

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής



Εργασία Μαθήματος «ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ»

«αριθμός άσκησης»	Απαλλακτική εργασία
Όνομα φοιτητή – Αριθμός Μητρώου	Πετρίδης Αχιλλέας Π18211 Φύτρος Ευάγγελος Π18220
Ημερομηνία παράδοσης	05/07/2021



Εκφώνηση της άσκησης

Εργασία Βιοπληροφορικής, 2021

Ημερομηνία παράδοσης: Παρασκευή 9/7, 23:59

Η εργασία εκπονείται σε ομάδες 1-3 φοιτητών. Παραδίδονται:

α) η τεκμηρίωση της εργασίας σε ένα αρχείο pdf, στην πρώτη σελίδα της οποίας αναγράφονται τα ονοματεπώνυμα των φοιτητών και οι ΑΜ. Δεν θα βαθμολογηθούν εργασίες που δεν περιέχουν τεκμηρίωση ή που δεν αναφέρουν τα ονόματα των μελών της ομάδας στην τεκμηρίωση.

β) τα αρχεία source code σε ένα συμπιεσμένο αρχείο με όνομα source2021.zip (ή .rar ή άλλη σχετική κατάληξη).

γ) οποιαδήποτε άλλα συνοδευτικά αρχεία η ομάδα κρίνει απαραίτητα σε ένα συμπιεσμένο αρχείο με το όνομα auxiliary2021.zip (ή .rar ή άλλη σχετική κατάληξη).

Θέμα 1 (2.5 βαθμοί): Άσκηση 7.2, βιβλίο "Βιοπληροφορική και Λειτουργική Γονιδιωματοική"

Θέμα 2 (2.5 βαθμοί): Άσκηση 11.4, βιβλίο "Εισαγωγή στους Αλγορίθμους Βιοπληροφορικής".

Θέμα 3 (2 βαθμοί): Υλοποιήστε μία από τις ασκήσεις 6.12, 6.13, 6.14 του βιβλίου "Εισαγωγή στους Αλγορίθμους Βιοπληροφορικής".

Θέμα 4 (3 βαθμοί): Η εκφώνηση παρατίθεται ως ξεχωριστό αρχείο

Για τα Θέματα 2 και 3, αποδεκτές γλώσσες υλοποίησης είναι οι Python, Matlab. Κάθε θέμα πρέπει να συνοδεύεται από τεκμηρίωση της λύσης. Το θέμα 3 συσχετίζεται με επιπρόσθετα δεδομένα, τις περιγραφές των οποίων θα βρείτε στον σχετικό κατάλογο της ενότητας εγγράφων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΘΕΜΑ 1°	4
1.1	Μεταβείτε στην βάση δεδομένων των συντηρημένων δομικών επικρατειών στο NCBI	4
1.2	Εισάγετε τον όρο λιποκαλίνες ή άλλο όνομα οικογένειας της επιλογής σας.	5
1.3	Επιλέξτε τη μορφή αρχείου mFasta και στην συνέχεια κάντε κλικ στην επιλογή <<Reformat>>. Το αποτέλεσμα είναι μια πολλαπλή στοίχιση αλληλουχιών. Αντιγράψτε το αποτέλεσμα σε έναν επεξεργαστή κειμένου και απλοποιήστε τα ονόματα των αλληλουχιών.....	5
1.4	Εισάγετε το αρχείο στο MEGA, όπως φαίνεται στην Εικόνα 7.9. Στοιχίστε τις αλληλουχίες και αποθηκεύστε τις σε μορφές αρχείων .mas και .meg.....	7
1.5	Επιλέξτε Phylogeny > Construct/Test για να δημιουργήσετε δέντρα με τις μεθόδους ένωσης γειτόνων, μέγιστης πιθανοφάνειας ή άλλες.	8
2	ΘΕΜΑ 2°	8
2.1	Κώδικας της άσκησης.....	9



2.2	Συνοπτική επεξήγηση	10
3	ΘΕΜΑ 3 ^ο	11
3.1	Κώδικας της άσκησης.....	12
3.2	Συνοπτική επεξήγηση	13
4	ΘΕΜΑ 4 ^ο	13
4.1	Ερώτημα 1 ^ο	13
4.2	Ερώτημα 2 ^ο	13
4.3	Ερώτημα 3 ^ο	14
4.4	Ερώτημα 4 ^ο	16
4.5	Ερώτημα 5 ^ο	17



1 ΘΕΜΑ 1^ο

1.1 Μεταβείτε στην βάση δεδομένων των συντηρημένων δομικών επικρατειών στο NCBI

The screenshot displays the NCBI Conserved Domains (CDD) website. At the top, there is a navigation bar with links to NCBI, Resources, and How To. A search bar is prominently featured, with a red box highlighting the input field. Below the search bar, a COVID-19 information banner is visible, providing links to public health information, research information, SARS-CoV-2 data, and prevention and treatment information. The main content area is titled "CDD" and includes a description of the database as a resource for protein annotation. To the left of the description, there is a diagram showing protein domain structures. Below the description, there are three columns of links: "Using CDD" (Quick Start Guide, How To Guides, Help, FTP, News, Publications), "CDD Tools" (Overview of CDD Resources, CD-Search, Batch CD-Search, CDART, SPARCLE, BLAST), and "Other Resources" (Structure Group Home Page, Entrez Structure, Entrez Gene, Entrez Protein). The footer of the page contains a "You are here" breadcrumb trail and a "Support Center" link. Below these, there are five columns of links: "GETTING STARTED" (NCBI Education, NCBI Help Manual, NCBI Handbook, Training & Tutorials, Submit Data), "RESOURCES" (Chemicals & Bioassays, Data & Software, DNA & RNA, Domains & Structures, Genes & Expression, Genetics & Medicine, Genomes & Maps, Homology, Literature, Proteins, Sequence Analysis), "POPULAR" (PubMed, Bookshelf, PubMed Central, BLAST, Nucleotide, Genome, SNP, Gene, Protein, PubChem), "FEATURED" (Genetic Testing Registry, GenBank, Reference Sequences, Gene Expression Omnibus, Genome Data Viewer, Human Genome, Mouse Genome, Influenza Virus, Primer-BLAST, Sequence Read Archive), and "NCBI INFORMATION" (About NCBI, Research at NCBI, NCBI News & Blog, NCBI FTP Site, NCBI on Facebook, NCBI on Twitter, NCBI on YouTube, Privacy Policy).



1.2 Εισάγετε τον όρο λιποκαλίνες ή άλλο όνομα οικογένειας της επιλογής σας.

NCBI Resources How To Sign in to NCBI

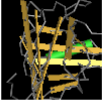
Conserved Domains Conserved Domains lipocalins Search Help


Create alert Advanced

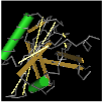
COVID-19 Information
Public health information (CDC) | Research information (NIH) | SARS-CoV-2 data (NCBI) | Prevention and treatment information (HHS) | Español


Summary 20 per page Sort by Default order Send to Filter your results:


Search results
Items: 1 to 20 of 57


1.  **Lipocalin_FABP: lipocalin/cytosolic fatty acid-binding protein family**
Lipocalins are diverse, mainly low molecular weight extracellular proteins that bind principally sma...
Accession: cl10502 ID: 415860
[View in Cn3D](#) [Protein](#) [Superfamily Members](#) [PubMed](#)

2.  **Lipocalin_LTBP1-like: 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), i...
Accession: cl19423 ID: 381198
[View in Cn3D](#) [Specific Protein](#) [Protein](#) [Superfamily](#) [Superfamily Members](#) [PubMed](#)

3.  **Lipocalin: Lipocalin / cytosolic fatty-acid binding protein family**
Lipocalins are transporters for small hydrophobic molecules, such as lipids, steroid hormones, bilin...
Accession: cl21528 ID: 419712
[View in Cn3D](#) [Protein](#) [Superfamily Members](#) [PubMed](#)

4.  **Palliysin: Palliysin beta barrel domain**
The *Treponema pallidum* protein, Tp0751 (also known as palliysin), possesses adhesive properties and...
Accession: cl39980 ID: 423742
[View in Cn3D](#) [Protein](#) [Superfamily Members](#) [PubMed](#)

5.  **Lipocalin_Bla_g_4_Per_a_4: major allergens Bla g 4 and Per a 4**
Inhalant allergens from cockroaches are an important cause of asthma. Bla g 4 and Per a 4 are male p...
Accession: cl19440 ID: 381215
[View in Cn3D](#) [Specific Protein](#) [Protein](#) [Superfamily](#) [Superfamily Members](#) [PubMed](#)

6.  **Lipocalin_Blc-like: bacterial lipocalin Blc, Arabidopsis thaliana temperature-induced lipocalin-1 and similar proteins**
Escherichia coli bacterial lipocalin (Blc, also known as YjeL) is an outer membrane lipoprotein invo...
Accession: cl19438 ID: 381213

Find related data
Database: Select Find items

Search details
lipocalins[All Fields]
Search See more...

Recent activity
Turn Off Clear

Q lipocalins (57)
Conserved Domains

Hypothetical protein
Protein

Q 81750622 (0)
Protein

Domain of unknown function DUF1794
[Acidimicrobium ferrooxidans DSM 103 Protein
RecName: Full=UPF0678 fatty acid-binding

1.3 Επιλέξτε τη μορφή αρχείου mFasta και στην συνέχεια κάντε κλικ στην επιλογή <<Reformat>>. Το αποτέλεσμα είναι μια πολλαπλή στοίχιση



αλληλουχιών. Αντιγράψτε το αποτέλεσμα σε έναν επεξεργαστή κειμένου και απλοποιήστε τα ονόματα των αλληλουχιών

Sequence Alignment

include consensus sequence ?

Reformat

Format:

mFasta

Row Display:

up to 10

Color Bits:

2.0 bit

Type Selection:

the most diverse members

>gi|609411913|pdb|3WJ8|B

--mwhsqfeknqlqqlqnpqesppvhpFVAPLSYLLGTWRGQGEYPTIPSFYRGEEIRFSH-SGKPVIAVTQKTWKL

ESgA-PLLAESGYFRPrPDGS--IEVVIACSTGLVEVQKGTYNv--dEqSIKLKS---DLV---GNASKVKEISREFELV

DGK---LSVVRLSTTTNPQLPLKAILDk1-----

>gi|504799881|ref|WP_014986983.1|

-----mvesqppiaphpdIAPLAALLGTWRGNHGEYPTIQPFDYLEEVRFGH-LGRPFLTYRQRTAA

DD-GrPMHAETGYLR--CPRPdrVELILAHPTGITEICEGALTvddgAlhLEFDS---SSIgrsSTAKLVTALGRFTQV-

KGDt--IDYTVRMAAVGEPLQHHLAATLIrae-----

>gi|218551765|sp|A1T297.1|Y449_MYCVP

-----madsvpalhpdlAALAPLLGTWVGEGSGEYPTIEPFYTEEITFGH-VGKPFLLTYAQRTRAA

DD-GrPLHAETGYLR--ASAPdrIEWILAHPTGITEIQEGQLTadgdGlrMELVS---SSIgrsGSAKEVTDVGRSIEL-

RGDt--LTYTLRMAAVGQPLQHHLASAVLRrvr-----

>gi|333487064|gb|AEF36456.1|

-----mELAPLLGTWSGRGRGVYPTIASFDYLEEVTFSH-VGKPFLLVYGQKTksA

AD-G1PLHAETGYLR--VPQPgrIEWLAHPSGITEIEVGSYRvtadGieLEMSA---PTIglPTAKEVTALSRRYRL-

ARDe--LSYTLDMGAVGEPANHLTAALRrtg-----

>gi|218551734|sp|B2HLY1.1|Y1995_MYCMM

-----mpadlhpdlDALAPLLGTWAGQSGEYPTIEPFYLEEVRFSH-VGKPFLLVYAQKTRAV

AD-GaPLHAETGYLR--VPKPGqVELVLAHPSGITEIEVGTYSasggVieMEMVT---TAIgmtPTAKEVTALSRFRM-

VGDDe--LSYRLRMGAVGLPLQHHLGARLRrks-----

>gi|81413567|sp|Q73W27.1|Y2833_MYCPA

-----mptdlhpdlAALAPLLGTWGRGSGKYPTIQPFDYLEEVTFSH-VGKPFLLAYAQKTRAA

AD-GkPLHAETGYLR--VPQPgrLELVLAHPSGITEIEVGSYAvtgGLieMRMST---TSIglTPSAKEVTALARWFRI-

DGDe--LSYSVQMGAvgQPLQDHLAAVLHrq-----

>gi|88193107|pdb|2FR2|A

-----mtrdlapaLQALSPLLGSWAGRGAGKYPTIRPFYLEEVRFAH-VGKPFLLTYTQQTTRAV

AD-GkPLHSETGYLR--VCRPGcVELVLAHPSGITEIEVGTYSvtgdVieLELSTradGSIglPTAKEVTALDRSYRI-

DGDe--LSYSLQMRavgQPLQDHLAAVLHrqrrshhhhhh

>gi|81537071|sp|Q9CCB8.1|Y1006_MYCLE

-----mpsdlcpdlQALAPLLGSWVGRGMKYPTIQPFYLEEVRFSH-LDRPFLTYTQKTRAI

TD-GkPLHAETGYLR--VPQPghIELVLAHSDIAEIEVGTYSvtgdLieVELVT---TTIglvPTAKQVTALGRFFRI-

DGDe--FAYSVMGAVGQPLQDHLVAVLHrkq-----

>gi|256008631|gb|ACU54198.1|

-----mtireMLEGTWTGSGIGSYPEVAEFSYQERLRFES-NGRPFLRMEQHT-TG

PD-GsPLHTEVGYLR--FVANgrLELVAAQPTGIVEVLEGGAEaeggvvISLTS---LVVattPSAKRVDVAVQRRFTL-

RGDe--LMTELWMDAMERGMMQHLESRLRrg-----

>gi|81750622|sp|Q8FTT5|Q8FTT5_COREF

mgarvermlfdarsapi1rstgmdlhpnlAPYGFLLGTWTGKGHGFYPTIEDFSYEETLNFSTiPGKPFPRYEQKTMGL

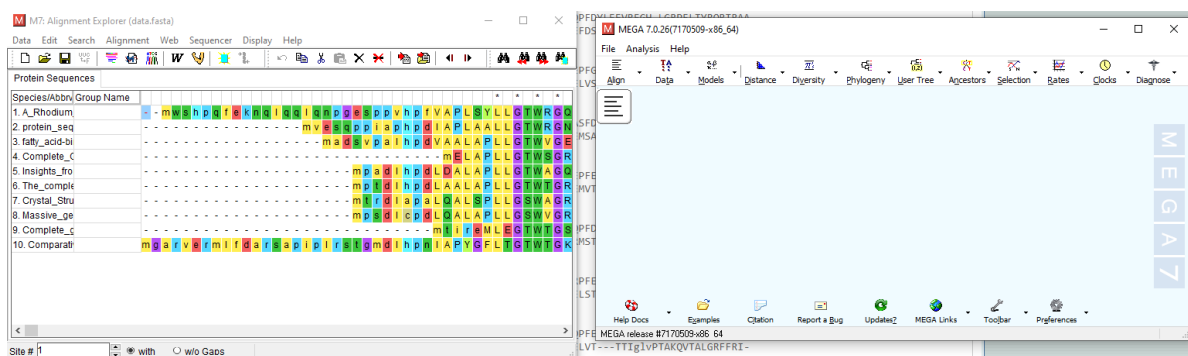
Dg---PLHTELGLFLR--ILDDgrAEFILAQPMGQTELEGMVSeegETlvDFDTR---STVansGSAKRVTGTARTYTF-

TADrtgLSAQFAMGAVGQPLQDHLSELETKqs-----



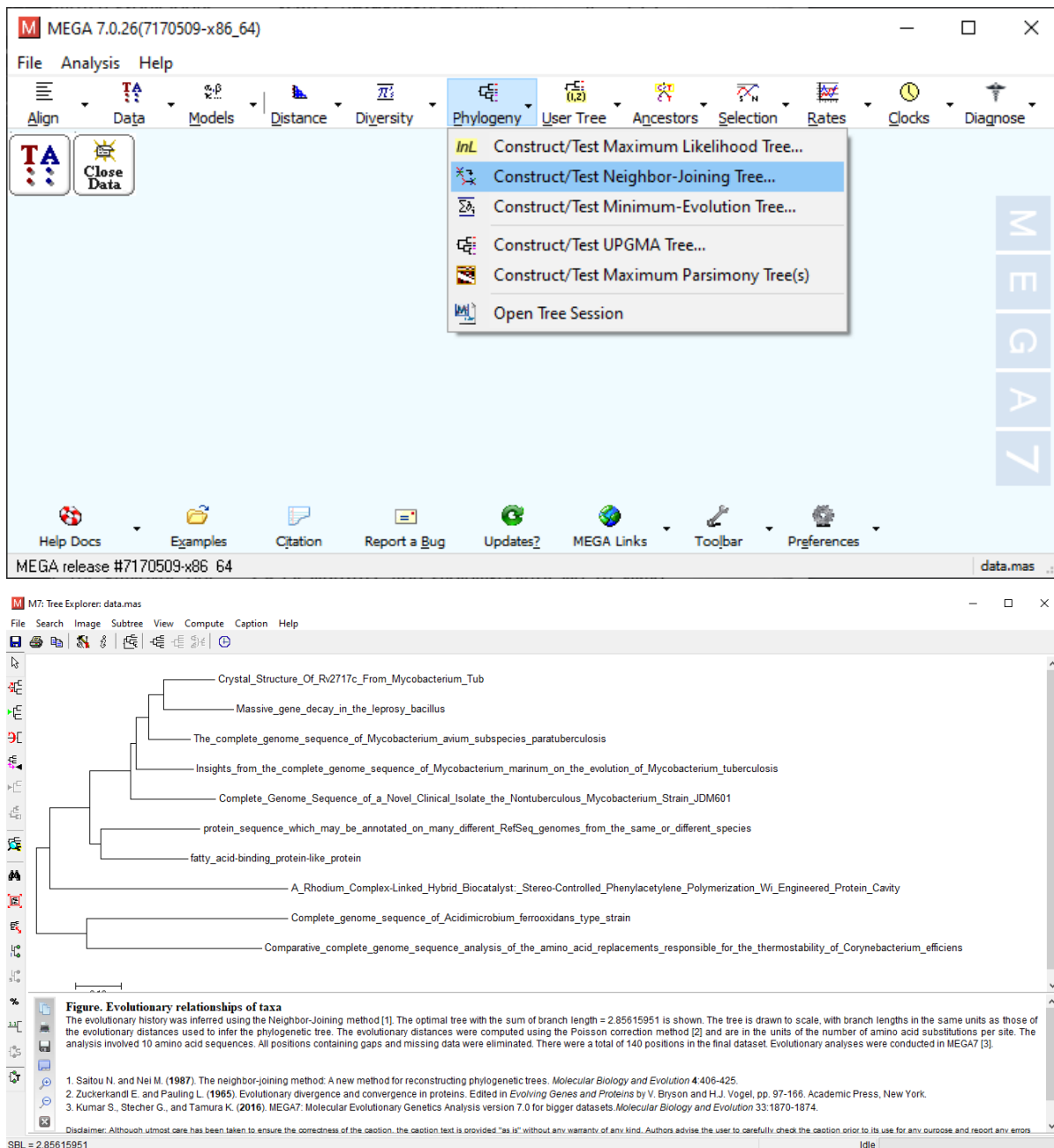
```
>A Rhodium Complex-Linked Hybrid Biocatalyst: Stereo-Controlled Phenylacetylene Polymerization Wi Engineered Protein Cavity
--mwshpqfeknqlqqlnpgespvpvhpFVAPLSYLLGTWRGQGEYPTIPSFYRGEIIRFSH-SGKPFVIAYTQKTKWL
ESgA-PLLAESGYFRprPDGS--IEVVIACSTGLVEVQKGTYNv--dEqsKCLKS---DLV---GNASKVKEISREFELv
DGK--LSYVVRLSTTTNPLQQLKAILDKl-----
> protein sequence which may be annotated on many different RefSeq genomes from the same, or different, species
-----mvesqppiaphdpdIAPLAALLGTWRGNGHGEYPTIQPFYILEEVVFGH-LGRPFITYRQRTAA
DD-GkPLHAETGYLR--CPRPdrVELILAHPTGITEICEGALTvdgdAlhLEFDS---TSigrsSTARLVTALGRTFQV-
RGDt--IDYTVRMAAVGELQHLAATLlrae-----
>fatty acid-binding protein-like protein
-----masvpalhpdlVAALAPLLGTWVGEGSGEYPTIEPFYITEEITFGH-VGKPFITYAQRTAA
DD-GkPLHAETGYLR--ASAPdrIEWILAHPTGITEIQEQLTadgdGlrMELVS---SSIGrSGSAKEVTDVGRSIEL-
RGDt--LTYTLRMAAVGQPLQHLASAVLRvr-----
>Complete Genome Sequence of a Novel Clinical Isolate, the Nontuberculous Mycobacterium Strain JDM601
-----mELAPLLGTWSGRGRGVPTIASFDYLEEVVFSH-VGKPFIVYGQKTRSA
AD-GlPLHAETGYLR--VPQpgrIEWVLAHPSGITEIEVGYSrvtadGleLEMSA---PTIglPTAKEVTALSRRYRL-
ARDe--LSYTLDMGAVGEPAQNHLTAALRrtg-----
>Insights from the complete genome sequence of Mycobacterium marinum on the evolution of Mycobacterium tuberculosis
-----mpadlhpdlDALAPLLGTWAGQSGEYPTIEPFYILEEVVFSH-VGKPFIVYAQKTRAV
AD-GkPLHAETGYLR--VPQpgrLELVLAHPSGITEIEVGYSasggVleMEMVT---TAIgmtPTAKEVTALSRFRM-
VGDe--LSYRLMGAVGLPLQHLGARLRks-----
>The complete genome sequence of Mycobacterium avium subspecies paratuberculosis
-----mptdlhpdlAALAPLLGTWGRSGKYPTIQPFYILEEVVFSH-VGKPFIVYAQKTRAA
AD-GkPLHAETGYLR--VPQpgrLELVLAHPSGITEIEVGYSavtgdlLleMRMST---TSIgtPSAKEVTALSRFRM-
DGDe--LSYSVQMGAVGQPLQDLHAAVLHrqr-----
>Crystal Structure Of Rv2717c From Mycobacterium Tub
-----mtrdlapaLQALSPLLGSWAGRGAGKYPTIRPFYILEEVVFAH-VGKPFITYTQQTAV
AD-GkPLHSETGYLR--VCRPgcVELVLAHPSGITEIEVGYSvtgdLleLELSTradGSiglaPTAKEVTALGRFFRI-
DGDe--LSYSLQMRVAVGQPLQDLHAAVLHrqrshhhhhh
>Massive gene decay in the leprosy bacillus
-----mpsdlcpdlQALAPLLGSWVGGMKYPTIQPFYILEEVVFSH-LDRPFITYTQKTRAI
TD-GkPLHAETGYLR--VPQpgrLELVLAHPSGITEIEVGYSvtgdLleLELSTradGSiglaPTAKEVTALGRFFRI-
DGDe--FAYSVMGAVGQPLQDLHAAVLHrkg-----
>Complete genome sequence of Acidimicrobium ferrooxidans type strain
-----mtireMLEGTWTSIGSGISYEPFAEFSYQERLRFES-NGRPFIRMEQHT-TG
PD-GsPLHTEYGYLR--FVANgrLELVVAOPTGIVEVLEGGAEaeggvvlSLTS---LVVattPSAKRVDVAVQRFTL-
RGDe--LMTLWMDAMERGMMQHLESRLRrg-----
>Comparative complete genome sequence analysis of the amino acid replacements responsible for the thermostability of Corynebacterium efficiens
ngarvermlfdarsapilrstmghlhpniAPYGLTGTWTRGRHGFPYTIEDFSYEETLNFSTIPGRPFYRQKTMGL
Dg---PLHTELGFLR--ILDDgrAEFILAQPMGQTELEGMVSeegvTlvFDFTR---STVansGSARRVGTARTYTF-
TADrtgLSAQFAMGAVGQPLQHLHSELTKqs-----+
```

1.4 Εισάγετε το αρχείο στο MEGA ,όπως φαίνεται στην Εικόνα 7.9. Στοιχίστε τις αλληλουχίες και αποθηκεύστε τες σε μορφές αρχείων .mas και .meg





1.5 Επιλέξτε Phylogeny > Construct/Test για να δημιουργήσετε δέντρα με τις μεθόδους ένωσης γειτόνων, μέγιστης πιθανοφάνειας ή άλλες.



2 ΘΕΜΑ 2^ο

Στο Σχήμα 11.7 φαίνεται ένα HMM με δύο καταστάσεις α και β. Όταν το HMM βρίσκεται στην κατάσταση α, έχει μεγαλύτερη πιθανότητα να εκπέμψει πουρίνες [Α 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.

2.1 Κώδικας της άσκησης

```

1 from numpy import log as ln
2 two_states=('A','B') #Δύο καταστάσεις (A,B)
3
4 a_pos = [['A','G','T','C'],[0.4,0.4,0.1,0.1]] #H πιθανότητα να εκπέμψει πουρίνες και πυριμιδίνες στην κατάσταση A
5 b_pos = [['T','C','A','G'],[0.3,0.3,0.2,0.2]] #H πιθανότητα να εκπέμψει πουρίνες και πυριμιδίνες στην κατάσταση B
6
7 goal=('G','G','C','T') #H ζητούμενη αλληλουχία
8
9 from_a_to_a_and_b=[0.9,0.1] #H πιθανότητα να συνεχιστεί από την A κατάσταση στην A και στην B
10 from_b_to_a_and_b=[0.1,0.9] #H πιθανότητα να συνεχιστεί από την B κατάσταση στην B και στην A
11
12 start_possibility=[0.5,0.5] #H αρχική πιθανότητα να ξεκινήσει από το A και το B
13
14 def algo(a_pos,from_a_to_a_and_b,goal,start_possibility,b_pos,from_b_to_b_and_a,two_states):
15     temp_a=[] #Αποθηκεύονται οι τιμές της A κατάστασης από τον viterbi αλγόριθμο
16     temp_b=[] #Αποθηκεύονται οι τιμές της B κατάστασης από τον viterbi αλγόριθμο
17     temp=True
18     temp1=False
19     temp2=False
20     start=False
21     log_num=0
22     i=0
23     best_path=[]
24     while(temp):
25         for i in range(0,len(goal)): #Μένει σε μια for με μέγεθος που είναι η (ζητούμενη αλληλουχία(goal))
26             if(start=False): #Εάν το start είναι false σημαίνει ότι βρίσκουμε στην πρώτη κατάσταση με (ζητούμενο) το G
27                 if(a_pos[0][i]==goal[i]): #Εάν το a_pos[0][i] ισούται με το (ζητούμενο)
28                     temp_a.append(start_possibility[0]*a_pos[0][i]) #Τότε η temp_a παίρνει την αρχική πιθανότητα του A και την πολλαπλασιάζει με την πιθανότητα του (ζητούμενου νουκλεοτιδίου)
29                     temp1=True
30                 if(b_pos[0][i]==goal[i]):
31                     temp_b.append(start_possibility[1]*b_pos[1][i]) #Τότε η temp_b παίρνει την αρχική πιθανότητα του B και την πολλαπλασιάζει με την πιθανότητα του (ζητούμενου νουκλεοτιδίου)
32                     temp2=True
33             if(temp1 and temp2): #Εάν το temp1 και το temp2 είναι true τότε έχουμε βρει το πρώτο (ζητούμενο(goal[0]))
34                 start=True #Το start γίνεται True οπότε το πρόγραμμα δεν μνημονεύει ξανά στην αρχική if
35                 i+=1
36                 if(temp_a[0]>temp_b[0]): #Εάν το temp_a που βρέθηκε είναι μεγαλύτερο από το temp_b που βρέθηκε τότε
37                     best_path.append(two_states[0]) #Το best_path παίρνει την πρώτη κατάσταση
38                     log_num=ln(temp_a[0]) #Μετατρέπεται τις αριθμητικές τιμές σε λογαριθμικές και τις προσθέτει σε μια μεταβλητή log_num
39                 elif(temp_a[0]<temp_b[0]): #Αλλιώς το best_path παίρνει την δεύτερη κατάσταση
40                     best_path.append(two_states[1])
41                     log_num=ln(temp_b[0]) #Μετατρέπεται τις αριθμητικές τιμές σε λογαριθμικές και τις προσθέτει σε μια μεταβλητή log_num
42                 break #Αφού βρεθεί ποιο είναι μεγαλύτερο κάνει break από την for και (εκκινεί για τις υπολοίπες καταστάσεις αφού το start γίνεται True στην σειρά 36
43             if(start and temp1 and temp2): #Εάν και τα τρία temp είναι true ψάχνουμε τις υπολοίπες καταστάσεις
44                 temp1=False #Εάνουμε το temp1 και το temp2 False ώστε να τα χρησιμοποιήσουμε ξανά στον υπολοίπο κώδικα
45                 temp2=False
46             if(start=True): #Εάν το start είναι True
47                 if(a_pos[0][i]==goal[i] and best_path[i-1]=='A'): #Εάν το νουκλεοτίδιο της a_pos ισούται με το (ζητούμενο νουκλεοτίδιο) και το best_path ισούται με την πρώτη κατάσταση(A)
48                     temp_a.append(round(from_a_to_a_and_b[0]*a_pos[0][i]*temp_a[i-1],10)) #H temp_a παίρνει την πιθανότητα να παει από το A στο A πολλαπλασιάζοντας την με την πιθανότητα
49                     #του (ζητούμενου νουκλεοτιδίου) και το temp_a της προηγούμενης κατάστασης
50                 temp1=True
51                 elif(b_pos[0][i]==goal[i] and best_path[i-1]=='B'): #Αλλιώς εάν το νουκλεοτίδιο της b_pos ισούται με το (ζητούμενο νουκλεοτίδιο) και το best_path ισούται με την δεύτερη κατάσταση(B)
52                     temp_b.append(round(from_b_to_a_and_b[0]*b_pos[0][i]*temp_b[i-1],10)) #H temp_b παίρνει την πιθανότητα να παει από το B στο B πολλαπλασιάζοντας την με την πιθανότητα
53                     #του (ζητούμενου νουκλεοτιδίου) και το temp_b της προηγούμενης κατάστασης
54
55                 temp1=False
56                 if(b_pos[0][i]==goal[i] and best_path[i-1]=='A'): #Εάν το νουκλεοτίδιο της b_pos ισούται με το (ζητούμενο νουκλεοτίδιο) και το best_path ισούται με την πρώτη κατάσταση(A)
57                     temp_b.append(round(from_a_to_a_and_b[1]*b_pos[0][i]*temp_b[i-1],10)) #H temp_b παίρνει την πιθανότητα να παει από το A στο B πολλαπλασιάζοντας την με την πιθανότητα
58                     #του (ζητούμενου νουκλεοτιδίου) και το temp_b της προηγούμενης κατάστασης
59                 temp2=True
60                 elif(b_pos[0][i]==goal[i] and best_path[i-1]=='B'): #Αλλιώς εάν το νουκλεοτίδιο της b_pos ισούται με το (ζητούμενο νουκλεοτίδιο) και το best_path ισούται με την δεύτερη κατάσταση(B)
61                     temp_b.append(round(from_b_to_a_and_b[1]*b_pos[0][i]*temp_b[i-1],10)) #H temp_b παίρνει την πιθανότητα να παει από το B στο B πολλαπλασιάζοντας την με την πιθανότητα
62                     #του (ζητούμενου νουκλεοτιδίου) και το temp_b της προηγούμενης κατάστασης
63                 temp2=False
64             if(temp1=True and temp2=False): #Εάν το temp1 και το temp2 είναι True
65                 if(temp_a[i]>temp_b[i]): #Εάν το temp_a είναι μεγαλύτερο από το temp_b
66                     best_path.append(two_states[0]) #Τότε το best_path παίρνει την πρώτη κατάσταση
67                     log_num=ln(temp_a[i]) #Μετατρέπεται τις αριθμητικές τιμές σε λογαριθμικές και τις προσθέτει σε μια μεταβλητή log_num
68                 else:
69                     best_path.append(two_states[1]) #Αλλιώς το best_path παίρνει την δεύτερη κατάσταση
70                     log_num=ln(temp_b[i]) #Μετατρέπεται τις αριθμητικές τιμές σε λογαριθμικές και τις προσθέτει σε μια μεταβλητή log_num
71                 i+=1
72             if(i==4 and len(best_path)==4): #Εάν το i έχει γίνει 4 και το best_path έχει 4 τιμές τότε
73                 temp=False #Το temp γίνεται false και όταν τελειώσει η while βγαίνει από αυτήν
74                 for i in range(0,len(best_path)): #Εκεί που έχουμε τις τιμές της A και της B καταστάσεις καθώς και το μοναδικό με την μεγαλύτερη πιθανότητα
75                     print('to A einai',temp_a[i], 'kai to B einai', temp_b[i])
76                 print('to best path einai', best_path, 'kai to logarithmiko athroisma tou best_path me stroggulopoihsh einai', round(log_num,4)) #Εκτυπώνει το καλύτερο μοναδικό και το άθροισμα των λογαριθμικών άθροισμα
77                 break #Αφού έχουν εκτιμηθεί όλα κάνει break, βγαίνει από την for και αφού το temp έχει γίνει False τελειώνει και η while
78
79 algo(a_pos,from_a_to_a_and_b,goal,start_possibility,b_pos,from_b_to_b_and_a,two_states) #Καλείται η algo

```

Και η εκτέλεση του προγράμματος

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

('to A einai', 0.2, 'kai to B einai', 0.1)
('to A einai', 0.072, 'kai to B einai', 0.004)
('to A einai', 0.00648, 'kai to B einai', 0.00216)
('to A einai', 0.0005832, 'kai to B einai', 0.0001944)
('to best path einai', ['A', 'A', 'A', 'A'], 'kai to logarithmiko athroisma tou best_path me stroggulopoihsh einai', -16.7265)
PS C:\Users\vagel\OneDrive\Έγγραφα\GitHub\software-engineering>

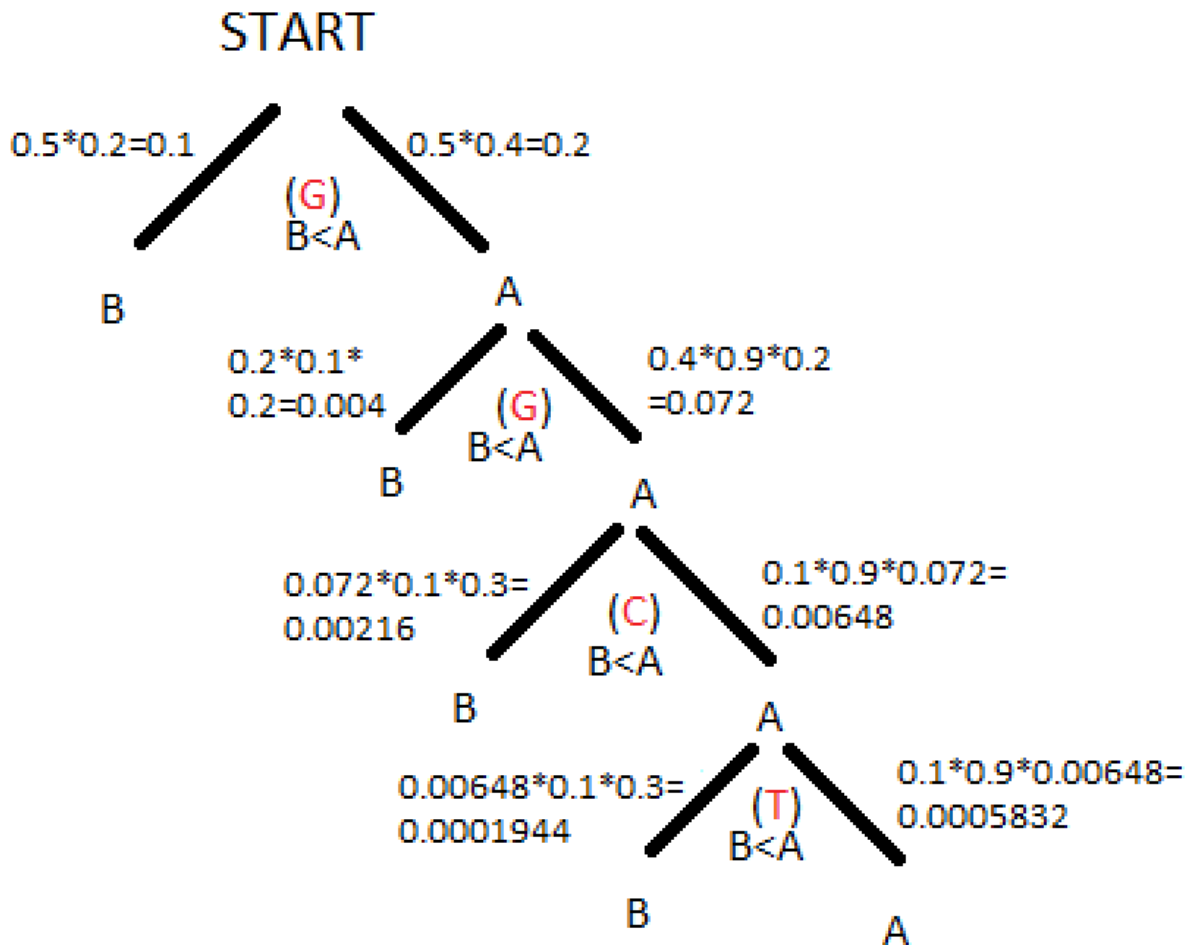
```

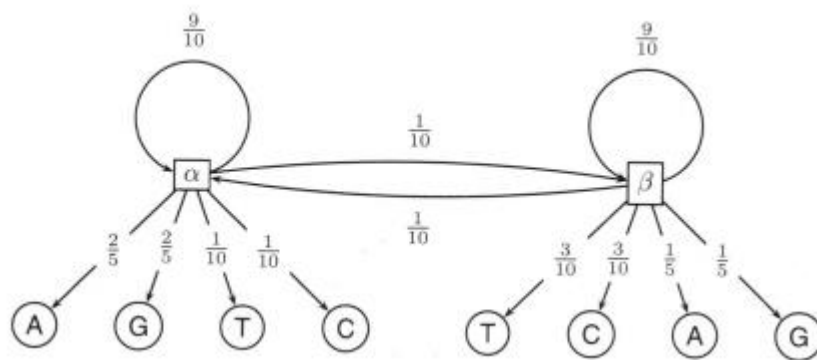
Το αρχείο μπορεί να βρεθεί και [εδώ](#) συνοδευόμενο με όλα τα απαραίτητα σχόλια.



2.2 Συνοπτική επεξήγηση

Σκοπός της εργασίας η δημιουργία ενός προγράμματος το οποίο έχει ως σκοπό να εκτελεί τον viterbi αλγόριθμο και να βρίσκει την πιθανότερη διαδρομή που θα ακολουθήσουν οι δυο καταστάσεις (α/β) για την αλληλουχία GGCT. Το πρόγραμμα παίρνει ως αρχικό input την πιθανότητα να εκπέμψει πουρίνες και πυριμιδίνες στις δύο καταστάσεις(α/β), την πιθανότητα να συνεχίσει από την κατάσταση (α) στην κατάσταση (β) και αντιστοίχως, την αρχική πιθανότητα και την ζητούμενη αλληλουχία. Στην συνέχεια εκτελεί τον viterbi αλγόριθμο και εκτυπώνει το βέλτιστο μονοπάτι.





3 ΘΕΜΑ 3^ο

Πρόβλημα 6.13

Δύο 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 τη νικηφόρα στρατηγική για όλες τις τιμές των n και m .



3.1 Κώδικας της άσκησης

```
1 import random
2 from Bio import SeqIO
3 #φορτώνουμε τις 2 αλληλουχίες
4 for sequence1 in SeqIO.parse("liver.fasta", "fasta"):
5     xromo_liver = sequence1.seq
6 for sequence2 in SeqIO.parse("brain.fasta", "fasta"):
7     xromo_brain = sequence2.seq
8 xromo_liver = len(sequence1) #Βρίσκουμε το μέγεθος της πρώτης αλληλουχιάς
9 xromo_brain = len(sequence2) #Βρίσκουμε το μέγεθος της δεύτερης αλληλουχιάς
10 player = 1
11 #Εδώ ελέγχουμε αν κέρδισε κάποιος παίκτης
12 def check(player):
13     if(xromo_brain==0 or xromo_liver==0):
14         print("Νικητής είναι ο παίκτης ",player)
15 while(True): #Το παιχνίδι παίζει μέχρι να νικήσει ένας από τους δύο παίκτες
16     if(xromo_brain!=0 and xromo_liver!=0): #Αν δεν έχει μηδενιστεί καμία αλληλουχία μπαίνει στην if
17         print("Παίζει ο παίκτης νούμερο ",player, " \n θέλεις να σβήσεις από την πρώτη(liver) ή την" +
18             "δεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα")
19         if(xromo_brain<xromo_liver): #αν η αλληλουχία 1 είναι μικρότερη από την 2 τότε ο τυχαίος αριθμός θα είναι από το 1 μέχρι το μέγεθος της αλληλουχίας 1
20             smaller=xromo_brain
21         else:
22             smaller=xromo_liver #αν η αλληλουχία 2 είναι μικρότερη από την 1 τότε ο τυχαίος αριθμός θα είναι από το 1 μέχρι το μέγεθος της αλληλουχίας 2
23         randomremove = random.randint(1,smaller) #τυχαίος αριθμός από το 1 μέχρι τον αριθμό που ορίσαμε από πάνω
24         epilogh = random.randint(1,3) #auto play
25         print(epilogh) #auto play
26         #epilogh= int(input()) #Ο χρήστης δίνει στο πρόγραμμα το τι θέλει να κάνει
27         while(True): #Πρέπει μέχρι να δέχεται σωστό τύπο input από τον χρήστη
28             if(epilogh==1 or epilogh==2 or epilogh==3): #Αν η επιλογή είναι 1 ή 2 ή 3 βγαίνει από την while και συνεχίζει
29                 break
30             else:
31                 print("Διαθέσιμες επιλογές 1 ή 2 ή 3") #Αν η επιλογή δεν είναι 1 ή 2 ή 3 εκτυπώνονται οι διαθέσιμες επιλογές
32                 epilogh = int(input()) #Ο χρήστης δίνει στο πρόγραμμα το τι θέλει να κάνει
33         if(epilogh==1): #Αν επιλέξει την επιλογή 1
34             xromo_liver-=randomremove #Αφαιρούμε από την αλληλουχία 1 τον τυχαίο αριθμό randomremove
35             check(player) #Ελέγχουμε μήπως τελείωσε το παιχνίδι
36         elif(epilogh==2):#Αν επιλέξει την επιλογή 2
37             xromo_brain-=randomremove #Αφαιρούμε από την αλληλουχία 2 τον τυχαίο αριθμό randomremove
38             check(player) #Ελέγχουμε μήπως τελείωσε το παιχνίδι
39         elif(epilogh==3):#Αν επιλέξει την επιλογή 3
40             xromo_brain-=randomremove #Αφαιρούμε από την αλληλουχία 1 τον τυχαίο αριθμό randomremove και
41             xromo_liver-=randomremove #Αφαιρούμε από την αλληλουχία 2 τον τυχαίο αριθμό randomremove
42             check(player)
43
44         if(player==1): #Αλλαγή παιχτών
45             player=2 #Αν παίζει ο παίκτης 1 τώρα έχει σειρά ο παίκτης 2
46         elif(player==2): #Αλλαγή παιχτών
47             player=1 #Αν παίζει ο παίκτης 2 τώρα έχει σειρά ο παίκτης 1
48         else: #Αν δεν μπαί στην if
49             break; #Βγαίνει από την while
```

Και η εκτέλεση του προγράμματος

```
Παίζει ο παίκτης νούμερο 1
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Παίζει ο παίκτης νούμερο 2
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Παίζει ο παίκτης νούμερο 1
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
1
Παίζει ο παίκτης νούμερο 2
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Παίζει ο παίκτης νούμερο 1
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Παίζει ο παίκτης νούμερο 2
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
3
Παίζει ο παίκτης νούμερο 1
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Παίζει ο παίκτης νούμερο 2
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Παίζει ο παίκτης νούμερο 1
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
3
Παίζει ο παίκτης νούμερο 2
  Θέλεις να σβήσεις από την πρώτη(liver) ή τηνδεύτερη(brain) ή και από τις δυο αλληλουχίες επέλεξε την επιλογή 1 ή 2 ή 3 αντίστοιχα
2
Νικητής είναι ο παίκτης 2
PS D:\OneDrive - unipi.gr\Biopliroforikh\Ask3>
```

Το αρχείο μπορεί να βρεθεί και [εδώ](#) συνοδευόμενο με όλα τα απαραίτητα σχόλια.



3.2 Συνοπτική επεξήγηση

Το χρωμόσωμα liver έχει αρχικά 39314 νουκλεοτίδια και το χρωμόσωμα brain έχει αρχικά 49943 νουκλεοτίδια.

Πρώτα παίζει ο παίκτης 1 και επιλέγει αν θέλει να σβήσει από το πρώτο χρωμόσωμα ή από το δεύτερο ή και από τα 2 χρωμοσώματα.

Ύστερα παίζει ο παίκτης 2 και επιλέγει αντίστοιχα από πού θέλει να σβήσει τυχαίο αριθμό νουκλεοτιδίων.

Το παιχνίδι τελειώνει όταν ο αριθμός των νουκλεοτιδίων σε ένα από τα 2 χρωμοσώματα γίνει ίσος με 0

4 ΘΕΜΑ 4^ο

4.1 Ερώτημα 1^ο

- a) Δείτε τα στοιχεία που παρουσιάζονται στην πρωτεϊνική βάση δεδομένων και προσδιορίστε τη μέθοδο με την οποία έχει προσδιορισθεί η δομή του συμπλόκου;

Η μέθοδος με την οποία έχει προσδιορισθεί η δομή του συμπλόκου είναι η : X-RAY DIFFRACTION

- b) Ποιο το resolution (διακριτική ικανότητα) στο οποίο προσδιορίστηκε η δομή;

Η διακριτική ικανότητα της δομής είναι 1.77

- c) Παραθέστε το Ψηφιακό αναγνωριστικό (Digital Object Identifier, DOI) της σχετικής επιστημονικής δημοσίευσης

Το DOI της δημοσίευσης είναι αυτό [10.1016/j.cell.2021.02.033](https://doi.org/10.1016/j.cell.2021.02.033)

4.2 Ερώτημα 2^ο

- a) Πόσες διακριτές πρωτεϊνικές αλυσίδες (molecular entities, macromolecules) περιλαμβάνει η εν λόγω δομή;

Περιλαμβάνει 3 διακριτές πρωτεϊνικές αλυσίδες

- b) Για κάθε μια από αυτές σημειώστε το πλήθος των αμινοξέων (sequence length) 3

COVOX-269 Fab heavy chain : το πλήθος των αμινοξέων είναι 222

COVOX-269 fab light chain : το πλήθος των αμινοξέων είναι 215

Spike glycoprotein : το πλήθος των αμινοξέων είναι 205



- c) Πόσους ολιγοσακχαρίτες περιλαμβάνει η δομή του συμπλόκου;
Περιέχει έναν ολιγοσακχαρίτη τον 2-acetamido-2-deoxy-beta-D-glucopyranose-(1-4)-[alpha-L-fucopyranose-(1-6)]2-acetamido-2-deoxy-beta-D-glucopyranose
- d) Η δομή του συμπλόκου έχει ένα άτομο χλωρίου (Cl⁻). Παραθέστε την αλυσίδα την οποία ανήκει
Ανήκει στην αλυσίδα CA

4.3 Ερώτημα 3^ο

- 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 light chain) καθώς και των επιπλέον στοιχείων (non-standard residues) που εμφανίζονται στο αρχείο



Log ⏏ x

UCSF ChimeraX version: 1.2.5 (2021-05-24)
 © 2016-2021 Regents of the University of California. All rights reserved.
[How to cite UCSF ChimeraX](#)

open "D:/OneDrive - unipi.gr/Bioplirorikh/Ask4/7neh.pdb"
7neh.pdb title:
**Crystal structure of the receptor binding domain of sars-cov-2 spike glycoprotein
 In complex with covox-269 fab** [\[more info...\]](#)

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

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

- 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:332-527

Στη συνέχεια χρησιμοποιώντας τη γραμμή εργαλείων :

Actions -> colour -> all options

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

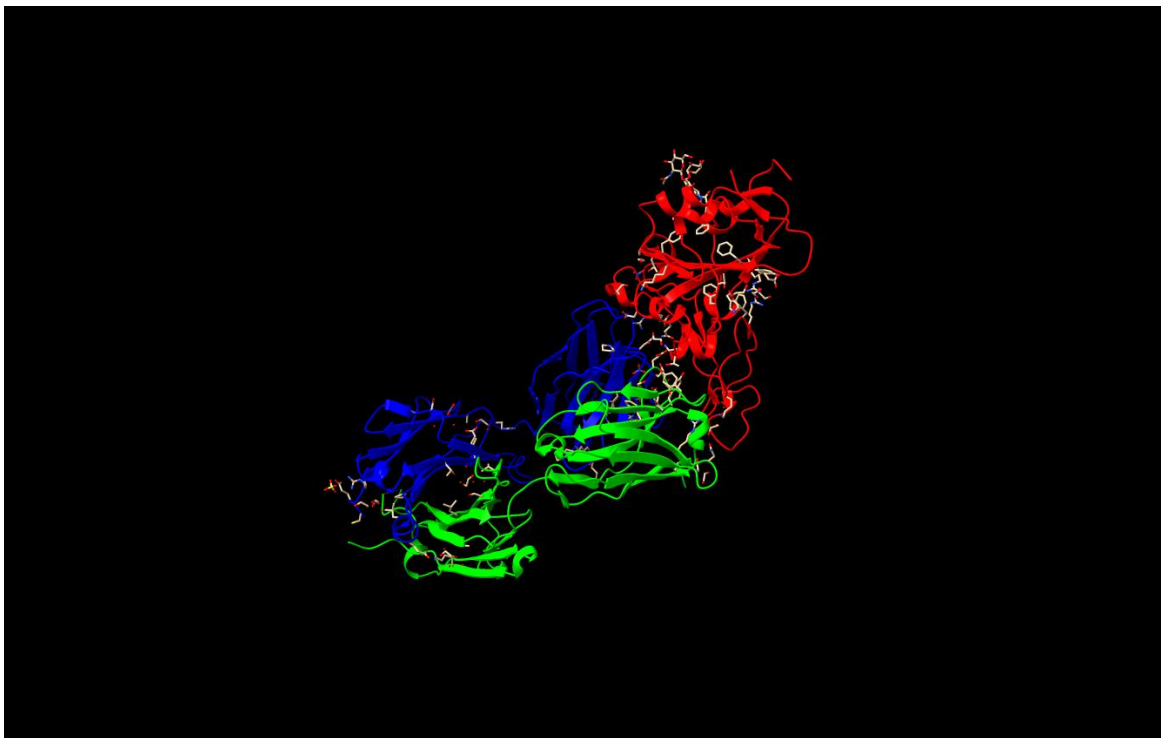


Select -> clear

Επαναλάβετε για τις υπόλοιπες πρωτεϊνικές αλυσίδες (βλ. ερώτημα 2α)

Αποθηκεύστε την εικόνα που δημιουργήσατε File -> save -> (επιλέξτε το format)

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



4.4 Ερώτημα 4^ο

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

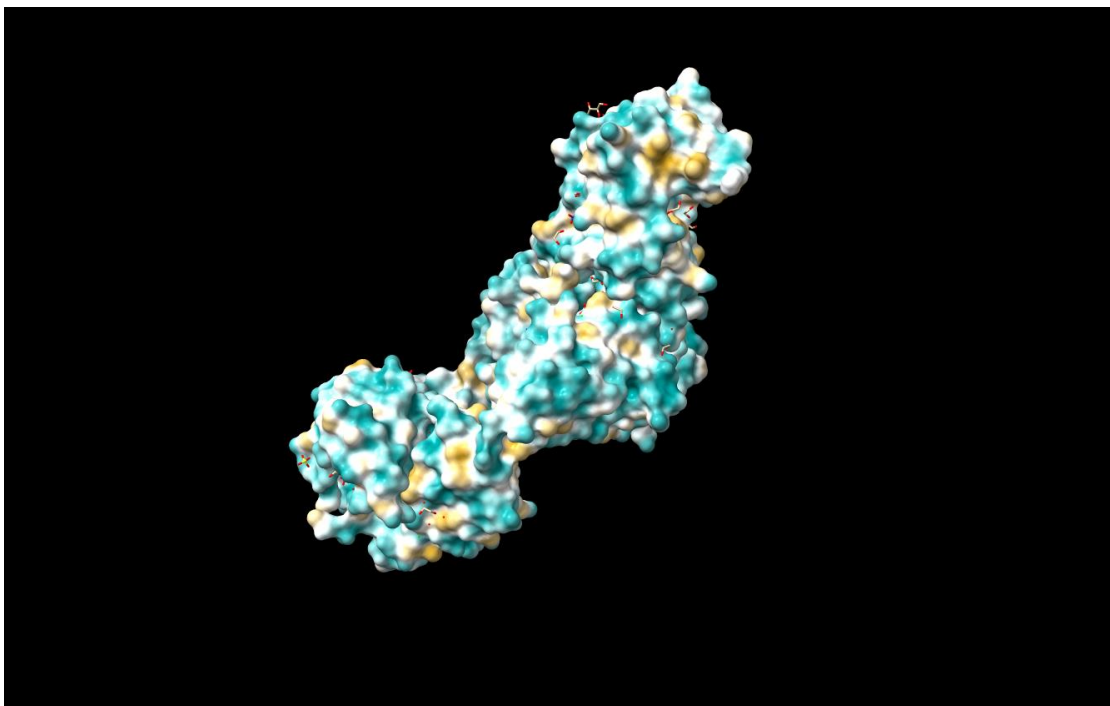
Command: select /E

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 E και για τις υπόλοιπες) Αποθηκεύστε την εικόνα που δημιουργήσατε

File-> save -> (επιλέξτε το format) Παρουσιάστε την εικόνα που δημιουργήσατε- Απάντηση για το (4A)



4.5 Ερώτημα 5°

- a) Σε συνέχεια του ερωτήματος 3: Επιλέξτε όπως και πριν 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 προσδεδεμένο στην πρωτεΐνη ακίδα.

Command: select :NAG



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

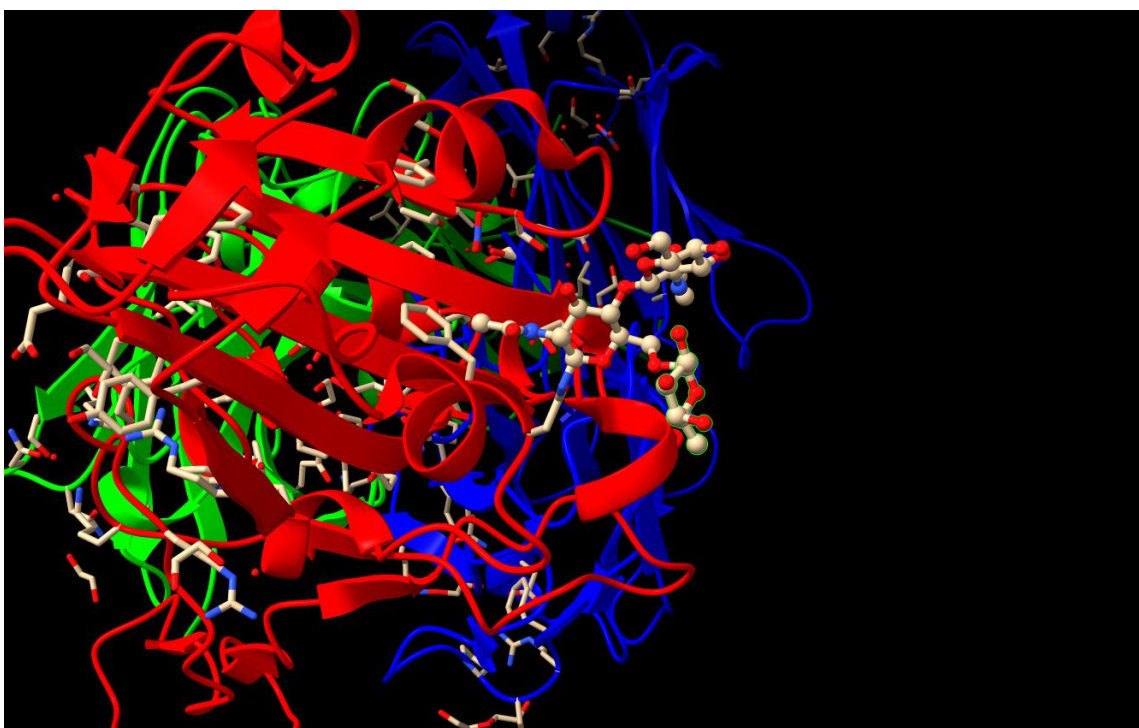
Molecule Display -> Ball and stick

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)

