ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ

Διδάσκοντες: Άγγελος Πικράκης, Ευαγγελία Χρυσίνα

- ΓΡΗΓΟΓΗ ΣΤΕΦΑΝΟΣ, Π18178
- ΚΑΛΟΓΗΡΟΥ ΣΤΥΛΙΑΝΗ, Π18181
- ΚΩΝΣΤΑΝΤΙΝΙΔΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ, Π18180

Ιούλης 2021

ΠΕΡΙΕΧΟΜΕΝΑ

Ζητούμενα Εργασίας:

- Θέμα 1:

  Άσκηση 7.2

  Βιβλίο “Βιοπληροφορική και Λειτουργική Γονιδιωματική"

- Θέμα 2:

  Άσκηση 11.4

  Βιβλίο "Εισαγωγή στους Αλγορίθμους Βιοπληροφορικής".

- Θέμα 3:

  Άσκηση 6.12

  Βιβλίο "Εισαγωγή στους Αλγορίθμους Βιοπληροφορικής"

- Θέμα 4:

  Η εκφώνηση βρίσκεται στις σελίδες 18 - 20

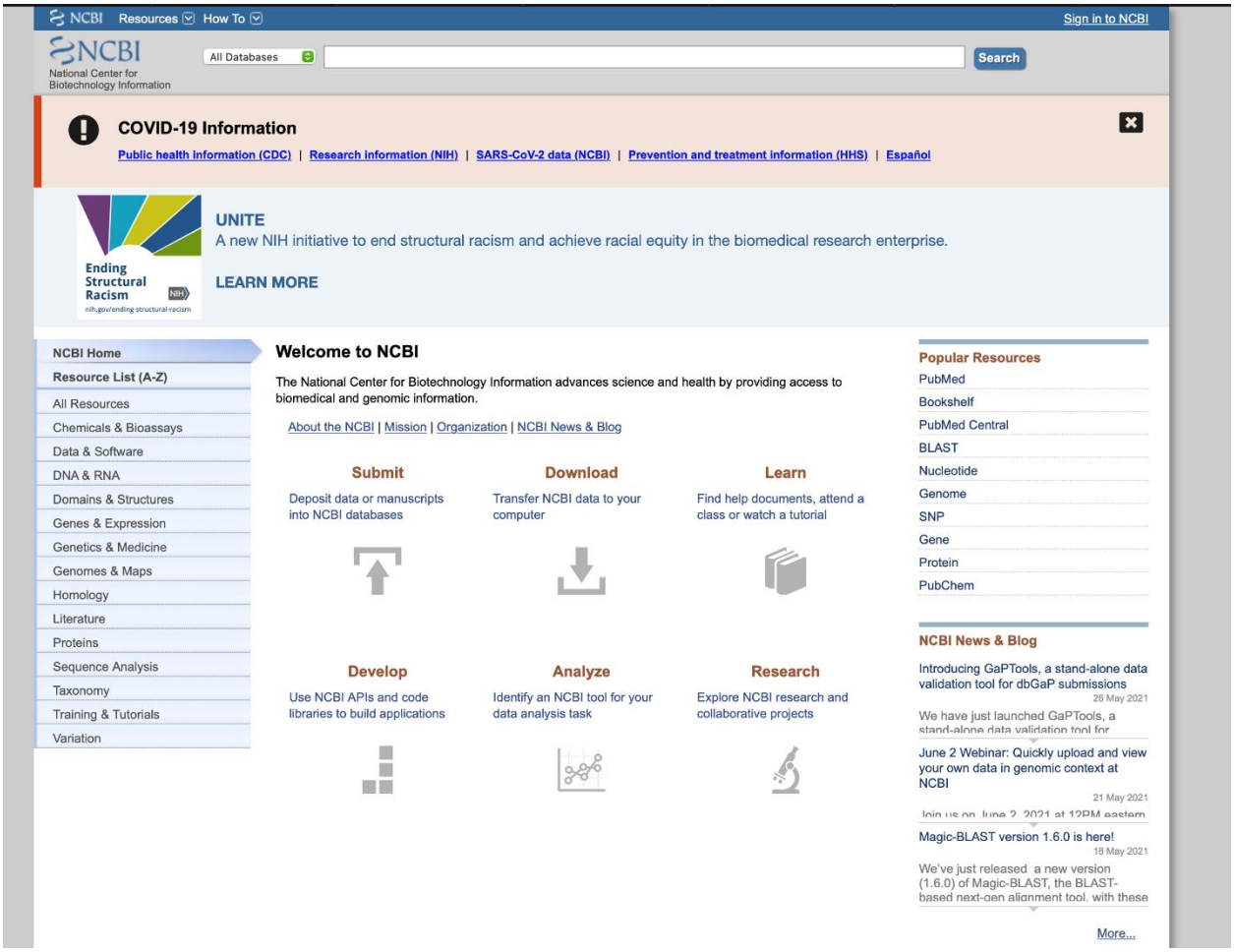
***Σημείωση**

Γλώσσα υλοποίησης των προγραμμάτων για τα θέματα 2 και 3 είναι η Python στο προγραμματιστικό περιβάλλον PyCharm. Το θέμα 1 με το λογισμικό Mega11 και το θέμα 4 με το ChimeraX1.2.5

Λύσεις:

➢ Θέμα 1:

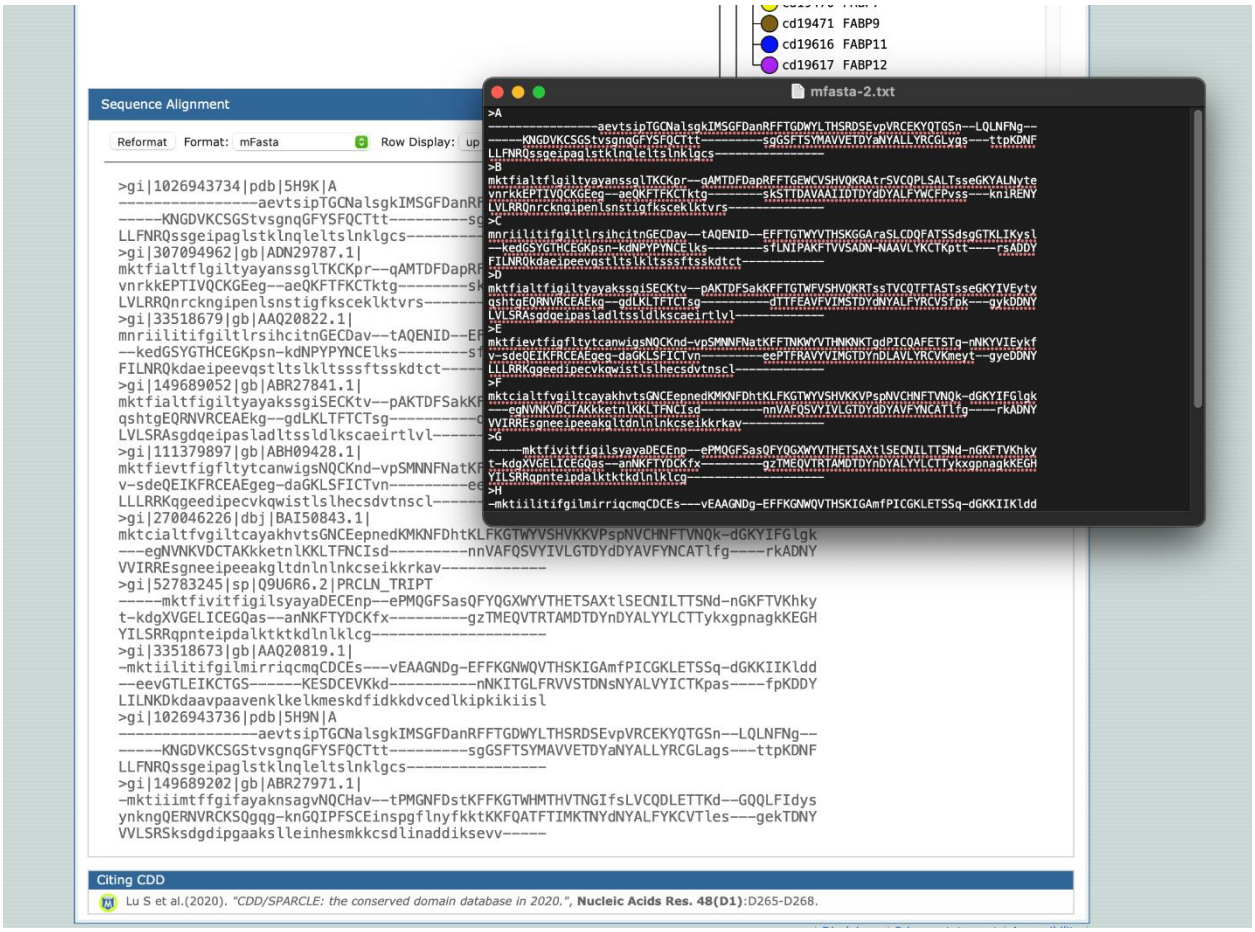
- 7.2.1



- 7.2.2

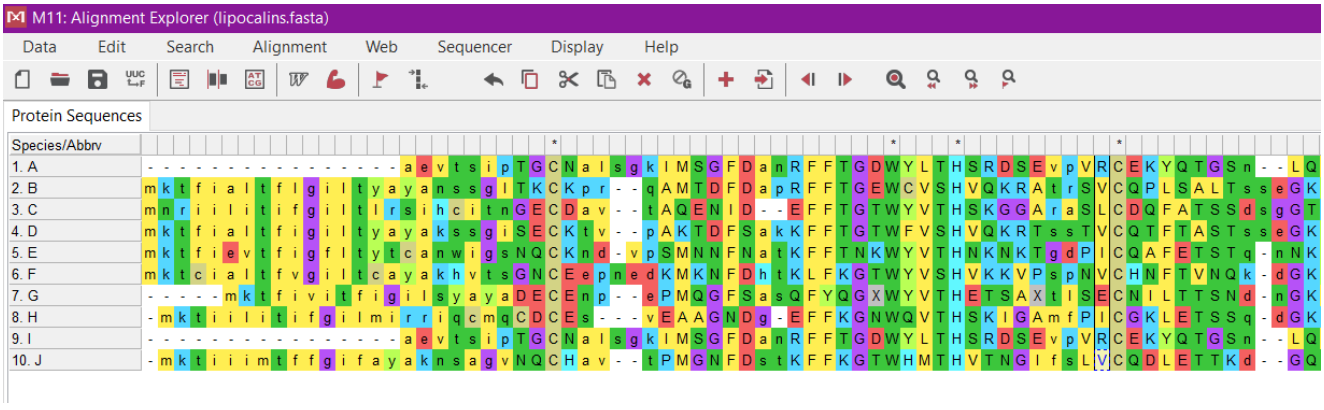

COVID-19 Information
Public health information (CDC) | Research information (NIH) | SARS-CoV-2 data (NCBI) | Prevention and treatment information (HHS) | Español
cd19423: lipocalin_LTBP1-like
Download alignment
Triatominae salivary lipocalins such as Rhodnius prolixus LTBP1 and Meccus pallidipennis triabin, and similar proteins
This subfamily includes various insect proteins found in the saliva of Triatominae (kissing bugs), including Rhodnius prolixus leukotriene-binding LTBP1. Rhodnius prolixus, a vector of the pathogen Trypanosoma cruzi, sequesters cysteinyl leukotrienes during feeding to inhibit immediate inflammatory responses; LTBP1 binds leukotrienes C4 (LTC4), D4 (LTD4), and E4 (LTE4). Meccus pallidipennis (syn Triatoma pallidipennis) triabin is a potent and selective thrombin inhibitor. It also includes Triatoma protracta procalin, a major salivary allergen which causes an allergic reaction in humans. It belongs to the lipocalin/cytosolic fatty-acid binding protein family which have a large beta-barrel ligand-binding cavity. Lipocalins are mainly low molecular weight extracellular proteins that bind principally small hydrophobic ligands, and form covalent or non-covalent complexes with soluble macromolecules, as well as membrane bound-receptors. They participate in processes such as ligand transport, modulation of cell growth and metabolism, regulation of immune response, smell reception, tissue development and animal behavior. Cytosolic fatty-acid binding proteins, also bind hydrophobic ligands in a non-covalent, reversible manner, and have been implicated in intracellular uptake, transport and storage of hydrophobic ligands, regulation of lipid metabolism and sequestration of excess toxic fatty acids, as well as in signaling, gene expression, inflammation, cell growth and proliferation, and cancer development.
Links
Source: cd00301
Taxonomy: Triatominae
PubMed: 16 links
Protein: Representatives
Specific Protein
Related Protein
Related Structure
Architectures
Superfamily: cl10502
Statistics
PSSM-Id: 381198
View PSSM: cd19423
Aligned: 47 rows
ThresholdBitScore: 101.663
ThresholdSettingGi: 33518669
Created: 1-Aug-2007
Updated: 2-Oct-2020
Structure
Structure View
Program: Cn3D
Drawing: All Atoms
Aligned Rows: up to 10
Download Cn3D
Conserved Features/Sites
PubMed References
ligand bind...
Feature 1: ligand binding cavity [chemical binding site]
Evidence:
Comment: hydrophobic cavity binds different hydrophobic ligands; ligands are bound within the beta-barrel in a central internal water-filled cavity lined with polar and hydrophobic amino acids
Citation: PMID 12222958
Comment: based on Triatominae salivary lipocalins and other lipocalin/cytosolic fatty-acid binding protein family members with structure
Citation: PMID 15642259
Structure: 5H9N; Rhodnius prolixus LTBP1 with bound leukotriene C4, contacts 4A
- View structure with Cn3D
Citation: PMID 27124118
Download Cn3D for Viewing 3D Structure
Scroll to Sequence Alignment Display
cd19423 is part of a hierarchy of related CD models.
Use the graphical representation to navigate this hierarchy.
cd19423 is a member of the superfamily cl10502
cd19423 Sequence Cluster
Zoom In
Sub-family Hierarchy
Interactive Display with CDTree
cd00301 lipocalin_FABP
cd00742 FABP
cd19441 CRABP
cd19460 CRABP1
cd19461 CRABP2
cd19442 CRBP

- 7.2.3

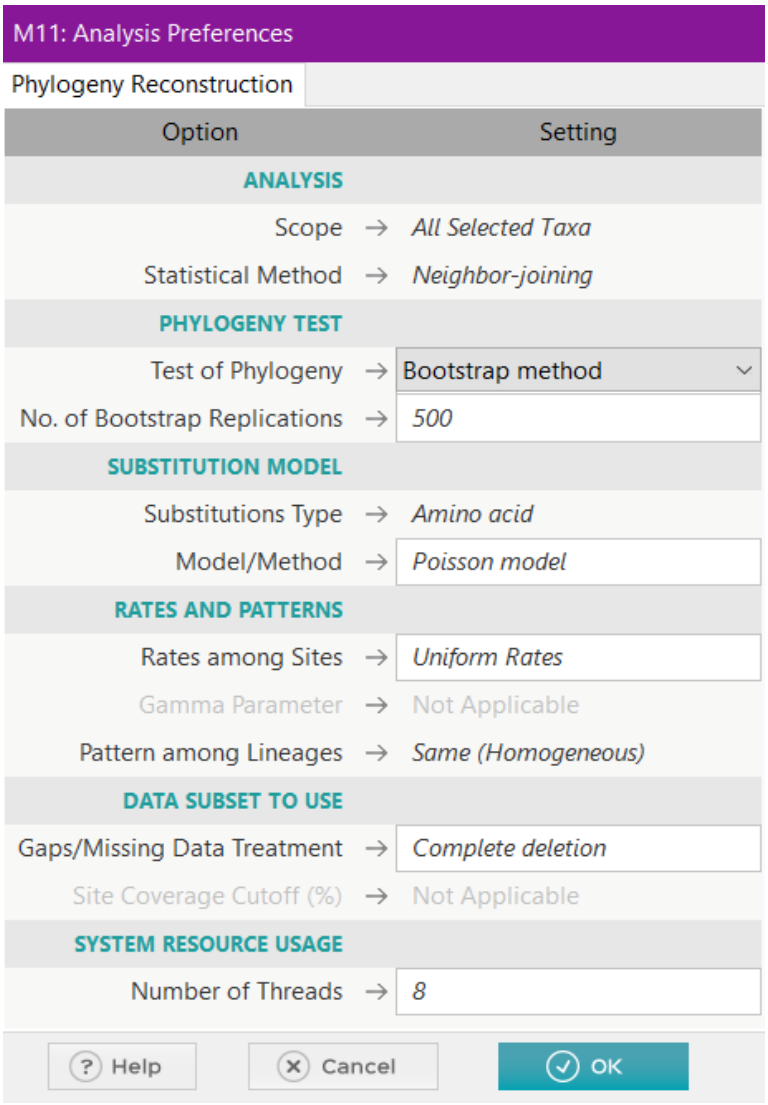


- 7.2.4

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

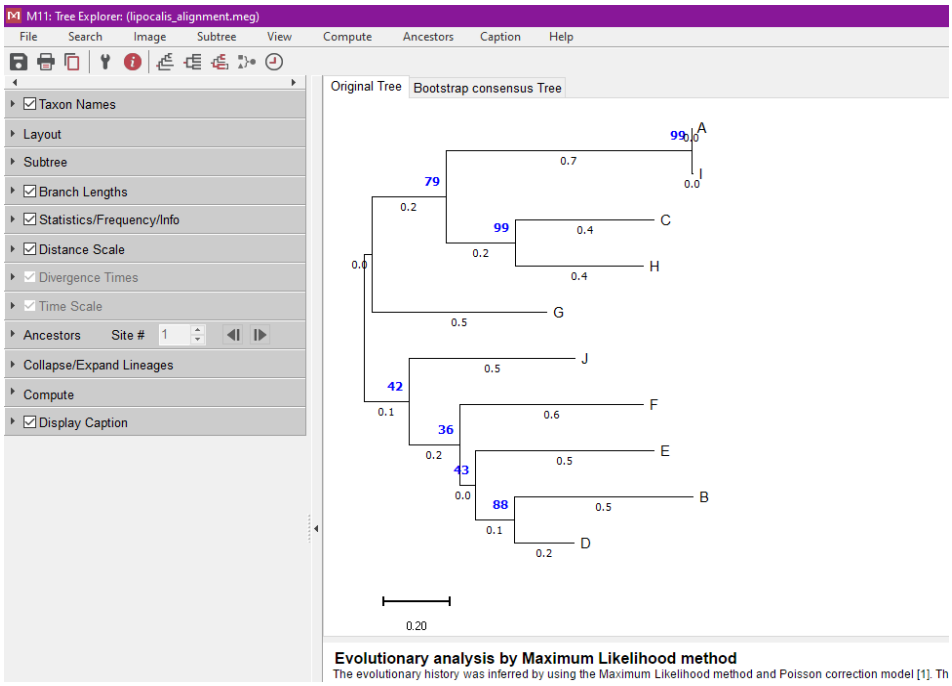


- 7.2.5, 7,2,6



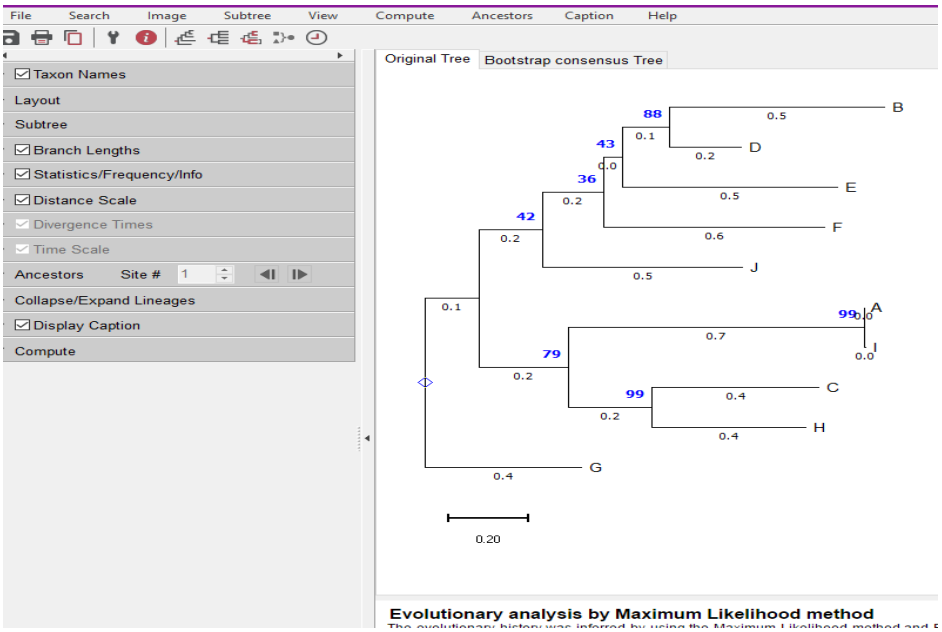
Likelihood Tree

Το δέντρο μέγιστης πιθανοφάνειας εμφανίζεται στην πιο κάτω εικόνα:

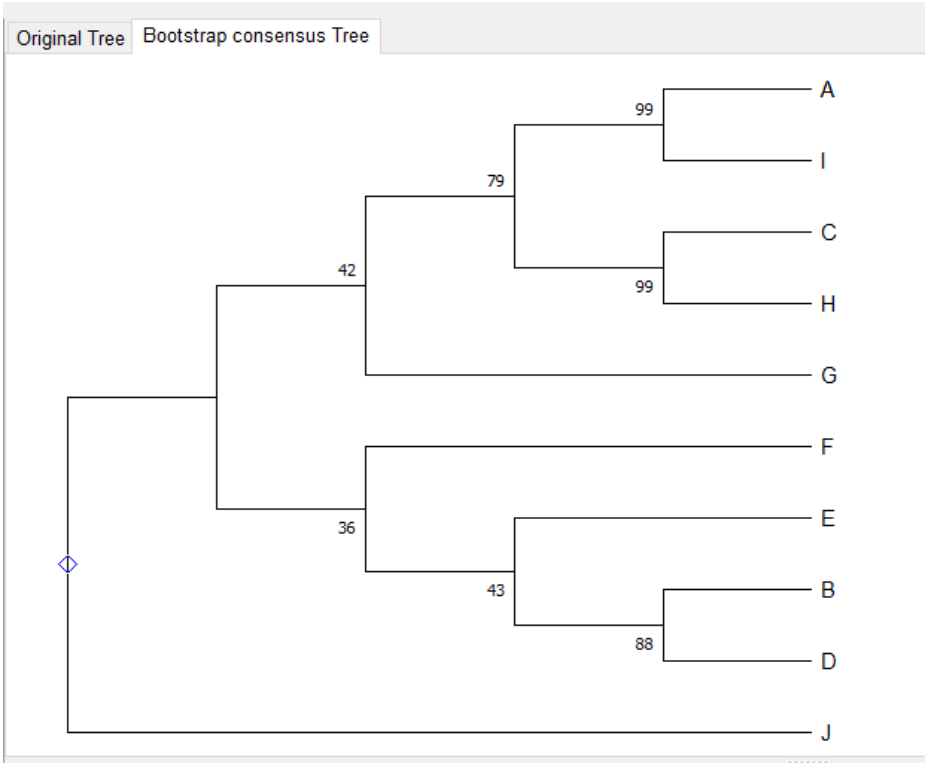


Η εξελικτική ιστορία συνάγεται με τη χρήση της μεθόδου Maximum Likelihood και του μοντέλου JTT που βασίζεται σε μήτρα. Εμφανίζεται το δέντρο με την υψηλότερη πιθανότητα καταγραφής (-4110.71). Τα αρχικά δέντρα για την ευρετική αναζήτηση ελήφθησαν αυτόματα με την εφαρμογή αλγορίθμων Neighbor-Join και BioNJ 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 10 αλληλουχίες αμινοξέων. Υπήρχαν συνολικά 218 θέσεις στο τελικό σύνολο δεδομένων.

Αλλαγή στην ρίζα του δέντρου:



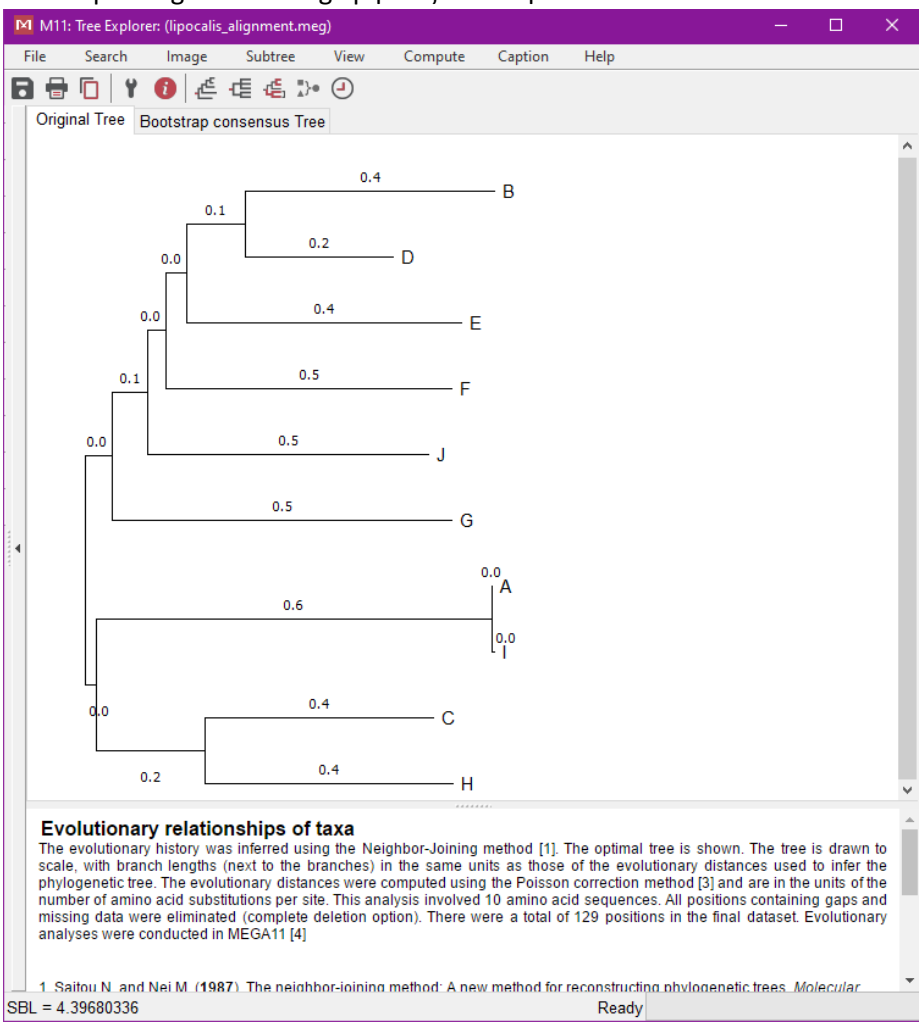
Το δέντρο μέγιστης πιθανοφάνιας με Boostrap(500):



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.

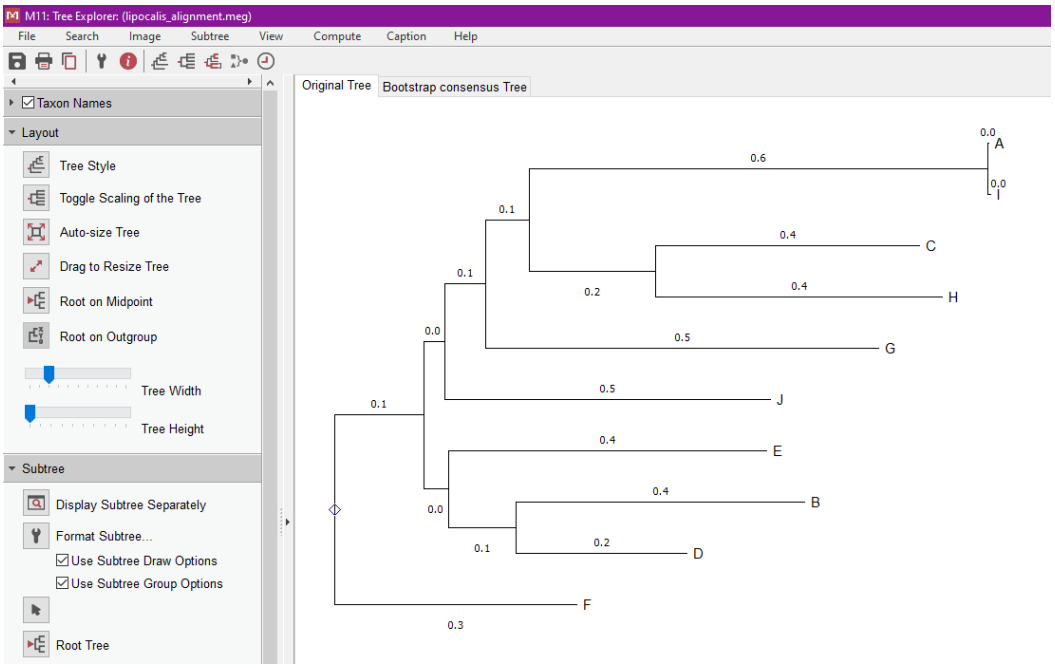
Neighbor-Join Tree

Το δέντρο Neighbor-Joining εμφανίζεται στην πιο κάτω εικόνα:

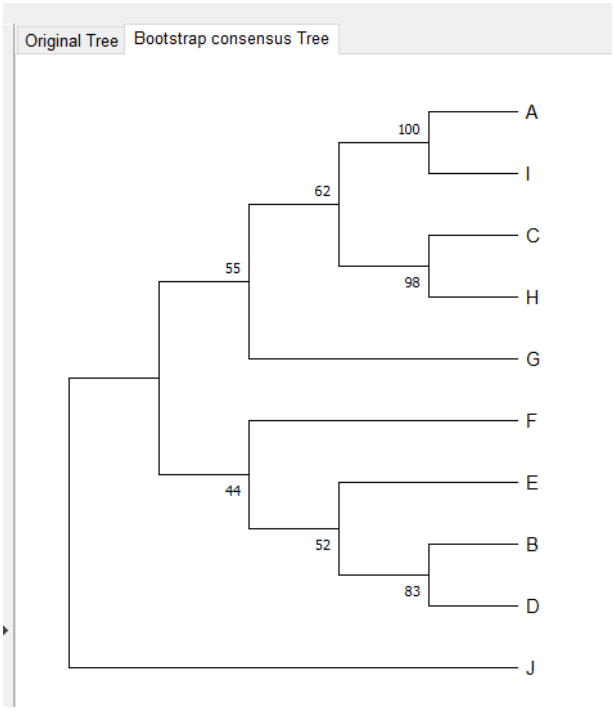


Η εξελικτική ιστορία συνήχθη με τη μέθοδο Neighbor-Joining. Εμφανίζεται το βέλτιστο δέντρο με το άθροισμα του μήκους διακλάδωσης =4.39680336. Οι εξελικτικές αποστάσεις υπολογίστηκαν χρησιμοποιώντας τη μέθοδο διόρθωσης Poisson και είναι στις μονάδες του αριθμού υποκαταστάσεων αμινοξέων ανά τοποθεσία. Αυτή η ανάλυση περιλάμβανε 10 αλληλουχίες αμινοξέων. Όλες οι θέσεις που περιέχουν κενά και ελλείποντα δεδομένα εξαλείφθηκαν (πλήρης επιλογή διαγραφής). Υπήρχαν συνολικά 129 θέσεις στο τελικό σύνολο δεδομένων.

Αλλαγή στην ρίζα του δέντρου:



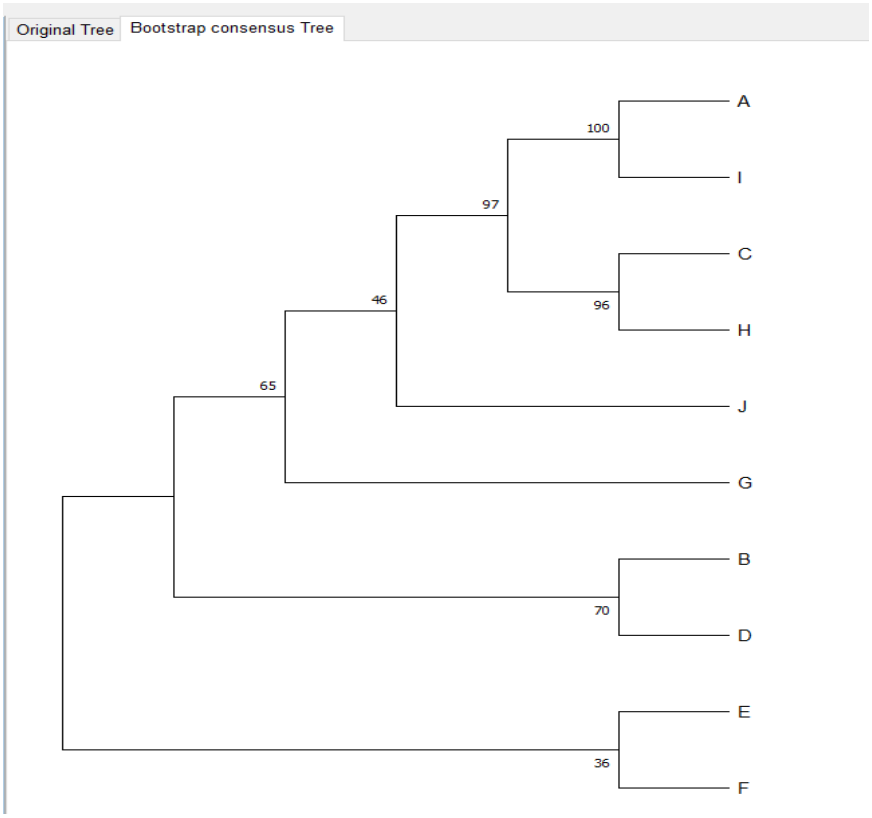
Το δέντρο Neighbor-Joining με bootstrap(500):



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.

- 7.2.7



Χαμηλά επίπεδα στήριξης έχουν συστάδες κλάδων που εμφανίζονται ως τιμές μεταξύ του 0 εως 1 (0-100%) να τείνουν προς το μηδέν.

Ένα παράδειγμα για εμάς, είναι ο κλάδος όπου περιέχει το A, I.

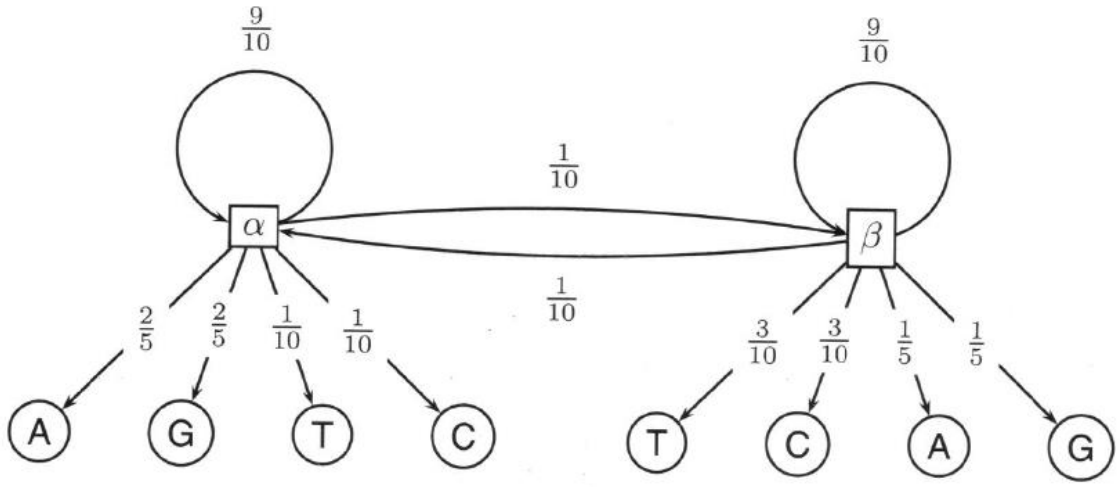
Αυτό συμβαίνει λόγω της ασάφειας, μη αξιοπιστίας διάσπασης-διακλάδωσης.

Οι κλάδοι που δεν υποστηρίζονται καλά ενδέχεται να καταρρεύσουν.

- Σημαίνει ασαφής τοπολογία.
- Ο κλαδίσκος είναι διακλαδούμενος.

➢ **Θέμα 2:**

Στο σχήμα 11.7 (βιβλίου σελ. 451) φαίνεται ένα HMM με δύο καταστάσεις α και β. Όταν το HMM βρίσκεται σε κατάσταση α, έχει μεγαλύτερη πιθανότητα να εκπέμψει πύρινες (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 αντί για κανονικές βαθμολογίες πιθανοτήτων.



Σχήμα 11.7 Το HMM που περιγράφεται στο Πρόβλημα 11.4.

Εξήγηση Κώδικα:

-Αρχικά με την λειτουργεία του προγράμματος εμφανίζεται ο πίνακας εκπομπής συμβόλων. (A,G,C,T)

-Στην συνέχεια το πρόγραμμα υπολογίζει την πιθανή ακολουθία παραγωγής της ακολουθίας GGCT

State\Nucleotides	A	G	T	C
α.	0.4	0.4	0.1	0.1
β.	0.2	0.2	0.3	0.3

Για την υλοποίηση έγινε χρήση της λογαριθμικής συνάρτησης:

$$Si,j+1= \log2\ el(xi+1) + \max \{Sk,I + \log2(akl)\}$$

Για ευκολία λύσης ορίσαμε τα εξής:

- Για καθένα από τα σύμβολα θεωρήσαμε την πιθανότητα να παραχθεί από την α ή την β κατάσταση είναι:

$$Pa(X) = P\beta\ (X) = 0.5 * Mk(X),$$

Όπου X ∃{A,G,C,T} και Mk(X) η πιθανότητα εκπομπής του συμβόλου X από την k κατάστασή του πιο πάνω πίνακα.

- Για την εύρεση της κατάσταση για κάθε ένα από τα επόμενα νουκλεοτίδια της ακολουθίας εργαστήκαμε ως εξής:

$$Pk(X,i) = Mk(X) * \max \{ Skj * Ak\lambda\},$$

Όπου Akλ το μητρώο μετάβασης καταστάσεων.
Από το HMM γνωρίζουμε ότι Aαα = Aββ = 0,9 και Aαβ = Aβα = 0,1

- Για όλα τα νουκλεοτίδια της ακολουθίας υπολογίσαμε:

Τη πιθανότητα εκπομπής ενός συμβόλου από την ακολουθία, αποθηκεύοντας το λογαριθμικό αποτέλεσμα σε ένα πίνακα.

State\Nucleotides	A	G	T	C
α.	-2.321928	-3.795859	-7.269790	-10.743722
β.	-3.321928	-5.795859	-7.684828	-9.573797

Διαλέγοντας την μεγαλύτερη πιθανότητα παραγωγής του συγκεκριμένου νουκλεοτιδίου προκύπτει η ακολουθία καταστάσεων για την ακολουθία GGCT.

1) -2.321928	2) -3.795859	3) -7.269790	4) -10.743722
5) -3.321928	6) -5.795859	7) -7.684828	8) - 9.573797

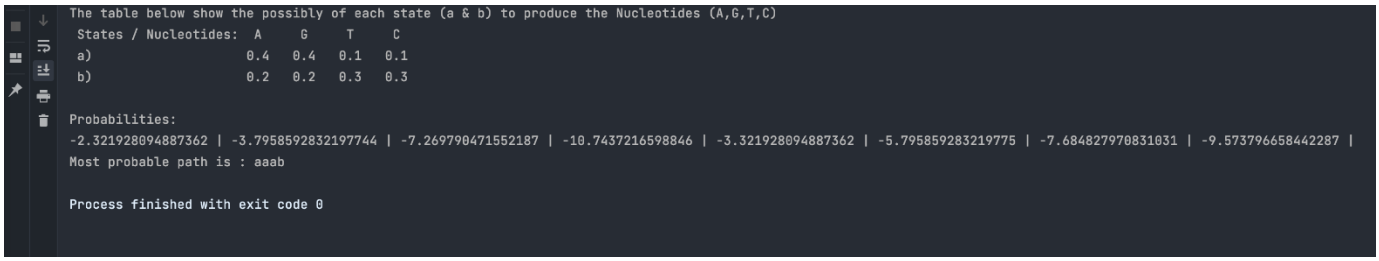
1) → 2) → 3) → 8)

Τα πιο πάνω εκτελούνται με την βοήθεια της findSituation() που δέχεται σαν ορίσματα την ακολουθία GGCT , και τον πίνακα εκπομπής συμβόλων. Εφόσον κληθεί η μέθοδος από την main, τότε με ένα βρόγχο for, διατρέχουμε την ακολουθία μέχρι να βρούμε την πιθανότερη ακολουθία καταστάσεων, από 'που προέρχεται η συμβολοσειρά. Στην πρώτη φάση της επανάληψης (δηλ., στο 0) υπολογίζεται από πια κατάσταση έχει μεγαλύτερη πιθανότητα να ξεκινήσει πρώτη, για κάθε ένα από τα σύμβολα A, C,G και T (ανάλογα με το πρώτο σύμβολό της ακολουθίας). Στην συνέχεια ανάλογα με το σύμβολο που ακολουθεί υπολογίζεται η πιθανότητα της εμφανίσεις μιας κατάστασης.
Αυτό γίνεται με την χρήση της μεθόδου findPath() σαν ορίσματα δέχεται τα προηγούμενα αποτελέσματα και μια μεταβλητή η όποια λαμβάνει 0 σε περίπτωση που βρισκόμαστε στην α κατάσταση και 1 αν βρισκόμαστε στην β κατάσταση.

Στο τέλος λογαριθμίζουμε τα αποτελέσματα και τα καταχωρούμε στον πίνακα, όπου και τα τυπώνουμε στην κονσόλα.

Όταν ο πίνακας γεμίσει, ελέγχουμε τη μεγαλύτερη πιθανότητα εμφάνισης μιας κατάστασης, και με αυτό τον τρόπο βρίσκουμε την ακολουθία καταστάσεων της ακολουθίας.

Παράδειγμα σωστής εκτέλεσης προγράμματος:



Τα αποτελέσματα είναι:

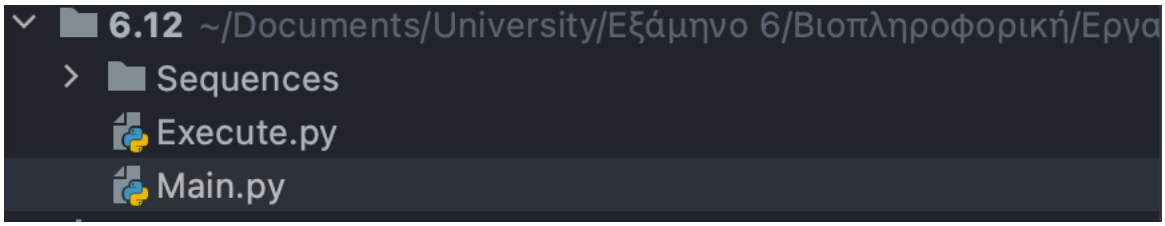
αααβ

➢ Θέμα 3:

Πρόβλημα 6.12

Δύο παίχτες παίζουν το παρακάτω παιχνίδι με δύο <<χρωμοσώματα>> που έχουν μήκος n και m νουκλεοτιδίων αντίστοιχα. Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να καταστρέψει ένα από τα χρωμοσώματα και να διαχωρίσει το άλλο σε δύο μη κενά τμήματα. Για παράδειγμα , ο πρώτος παίχτης μπορεί να καταστρέψει ένα χρωμόσωμα μήκους n και να διαχωρίσει ένα άλλο χρωμόσωμα σε δύο χρωμοσώματα με μήκη m/3 και m- (m/3). Ο παίχτης που διαγράφει το τελευταίο νουκλεοτίδιο κερδίζει. Ποιος θα κερδίσει; Περιγράψτε τη νικηφόρα στρατηγική για όλες τις τιμές των n και m.

Υλοποίηση:



Τα πιο πάνω είναι τα αρχεία κώδικα που χρησιμοποιήθηκαν για το ερώτημα.Η main καλή την συνάρτηση run από το αρχείο execute

Execute.py Main.py

```python
from Execute import run

seq = ["brain.txt", "liver.txt", "muscle.txt"]

seq1 = int(input("Choose first Sequence(1 for brain, 2 for liver, 3 for muscle): "))
seq2 = int(input("Choose second Sequence(1 for brain, 2 for liver, 3 for muscle): "))

while (seq1 == seq2) or (not 1 <= seq1 <= 3) or (not 1 <= seq2 <= 3):
    print("Please Sequences must be deference and in range 1-3!")
    seq1 = int(input("Choose first Sequence(1 for brain, 2 for liver, 3 for muscle): "))
    seq2 = int(input("Choose second Sequence(1 for brain, 2 for liver, 3 for muscle): "))

run(seq[seq1 - 1], seq[seq2 - 1])
```

Βάση των αλληλουχιών που διαλέγει ο χρήστης, καλεί την loadseq για να φορτωθούν οι σχετικές αλληλουχίες.

```
13      def run(seq1, seq2):
14          seq1 = loadSeq(seq1)
15          seq2 = loadSeq(seq2)
```

Αρχικά, διαβάζουμε τις 2 ακολουθίες αλληλουχιών που θέλουμε, οι οποίες βρίσκονται αποθηκευμένες σε txt αρχεία. Ο χρήστης διαλέγει αλληλουχίες από το 1 μέχρι το 3, ελέγχει τις εισόδους για εξασφάλιση.

```
17      if len(seq2) > len(seq1):
18          seq1, seq2 = seq2, seq1
```

Η μεγαλύτερη σε μήκος ακολουθία εισάγεται στην seq1 λίστα, και η μικρότερη στη seq2 εχτός και αν τοποθετήθηκαν από την αρχή:

```
20      canWin = []
21      winningRmv = []
22      canWin.append(False)
23      canWin.append(True)
```

Ορίζουμε τις λίστες canWin και winningRmv (κενές) με σκοπό στην συνέχεια του προγράμματος να τοποθετήσουμε την νικηφόρα ακολουθία στην winningRmv και το πως θα πρέπει να διαιρεθούν για να κερδίσουν στην canWin.

Στην αρχική θέση(0) δώσαμε την τιμή 0 (δλδ. False), γιατί αν η αλληλουχία είναι με μηδενικά στοιχεία θα έχουμε χάσει. Στη θέση 1 δώσαμε την τιμή True επειδή ο στόχος μας είναι να βγάλουμε το τελευταίο στοιχείο.

Ερευνούμε όλες τις πιθανές περιπτώσεις ξεκινώντας από το μήκος 2(αφού στο 0,1 θέσεις βάλαμε τις πιο πάνω τιμές)μέχρι το μήκος της μεγαλύτερης αλληλουχίας. Εκτός από αυτό, θα πρέπει να εξετάσουμε όλες τις διαίρεσής της ακολουθίας αυτής.

Για παράδειγμα, έχουμε τον αριθμό 8, θα πάρουμε να εξετάσουμε: [1,7], [2,6], [3,5], [4,4].

```
for i in range(2, len(seq1) + 1):
    for j in range(1, int((i / 2) + 1)):
```

Παρακάτω, ελέγχουμε τις αλληλουχίες σε κάθε περίπτωση επιστρέφουν False. Αν επιστρέφουν False, σημαίνει ότι ο επόμενος παίκτης θα χάσει οτιδίποτε και να επιλέξει, άρα εμείς θα κερδίσουμε. Αν συμβεί αυτό, αποθηκεύουμε στην λίστα winningRmv το σημείο που διαχωρίστηκε. Επιπλέον, ενημερώνουμε στην λίστα canWin ότι στο σημείο αυτό ο παίκτης μπορεί να κερδίσει. Αν ολοκληρωθεί το εμφωλευμένο loop χωρίς να βρει το κομμάτι που κερδίζει, θα τοποθετηθεί στην λίστα canWin False.

```
for i in range(2, len(seq1) + 1):
    for j in range(1, int((i / 2) + 1)):
        if not (canWin[j]) and not (canWin[i - j]):
            canWin.append(True)
            winningSplit = [j, i - j]
            winningRmv.append(winningSplit)
            break
    else:
        canWin.append(False)
```

Ελέγχουμε τα στοιχεία στις θέσεις μήκους των δύο αλληλουχιών μας, αν είναι True,θα κερδίσει ο παίχτης A, αλλιώς ο B.

```
if canWin[-1] or canWin[len(seq2)]:
    print("A is the winner.")
else:
    print("B is the winner.")
```

Τέλος, τυπώνουμε τα διαιρετέα κομμάτια που επιτρέπουν στους χρήστες

```
print("Winning splits: ")

for i in winningRmv:
    print(i)
```

Παραδείγματα σωστής εκτέλεσης προγράμματος:

```
Choose first Sequence(1 for brain, 2 for liver, 3 for muscle): 1
Choose second Sequence(1 for brain, 2 for liver, 3 for muscle): 2
Loading sequence file brain.txt...
Loading sequence file liver.txt...
A is the winner.
Winning splits:
[2, 2]
[2, 3]
[3, 3]
[2, 7]
[2, 8]
[3, 8]
[2, 12]
[2, 13]
[3, 13]
[2, 17]
[2, 18]
[3, 18]
[2, 22]
[2, 23]
```

➢ Θέμα 4:

Αναζητήστε τον κωδικό **PDB-code: 7NEH**, στην πρωτεϊνική βάση δεδομένων :
https://www.rcsb.org/
Πρόκειται για την τριδιάστατη δομή του συμπλόκου ενός αντισώματος με μια πρωτεΐνη ακίδα.

Ερώτημα 1:

a) Δείτε τα στοιχεία που παρουσιάζονται στην πρωτεϊνική βάση δεδομένων και προσδιορίστε τη μέθοδο με την οποία έχει προσδιορισθεί η δομή του συμπλόκου;
b) Ποιο το resolution (διακριτική ικανότητα) στο οποίο προσδιορίστηκε η δομή;
c) Παραθέστε το Ψηφιακό αναγνωριστικό (Digital Object Identifier, DOI) της σχετικής επιστημονικής δημοσίευσης

Ερώτημα 2:

a) Πόσες διακριτές πρωτεϊνικές αλυσίδες (molecular entities, macromolecules) περιλαμβάνει η εν λόγω δομή;
b) Για κάθε μια από αυτές σημειώστε το πλήθος των αμινοξέων (sequence length)
c) Πόσους ολιγοσακχαρίτες περιλαμβάνει η δομή του συμπλόκου;
d) Η δομή του συμπλόκου έχει ένα άτομο χλωρίου (Cl^-). Παραθέστε την αλυσίδα την οποία ανήκει

Ερώτημα 3:

a) Με χρήση του λογισμικού Chimera-X «διαβάστε» το αρχείο 7neh.pdb για να απεικονίσετε τη δομή. Παραθέστε με τη μορφή πίνακα τα στοιχεία που εμφανίζονται στο Log αρχείο και δείχνουν τις επί μέρους αλυσίδες της γλυκοπρωτεΐνης και του αντισώματος (heavy and lightchain) καθώς και των επιπλέον στοιχείων (non-standard residues) που εμφανίζονται στο αρχείο
b) Επιλέξτε την αλυσίδα που αντιστοιχεί στην πρωτεΐνη ακίδα είτε μέσω του log αρχείου είτε χρησιμοποιώντας τη γραμμή εντολών στο κάτω μέρος της οθόνης
   Command: select /E , ή εναλλακτικά
   Command: select /E:332-52
   Στη συνέχεια χρησιμοποιώντας τη γραμμή εργαλείων :
   Actions → colour → all options

Επιλέξτε μόνο το Cartoons και χρωματίστε την αλυσίδα με το χρώμα της αρεσκείας σας. Ακυρώστε την επιλογή χρησιμοποιώντας τη γραμμή εργαλείων :

Select → clear

Επαναλάβετε για τις υπόλοιπες πρωτεϊνικές αλυσίδες (βλ. ερώτημα 2_{α})Αποθηκεύστε την εικόνα που δημιουργήσατε

File → save → (επιλέξτε το format)

Παρουσιάστε την εικόνα που δημιουργήσατε- Απάντηση για το (3β)

Ερώτημα 4:

a) Σε συνέχεια του ερωτήματος 3: Επιλέξτε όπως και πριν μια μια τις αλυσίδες π.χ.

Command: select /E

Στη συνέχεια χρησιμοποιώντας τη γραμμή εργαλείων που βρίσκεται το ίδιο το παράθυρο των γραφικών:

Molecule Display → hydrophobic

Εμφανίστε την επιφάνεια πρωτεΐνης για κάθε μία αλυσίδα της πρωτεΐνης (επαναλάβετε δηλαδή εκτός από την Ε και για τις υπόλοιπες)

Αποθηκεύστε την εικόνα που δημιουργήσατε

File → save → (επιλέξτε το format)

Παρουσιάστε την εικόνα που δημιουργήσατε- Απάντηση για το (3β)

Ερώτημα 5:

a) Σε συνέχεια του ερωτήματος 3: Επιλέξτε όπως και πριν μια μια τις αλυσίδες π.χ.

Command: select /E

Στη συνέχεια χρησιμοποιώντας τη γραμμή εργαλείων

Tools → Sequence → Show Sequence viewer

Εκεί με διαφορετικό χρώμα φαίνονται τα δευτεροταγή στοιχεία της πρωτεΐνης. Επιλέξτε μόνο τους β-κλώνους (β-strands) με το ποντίκι σας ως εξής:

Επιλέξτε με το ποντίκι μια ζώνη όπως υποδεικνύεται και κρατώντας πατημένο το shift προσθέστε επιπλέον ζώνες ώστε να επιλέξετε όλες τις περιοχές που έχουν την ίδια απόχρωση και αντιστοιχούν σε β-stands.

Στη συνέχεια χρησιμοποιώντας τη γραμμή εργαλείων :

Actions → colour → all options

Επιλέξτε μόνο το Cartoons και χρωματίστε την αλυσίδα με το χρώμα της αρεσκείας σας. Επαναλάβετε για τις α-έλικες επιλέγοντας τις περιοχές με το άλλο χρώμα.

Παρουσιάστε την εικόνα που δημιουργήσατε- Απάντηση για το (5α)

b) Σε συνέχεια του ερωτήματος 3: Επιλέξτε το σάκχαρο που είναι προσδεμένο στην πρωτεΐνη ακίδα.

Command: select :NAG

Στη συνέχεια χρησιμοποιώντας τη γραμμή εργαλείων που βρίσκεται το ίδιο το παράθυρο των γραφικών:

Molecule Display → Ball and stick

Θα παρατηρήσετε ότι ένα μέρος του σακχάρου δεν έχει εφαρμόσει την εντολή. Αν περάσετε τον cursor πάνω από αυτό θα δείτε ότι το «όνομα» του σακχάρου δεν είναι NAG αλλά FUC είναι δηλαδή ένα άλλο είδος σακχάρου συνδεδεμένο με το πρώτο. Συνεπώς για να το φτιάξετε όλο με την ίδια αναπαράσταση :

Command: select :FUC

Molecule Display → Ball and stick

Παρουσιάστε την εικόνα που δημιουργήσατε- Απάντηση για το (5β)

c) Σε συνέχεια του ερωτήματος 5β:

Επαναλάβετε ό,τι και στο 5β) μόνο που τώρα τα δύο σάκχαρα θα τα «ζωγραφίσετε» με την επιλογή “sphere” δηλαδή με σφαίρες Van der Waals

Παρουσιάστε την εικόνα που δημιουργήσατε- Απάντηση για το (5γ)

Σημείωση : Σε περίπτωση που θέλετε να αλλάξετε τα χρώματα για την αναπαράσταση των σακχάρων

Command: select :FUC μετά Actions → colour → all options

Επιλέξτε μόνο το Atoms/Bonds και χρωματίστε τα άτομα με το χρώμα της αρεσκείας σας. Θα έχουν όλα το ίδιο χρώμα. Αν θέλετε να δείξετε το άζωτο με μπλέ και το οξυγόνο με κόκκινο όπως συνηθίζεται στη συνέχεια χρησιμοποιήστε την επιλογή By Heteroatom ή εναλλακτικά By Element

Απαντήσεις:

Ερώτημα 1:

a) Method: X-RAY DIFFRACTION

b) Resolution: 1.77 Å

c) 10.1016/j.cell.2021.02.033
{ http://dx.doi.org/10.1016/j.cell.2021.02.033 }

Ερώτημα 2:

a) 3 chains (A [auth H], B [auth L], C [auth E])
b) -Chain H: 222
   -Chain L: 215
   -Chain E: 205
c) Μια αλυσίδα μεγέθους 3. (D [auth A] chain)
   2-acetamido-2-deoxy-beta-D-glucopyranose-(1-4)-[alpha-L-fucopyranose-(1-6)]2-acetamido-2-deoxy-beta-D-glucopyranose
d) Chain E

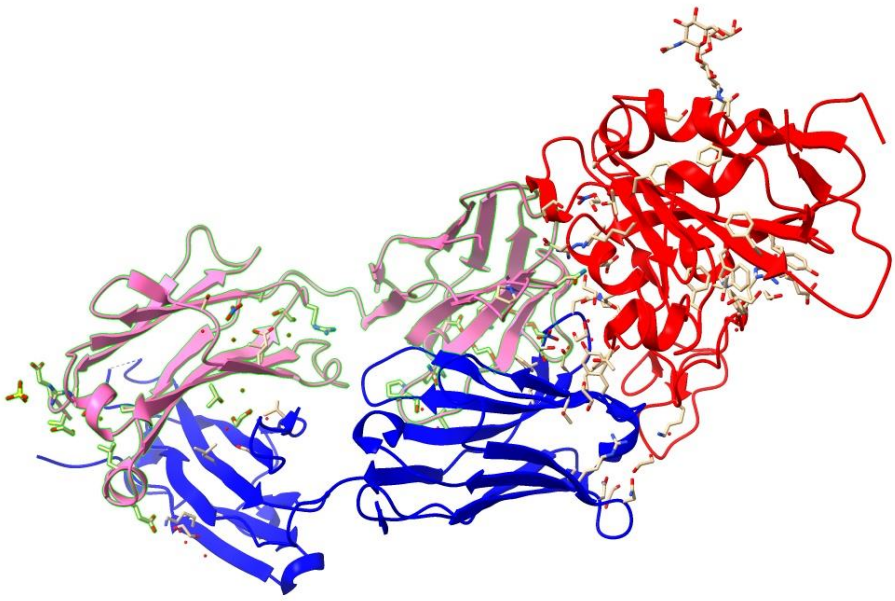
Ερώτημα 3:

a)

Non-standard residues in 7neh.pdb #1
CL — chloride ion
EDO — 1,2-ethanediol (ethylene glycol)
FUC — α-L-fucopyranose (α-L-fucose; 6-deoxy-α-L-galactopyranose; L-fucose; fucose)
NAG — 2-acetamido-2-deoxy-β-D-glucopyranose (N-acetyl-β-D-glucosamine; 2-acetamido-2-deoxy-β-D-glucose; 2-acetamido-2-deoxy-D-glucose; 2-acetamido-2-deoxy-glucose; N-acetyl-D-glucosamine)
NO3 — nitrate ion
PEG — di(hydroxyethyl)ether
SO4 — sulfate ion

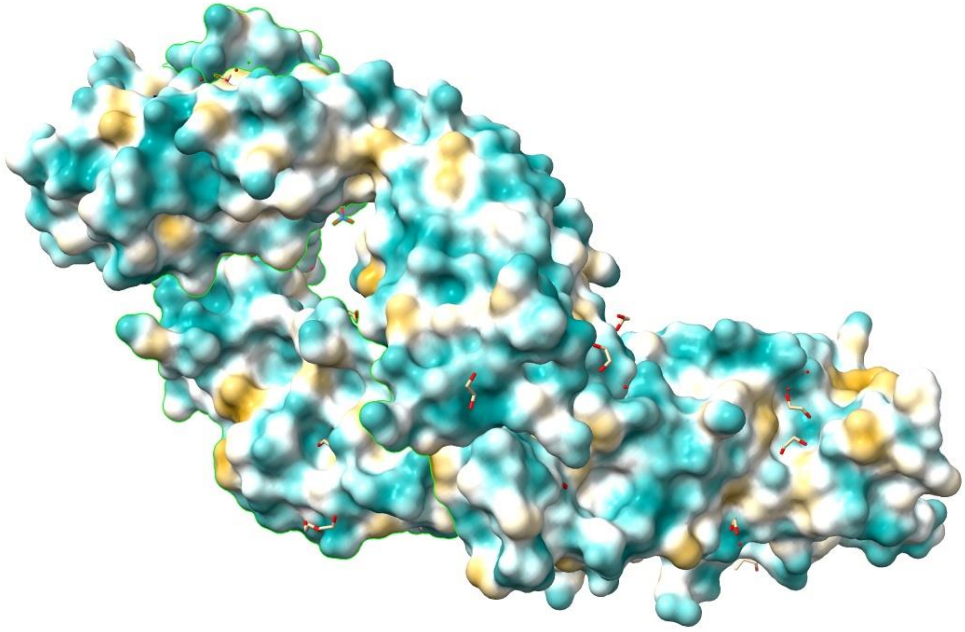
Chain information for 7neh.pdb #1	
Chain	**Description**
E	spike glycoprotein
H	covox-269 fab heavy chain
L	covox-269 fab light chain

b)



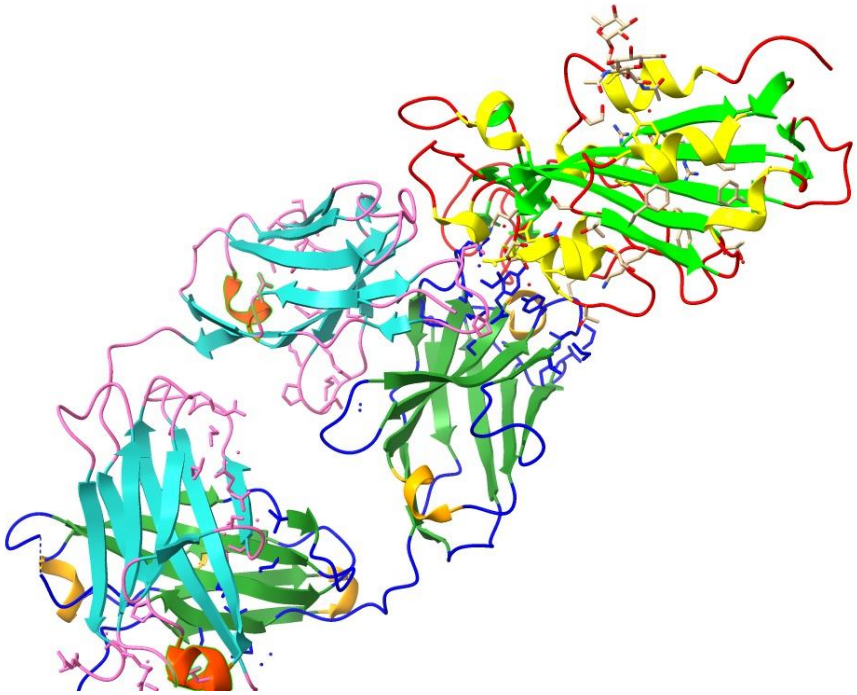
Ερώτημα 4:

a)

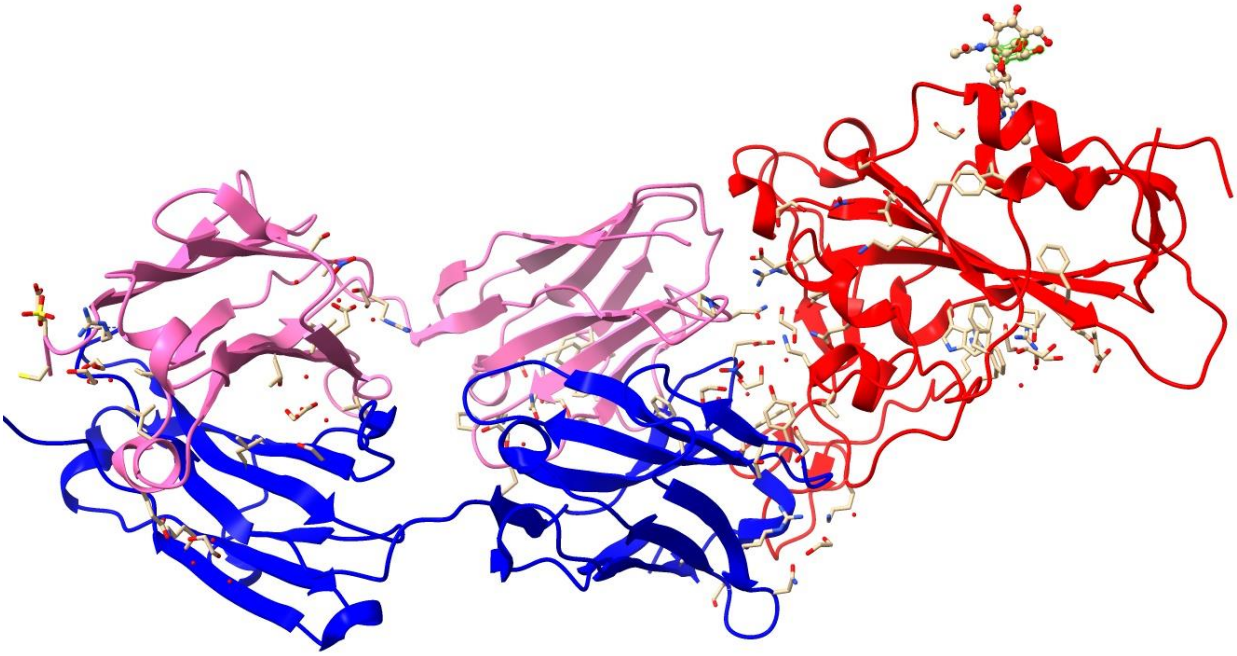


Ερώτημα 5:

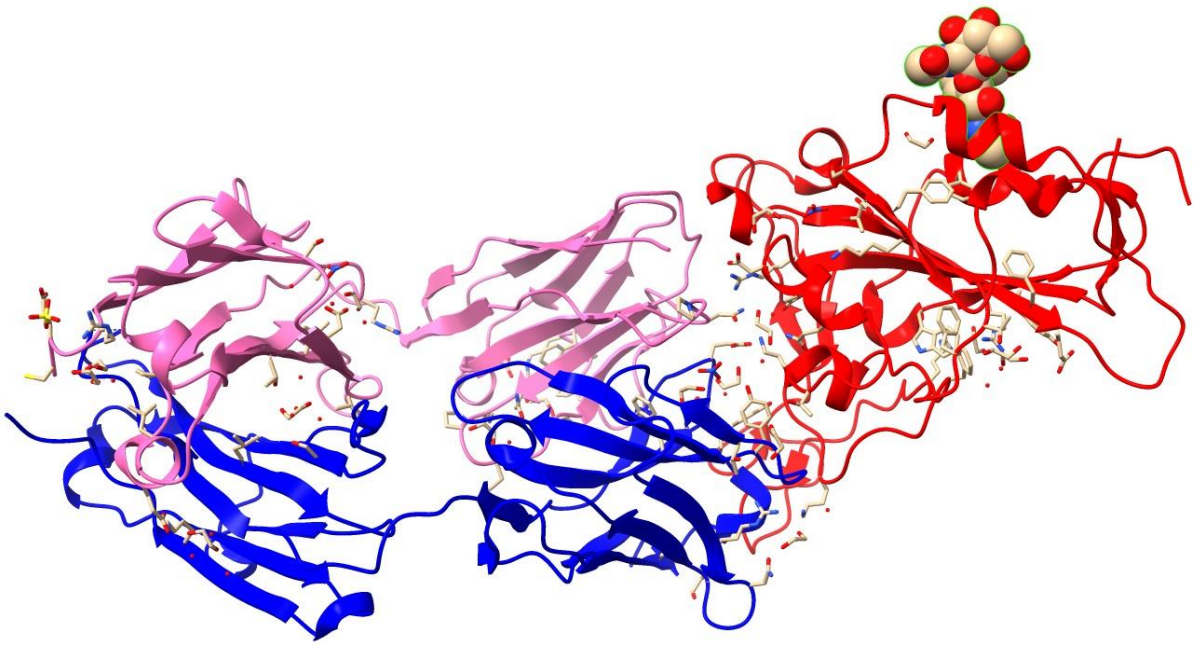
a)



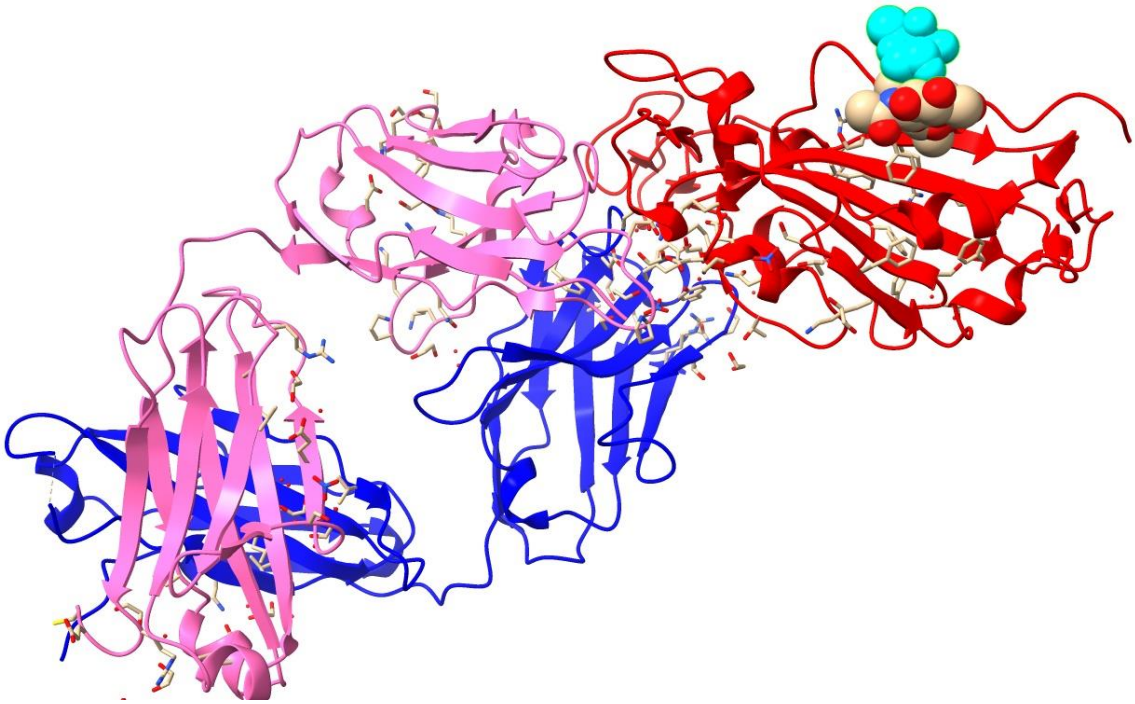
b)



c)



Σημείωση:



ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ

1. ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΚΗ ΓΟΝΙΔΙΩΜΑΤΙΚΗ: JONATHAN PEVSNER
2. BIOINFORMATICS FOR BEGINNERS: GENES, GENOMES, MOLECULAR EVOLUTION, DATABASES AND ANALYTICAL TOOLS: SUPRATIM CHOUDHURI
3. https://www.ncbi.nlm.nih.gov/
4. https://www.rcsb.org/
5. https://www.ncbi.nlm.nih.gov/pmc/