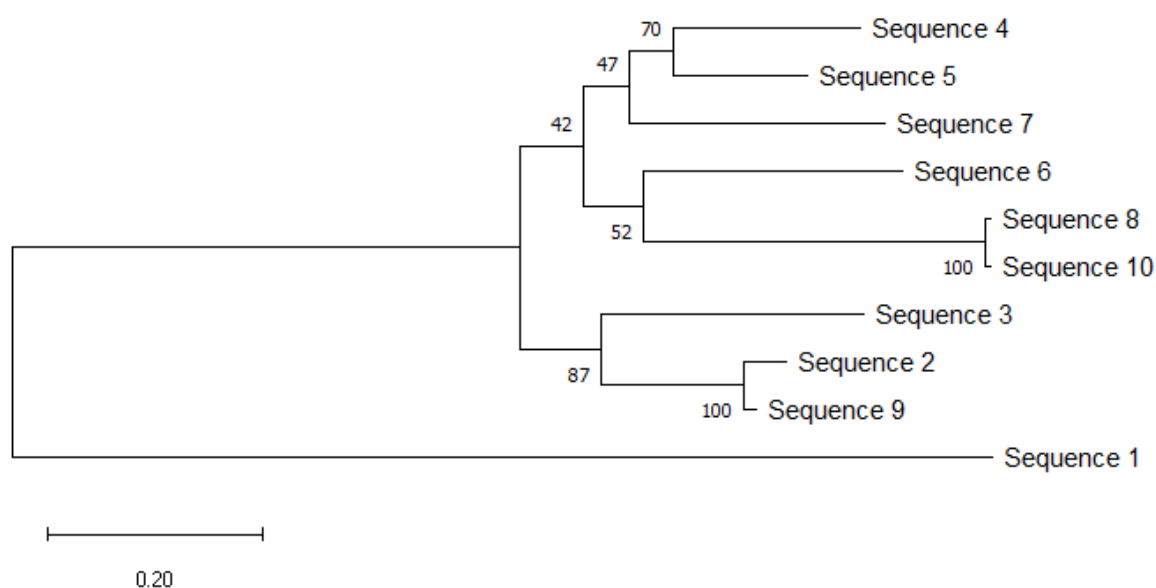
Ακολουθούν οι εικόνες – αποτελέσματα από την προαναφερθέντα διαδικασία:

MX: Analysis Preferences

Phylogeny Reconstruction

Option	Setting
ANALYSIS	
Scope →	All Selected Taxa
Statistical Method →	Neighbor-joining
PHYLOGENY TEST	
Test of Phylogeny →	Bootstrap method
No. of Bootstrap Replications →	500
SUBSTITUTION MODEL	
Substitutions Type →	Amino acid
Model/Method →	Poisson model
RATES AND PATTERNS	
Rates among Sites →	Uniform Rates
Gamma Parameter →	Not Applicable
Pattern among Lineages →	Same (Homogeneous)
DATA SUBSET TO USE	
Gaps/Missing Data Treatment →	Complete deletion
Site Coverage Cutoff (%) →	Not Applicable
SYSTEM RESOURCE USAGE	
Number of Threads →	7

Εικόνα Αποτελεσμάτων 22 : Neighbor Joining Method Tree Settings.

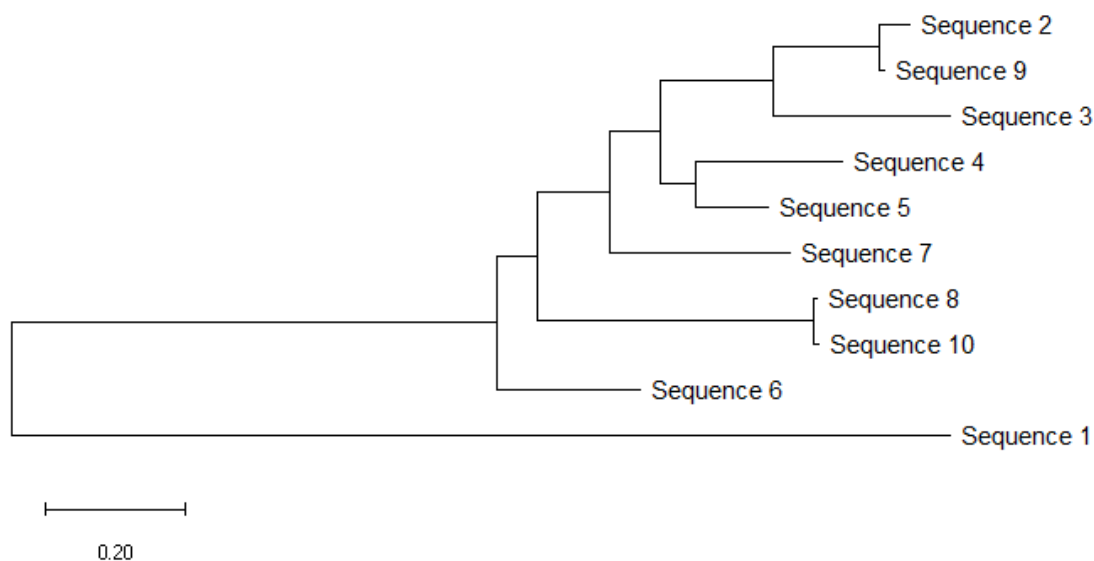


Εικόνα Αποτελεσμάτων 23 : Neighbor Joining Method Tree.

MX: Analysis Preferences

Option	Setting
ANALYSIS	
Statistical Method →	Maximum Likelihood
PHYLOGENY TEST	
Test of Phylogeny →	None
No. of Bootstrap Replications →	Not Applicable
SUBSTITUTION MODEL	
Substitutions Type →	Amino acid
Model/Method →	Jones-Taylor-Thornton (JTT) model
RATES AND PATTERNS	
Rates among Sites →	Uniform Rates
No of Discrete Gamma Categories →	Not Applicable
DATA SUBSET TO USE	
Gaps/Missing Data Treatment →	Complete deletion
Site Coverage Cutoff (%) →	Not Applicable
TREE INFERENCE OPTIONS	
ML Heuristic Method →	Nearest-Neighbor-Interchange (NNI)
Initial Tree for ML →	Make initial tree automatically (Default - NJ/BioNJ)
Initial Tree File →	Not Applicable
Branch Swap Filter →	None
SYSTEM RESOURCE USAGE	
Number of Threads →	7

Εικόνα Αποτελεσμάτων 24 : Maximum Likelihood Method Tree Settings



Εικόνα Αποτελεσμάτων 25 : Maximum Likelihood Method Tree

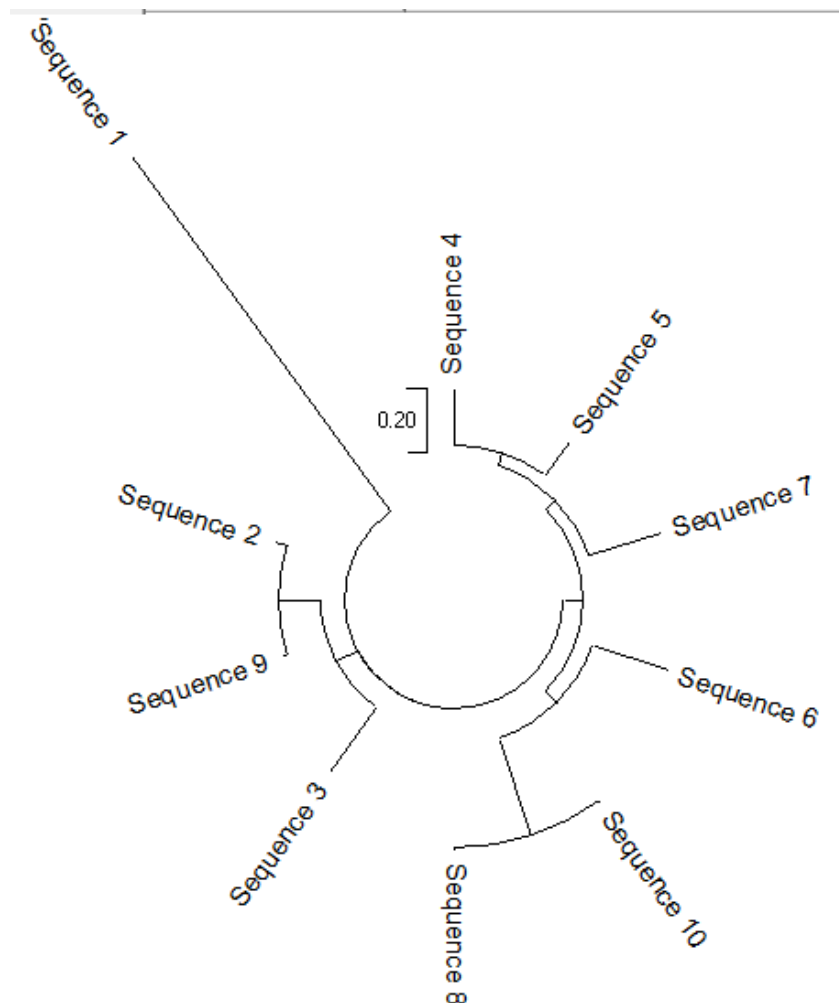
3.6. Ζητούμενο 6

Στο 6^ο ζητούμενο της άσκησης 7.2 μας ζητείτε να πειραματιστούμε με τα εργαλεία του λογισμικού που σχετίζονται με τα δέντρα που δημιουργήσαμε στο προηγούμενο ερώτημα, τροποποιώντας τα, αφού πρώτα έχουμε διαβάσει την εκάστοτε λεζάντα.

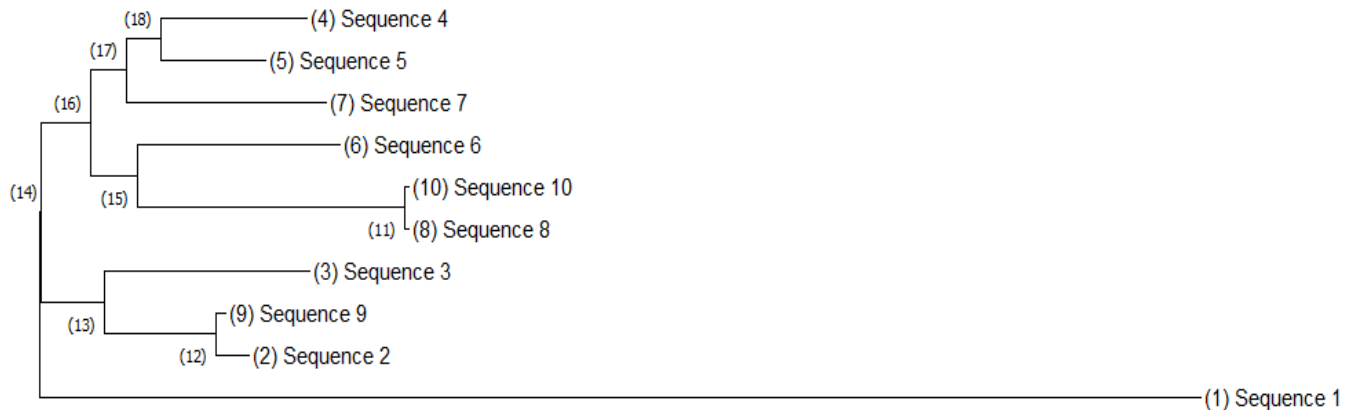
Παρακάτω απεικονίζονται οι τροποποιήσεις που υλοποιήσαμε στο δέντρο ένωσης γειτόνων (Neighbor Joining Method), με το ίδιο τρόπο γίνονται για οποιαδήποτε μέθοδο.

Ειδικότερα, προσθέσαμε μια ρίζα, κάναμε swap τους κόμβους 8-10 και 9-2, αποκρύψαμε τα μήκη των κλάδων και πειραματιστήκαμε με τις μορφές απεικόνισης.

Ακολουθούν εικόνες από τα τελικά αποτελέσματα σε κανονική και κυκλική μορφή απεικόνισης.



Εικόνα Αποτελεσμάτων 26 : Neighbor Joining Method Modified Circle Tree .



Εικόνα Αποτελεσμάτων 27 : Neighbor Joining Method Modified Regular Tree .

3.7. Ζητούμενο 7

Τέλος, στο 7^ο και τελευταίο ζητούμενο της άσκησης θα πραγματοποιήσουμε τη διαδικασία του bootstrapping και θα αιτιολογήσουμε τη νοηματική υπόσταση για τις συστάδες κλάδων που έχουν χαμηλά επίπεδα στήριξης.

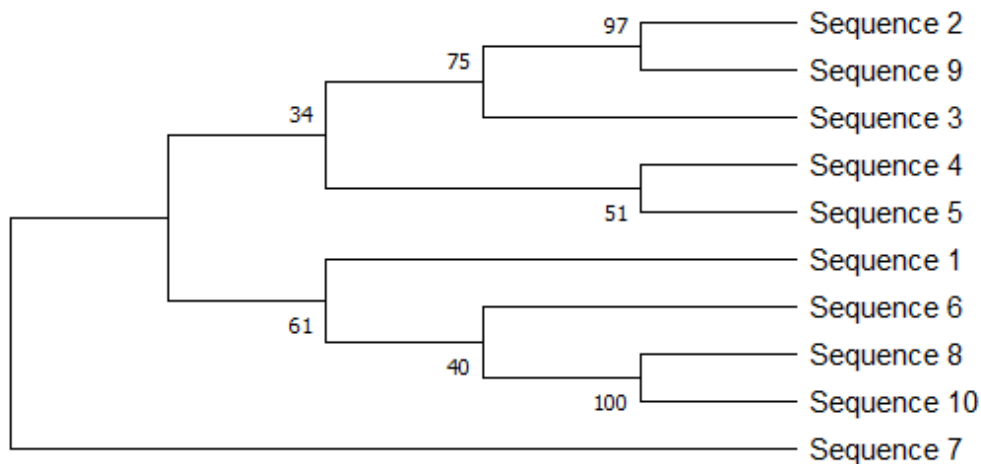
Αρχικά, με τη μέθοδο bootstrapping εκτελέσαμε 500 δοκιμές bootstrap (βλέπε εικόνα 29, παρακάτω) για να υπολογίσουμε με όσο το δυνατόν μεγαλύτερη ακρίβεια το ποσοστό των επαναλήψεων στα οποία τα δέντρα bootstrap υποστηρίζουν κάθε συστάδα κλάδων του υπό αξιολόγηση δένδρου.

Ειδικότερα, οι ακολουθίες 8 και 10 (Sequence 8 και Sequence 10), στο 100% των δοκιμών bootstrap τοποθετήθηκαν στην ίδια συστάδα κλάδων. Με την ίδια λογική η συστάδα κλάδων που περιέχει τις ακολουθίες 2, 9, 3 έχει ποσοστό εμφάνισης 75% με την υπό συστάδα των ακολουθιών 2 και 9 να εμφανίζεται στο 97% των επαναλήψεων και μόλις το 3% των δοκιμών αυτή η ομάδα πρωτεϊνών περιέχει κάποια άλλη ακολουθία (κυρίως την ακολουθία 3).

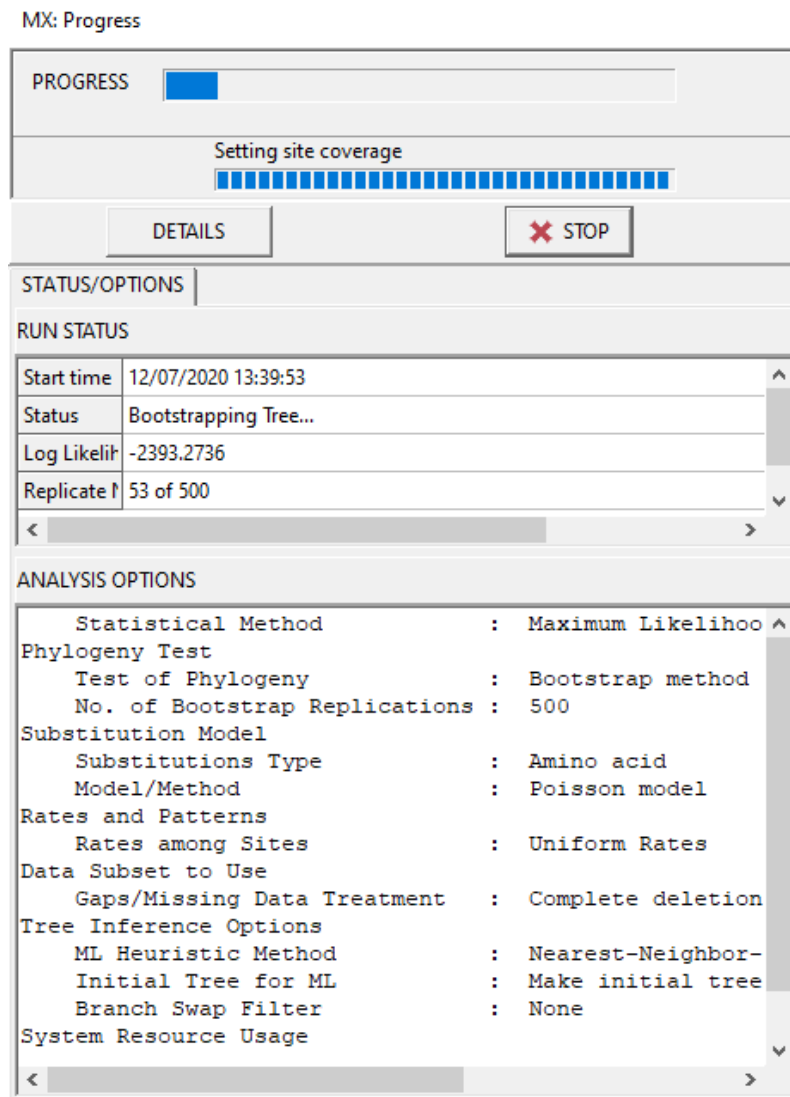
Τέλος, παρατηρούμε ότι η συστάδα κλάδων των ακολουθιών 6, 8 και 10 εμφανίζεται μόνο στο 40% των δοκιμαστικών δέντρων (ενώ η επιμέρους υπό συστάδα των 8 και 10 στο 100%). Επομένως, μπορούμε να συμπεράνουμε ότι δεν υπάρχει ισχυρή υποστήριξη για μια συγγενική ομάδα των ακολουθιών 8 και 10 που είχε έναν κοινό πρόγονο με την ακολουθία 6.

Συμπεραίνουμε ότι, η μέθοδος bootstrapping, μέσω της επαναληπτικής διαδικασίας δοκιμών του συνόλου των δεδομένων, μας παρέχει – με έναν

αρκετά γρήγορο τρόπο – ένα μέτρο αξιολόγησης του πόσο καλή είναι μία τοπολογία δέντρου με βάση την ολότητα των δεδομένων.



Εικόνα Αποτελεσμάτων 28 : Neighbor Joining Method Bootstrap Consensus Tree .



Εικόνα Αποτελεσμάτων 29 : Neighbor Joining Method Bootstrap Consensus Process .

Άσκηση 11.4

1. Εκφώνηση

Στο σχήμα 11.7 φαίνεται ένα HMM με δύο καταστάσεις α και β . Όταν το HMM βρίσκεται στην κατάσταση α , έχει την μεγαλύτερη πιθανότητα να εκπέμψει πουρίνες (A και G). Όταν βρίσκεται στην κατάσταση β , έχει μεγαλύτερη πιθανότητα να εκπέμψει πυριμιδίνες (C και T). Αποκωδικοποιήστε την πιο πιθανή ακολουθία των καταστάσεων (α/β) για την αλληλουχία GGCT. Χρησιμοποιήστε λογαριθμικές βαθμολογίες για κανονικές βαθμολογίες πιθανοτήτων.

2. Κεντρική Ιδέα Υλοποίησης

Το πρόγραμμα μας αποτελείται από ένα μοναδικό αρχείο το οποίο πρέπει και να εκτελέσουμε ώστε να δούμε τα αποτελέσματα.

Σε αυτήν την άσκηση προσπαθούμε να ανακαλύψουμε την ‘καλύτερη’ διαδρομή (αυτή με το μεγαλύτερο σκορ σαν σύνολο) για την κωδικοποίηση της ακολουθίας νουκλεοτιδίων “GGCT”.

2.1. Δηλώσεις Μεταβλητών

Θεωρούμε ισοπίθανες αρχικές πιθανότητες να ξεκινάει είτε από την κατάσταση A είτε από την B (0.5 για κάθε μια) εφόσον το πρόγραμμα δεν μας υποδεικνύει κάτι διαφορετικό.

Έπειτα δημιουργήσαμε έναν πίνακα που αναφέρεται στην πιθανότητα να πηγαίνει από την μία κατάσταση σε άλλη ή ίδια να πηγαίνει στον εαυτό της.

Η ακολουθία GGCT μετατρέπεται σε αριθμούς μέσω της εντολής nt2int για να μπορεί να γίνει αντιληπτή. Επομένως μετατρέπεται σε ακολουθία αριθμών (3,3,2,4).

2.2. Αλγόριθμος Viterbi

Η υλοποίηση του αλγόριθμου Viterbi σε αυτήν την άσκηση έγινε ως εξής: Αρχικά, μετατρέπουμε όλες τις τιμές των πινάκων πιθανοτήτων σε λογαριθμικές τιμές για να μην υπάρχει πρόβλημα - underflow με τις κανονικές πιθανότητες που θα υπολογίσουμε (προτείνεται από την άσκηση).

Μετά ξεκινάμε βάζοντας στην αρχή του πίνακα Viterbi τις αρχικές πιθανότητες των καταστάσεων μας (50% - 50% για κάθε κατάσταση) και συνεχίζουμε για κάθε σύμβολο της ακολουθίας τον υπολογισμό της πιθανότητας να εκπεμφθεί δοσμένης της τωρινής κατάστασης και της πιθανότητας της προηγούμενης κάθε φορά.

Αφού υπολογίσουμε ολόκληρο το πίνακα Viterbi και βρούμε την μέγιστη πιθανότητα στο τέλος ανάμεσα στις δύο καταστάσεις ξεκινάμε τη διαδικασία του backtracking.

2.3. Κύριο Μέρος

Αρχικά, δημιουργούμε έναν πίνακα δύο διαστάσεων ο οποίος αποτελείται μόνο από μηδενικά. Έπειτα προσπαθούμε μέσω εμφολευμένων loops να ακολουθήσουμε όλες τις πιθανές διαδρομές που προκύπτουν από το διάγραμμα που κατασκευάζουμε στο χαρτί.

Πιο συγκεκριμένα, κοιτάμε κάθε φορά ποιο στοιχείο εξετάζουμε (πρώτο loop) και στην συνέχεια ελέγχουμε σε ποια κατάσταση βρισκόμαστε και έπειτα τα αποθηκεύουμε στον πίνακα που υπήρχαν τα μηδενικά ο οποίος χρησιμεύει σαν ένας λευκός πίνακας πάνω στον οποίο γράφουμε τις αντλούμενες πληροφορίες.

2.4. Τελικό Στάδιο

Στο τελικό στάδιο της εφαρμογής, υπολογίζεται το συνολικό σκορ των δύο διαδρομών (στο συγκεκριμένο πρόβλημα, σε κάποιο άλλο θα μπορούσε να έχει παραπάνω καταστάσεις άρα και παραπάνω διαδρομές), συγκρίνονται και αποφασίζεται ποια διαδρομή είναι η καλύτερη.

3. Αποτελέσματα

Όπως φαίνεται και στην παραπάνω φωτογραφία συγκρίνονται στο τέλος οι τιμές του πίνακα (το 1ο και το 2ο στοιχείο της 5ης στήλης) και επιλέγεται η ο μεγαλύτερος ο αριθμός δηλαδή η διαδρομή με το καλύτερο σκορ.

Άρα επιλέγουμε την διαδρομή 'bbbb' αφού είναι αυτή που μας δίνει το καλύτερο αποτέλεσμα με βάση τον αλγόριθμο Viterbi.

Trellis' Diagram:

State b:

-3.4739 -5.9479 -7.8368 -9.7258

State a:

-2.4739 -3.9479 -7.4218 -10.8957

Best score for the sequence 'GGCT' is: -9.7258

Best path for the sequence 'GGCT' is: bbbb

Εικόνα Αποτελεσμάτων 30

Συνεπώς, και όπως προαναφέρθηκε βλέπουμε ότι για την ακολουθία GGCT η καλύτερη ακολουθία καταστάσεων που δίνει μέγιστη πιθανότητα (λογαριθμική) -9.7258 είναι η BBBB, δηλαδή μόνο η δεύτερη κατάσταση.

Ενδεικτικά, εάν το λύναμε το πρόβλημα χωρίς να άγουμε τις πιθανότητες στη λογαριθμική κλίμακα, τότε το αποτέλεσμα του διαγράμματος Trellis, θα ήταν το παρακάτω:

Κατάσταση B	0,1 -->	0,018 -->	0,00486 -->	0,0013122
Κατάσταση A	0,2 -->	0,072 -->	0,00648 -->	0,0005832
	G	G	C	T

Απαραίτητα Εργαλεία

Η εργασία μας υλοποιήθηκε στο προγραμματιστικό-εκπαιδευτικό εργαλείο Matlab και του λογισμικού Mega.

Απαραίτητο εργαλείο-βιβλιοθήκη αποτελεί το Bionformantics tool που πρέπει να είναι και αυτό εγκατεστημένο για την υλοποίηση της. Κάνουμε ιδιαίτερη αναφορά σε αυτήν την βιβλιοθήκη επειδή ο χρήστης θα πρέπει να φροντίσει να την επιλέξει κατά την εγκατάσταση του Matlab μέσω installer ή σε μεταγενέστερη τροποποίησης της εγκατάστασης.

Το εργαλείο Mega είναι πιο απλό και απλά χρειάζεται να το εγκαταστήσουμε στον υπολογιστή μας από την επίσημη ιστοσελίδα του.

Βιβλιογραφία

1. Βιοπληροφορική και Λειτουργική Γονιδιωματική», Τρίτη Έκδοση, Jonathan Pevsner.
2. Σημειώσεις μαθήματος «Βιοπληροφορική», Διδάσκων Άγγελος Πικράκης, Ph.D.

Περιεχόμενα Απεσταλμένου Αρχείου

Το τελικό αρχείο της εργασίας (Εργασία_Βιοπληροφορικής.zip) θα περιέχει τα παρακάτω:

1. Ένα φάκελο για κάθε θέμα της εργασίας που θα εμπεριέχει σε υποφακέλους τις υλοποιήσεις και όλα τα απαραίτητα αρχεία για το κάθε ζητούμενο ερώτημα.
2. Ένα αρχείο κειμένου με τα ονόματα και τους Αριθμούς Μητρώου των μελών της ομάδας (Μέλη.txt).
3. Το παρόν έγγραφο Εργασία Βιοπληροφορικής.pdf, το οποίο περιλαμβάνει όλα όσα ζητήθηκαν από την εκφώνηση της εργασίας.