Εργασία 1
Εισαγωγή στην βιοπληροφορική

Τζωρτζάκης Γρηγόρης αμ:1084538

Περιεχόμενα

Ο κώδικας μπορεί να βρεθεί εδώ:

https://github.com/GrigorisTzortzakis/Bioinformatics/tree/main/Exercise%201

Ερώτημα 1

i)

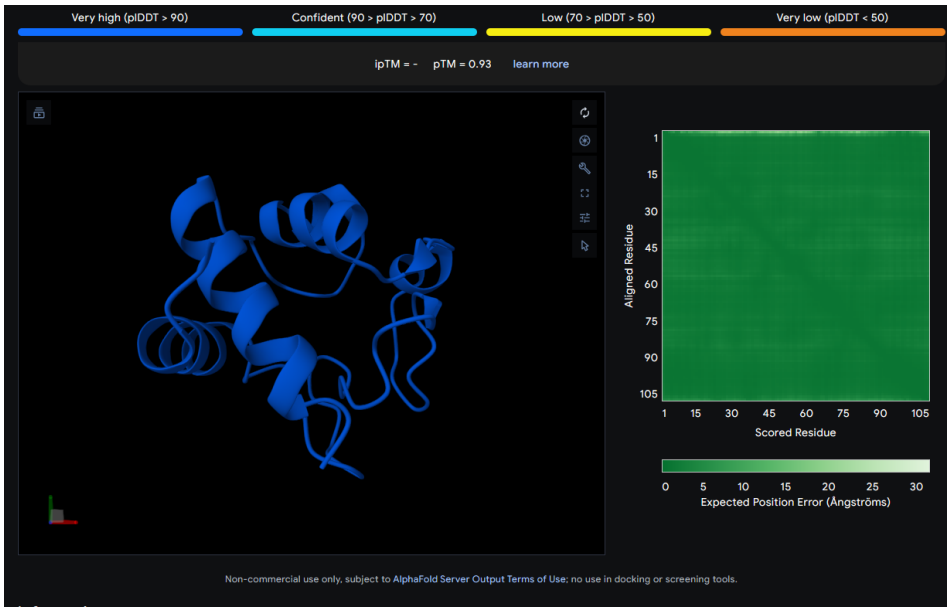
AlphaFold:

Αντιπροσωπεύει μια οικογένεια υπολογιστικών εργαλείων σχεδιασμένων να προβλέπουν τις τρισδιάστατες δομές των πρωτεϊνών και στην πιο πρόσφατη μορφή του, των βιομοριακών συμπλεγμάτων, απευθείας από αλληλουχίες αμινοξέων. Αναλυτικότερα, αξιοποιεί αρχιτεκτονικές βαθιάς μάθησης εκπαιδευμένες σε τεράστιες βάσεις δεδομένων γνωστών δομών πρωτεϊνών, προκειμένου να μάθει μοτίβα που συνδέουν την πληροφορία της αλληλουχίας με τις τρισδιάστατες διαμορφώσεις.

Η αρχική έκδοση του AlphaFold κυκλοφόρησε το 2021 και απέδειξε ασύλληπτη ακρίβεια στην πρόβλεψη δομών single chain πρωτεϊνών, επιλύοντας πολλά αναπάντητα ερωτήματα της μοριακής βιολογίας. Οι σχεδιαστές έκαναν διαθέσιμες τόσο τις λεπτομέρειες του αλγορίθμου όσο και μια πλήρως λειτουργική βάση κώδικα μέσω του GitHub (https://github.com/google-deepmind/alphafold) . Τότε, ερευνητές σε όλο τον κόσμο απέκτησαν πρόσβαση σε ένα εργαλείο που λαμβάνει ως είσοδο ένα αρχείο FASTA (αλληλουχία πρωτεΐνης), εκτελεί αναζήτηση multiple sequence alignment (MSA) σε μεγάλες βάσεις δεδομένων αλληλουχιών, επιλέγει δομικά πρότυπα όταν είναι διαθέσιμα και στη συνέχεια, παράγει προβλέψεις high confidence συντεταγμένες.

Προσθέτοντας, η google συνέχισε να εξελίσσει το AlphaFold. Τον Μάιο του 2024, το άρθρο στο Nature με τίτλο “Accurate structure prediction of biomolecular interactions with AlphaFold 3“ παρουσίασε ένα νέο μοντέλο που επεκτείνει τις δυνατότητες πρόβλεψης πέρα από μεμονωμένες πρωτεΐνες σε ολόκληρες βιομοριακές κατασκευές. Ενώ η προηγούμενη έκδοση εκπαιδεύτηκε ειδικά για την πρόβλεψη single chain πρωτεϊνών, το AlphaFold 3 υιοθετεί μια αρχιτεκτονική βασισμένη σε διάχυση (diffusion based neural architecture), ικανή να χειριστεί πρωτεΐνες που αλληλοεπιδρούν με νουκλεϊκά οξέα, μικρά μόρια, ιόντα και χημικά τροποποιημένα υπολείμματα.

Συνεχίζοντας, παρακάτω μπορουμε να δούμε ένα παράδειγμα χρήσης του εργαλείου με μια πρωτεΐνη:



Η εικόνα στα αριστερά είναι ένα τρισδιάστατο ribbon diagram του backbone της πρωτεΐνης μας. Σε αυτή την προβολή, η συνεχής μπλε κορδέλα ακολουθεί τη διαδρομή της πολυπεπτιδικής αλυσίδας από το N-τερματικό έως το C-τερματικό, δείχνοντας πώς τα αμινοξέα αναδιπλώνονται σε έλικες και βρόχους.

Η εικόνα στα δεξιά είναι ένας heatmap του Προβλεπόμενου Predicted Aligned Error (PAE), που αποτελεί ένα τετράγωνο πλέγμα όπου οι άξονες (οριζόντιος και κατακόρυφος) αντιστοιχούν στα κατάλοιπα της πρωτεΐνης με τη σειρά. Κάθε κελί σε αυτό το πλέγμα αναπαριστά πόσο σίγουρο είναι το μοντέλο για την απόσταση μεταξύ των δύο καταλοίπων που αντιστοιχούν στη συγκεκριμένη γραμμή και στήλη. Για παράδειγμα, το σκοτεινό πράσινο σημαίνει χαμηλή αναμενόμενη απόκλιση, δηλαδή ο αλγόριθμος έχει υψηλή εμπιστοσύνη ότι αυτά τα δύο έχουν σωστή σχετική θέση στο τελικό μοντέλο.

Τέλος, εκτός από την κύρια έκδοση, υπάρχει και η επιλογή να τρέξουμε το alphafold στο περιβάλλον google collab. Τότε, μπορουμε να αλλάξουμε διάφορες παραμέτρους και γενικότερα να δούμε αναλυτικά πως δουλεύει όλη η διαδικασία, η οποία δεν είναι ορατή στο κύριο πρόγραμμα για λόγους απλότητας. Επιπλέον, μπορουμε να βρούμε όλες τις εκδόσεις στο github καθώς και διαφορά add on που προσθέτουν επιπλέον λειτουργίες.

Esmfold:

Είναι ένα εργαλείο πρόβλεψης δομής πρωτεΐνης που χρησιμοποιεί ένα γλωσσικό μοντέλο πρωτεϊνών για να εξάγει τρισδιάστατες συντεταγμένες απευθείας από τις αλληλουχίες αμινοξέων, χωρίς να βασίζεται σε ευθυγραμμίσεις πολλαπλών αλληλουχιών (MSA). Αντί να ευθυγραμμίζει μια αλληλουχία με μεγάλες βάσεις δεδομένων για εξαγωγή πληροφοριών, το

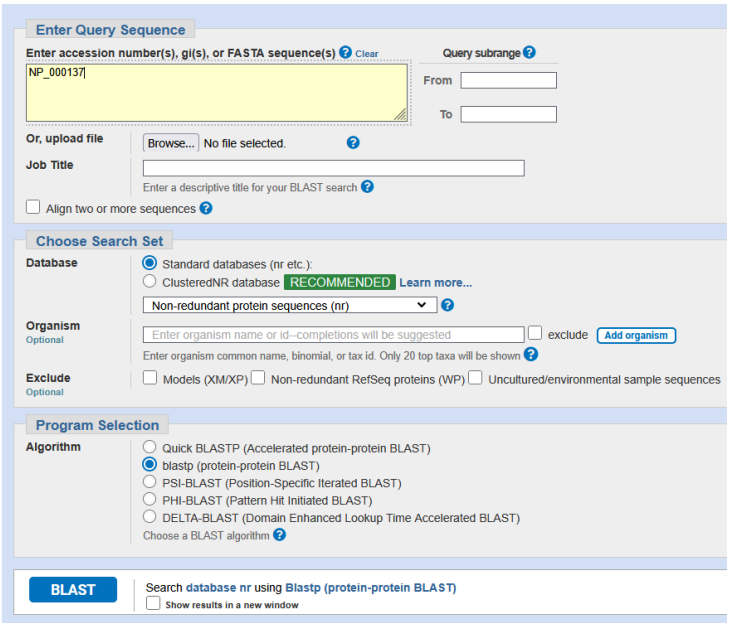
ESMFold επεξεργάζεται μια μεμονωμένη αλληλουχία μέσω ενός προ εκπαιδευμένου γλωσσικού μοντέλου, το οποίο έχει εκτεθεί σε δισεκατομμύρια αλληλουχίες πρωτεϊνών. Αυτή η προσέγγιση μειώνει δραστικά το υπολογιστικό φορτίο, διατηρώντας παράλληλα ακρίβεια συγκρίσιμη με τις παραδοσιακές μεθόδους που χρησιμοποιούν MSA.

Προσθέτοντας, το ESMFold αποτελείται από δύο κύρια συστατικά. Το πρώτο είναι ένα γλωσσικό μοντέλο πρωτεΐνης μεγάλης κλίμακας εκπαιδευμένο σε δεδομένα αλληλουχιών και το δεύτερο είναι ένα δομικό υποσύστημα που μετατρέπει τις υψηλής διάστασης ενσωματώσεις (embeddings) που παράγονται από το γλωσσικό μοντέλο σε ατομικές συντεταγμένες. Αναλύοντας, το γλωσσικό μοντέλο εκπαιδεύεται χρησιμοποιώντας τον στόχο της απόκρυψης αμινοξέων (masked language modeling) σε μια βάση δεδομένων ποικίλων αλληλουχιών. Στη συνέχεια, το δομικό υποσύστημα βελτιώνει αυτές τις ενσωματώσεις μέσω μιας σειράς γραφικών και γεωμετρικών μετασχηματισμών, με σκοπό την πρόβλεψη των θέσεων των ατόμων της κορυφής (backbone) και των πλευρικών αλυσίδων.

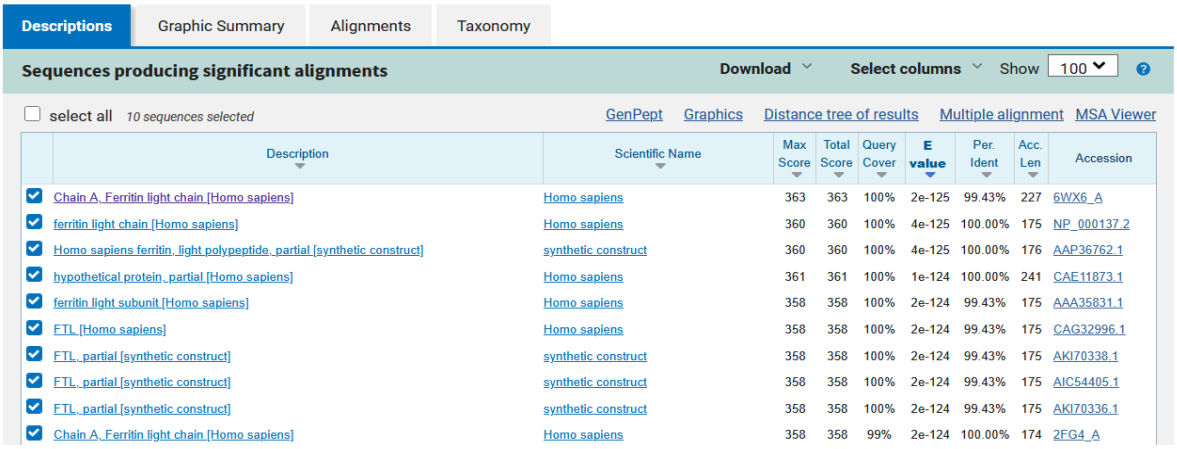
Δεδομένου ότι λειτουργεί σε μεμονωμένη αλληλουχία, το ESMFold προσφέρει σημαντικά πλεονεκτήματα ως προς την ταχύτητα σε σχέση με μεθόδους που εξαρτώνται από MSA. Σε δοκιμές, το ESMFold προβλέπει δομές έως έξι φορές πιο γρήγορα από το AlphaFold2, με ελάχιστη θυσία στην ακρίβεια για καλά μελετημένες πρωτεΐνες.

ii)

Το BLASTP είναι μια υπολογιστική μέθοδος που χρησιμοποιείται για τη σύγκριση μιας αλληλουχίας αμινοξέος έναντι μιας βάσης δεδομένων πρωτεϊνών, προκειμένου να εντοπιστούν παρόμοιες αλληλουχίες.



Από τα αποτελέσματα διαλέγουμε τα 10 πρώτα και αποθηκεύουμε το fasta:



Descriptions | Graphic Summary | Alignments | Taxonomy

Sequences producing significant alignments — Download — Select columns — Show 100

select all 10 sequences selected — GenPept | Graphics | Distance tree of results | Multiple alignment | MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Chain A, Ferritin light chain [Homo sapiens]	Homo sapiens	363	363	100%	2e-125	99.43%	227	6WX6_A
ferritin light chain [Homo sapiens]	Homo sapiens	360	360	100%	4e-125	100.00%	175	NP_000137.2
Homo sapiens ferritin, light polypeptide, partial [synthetic construct]	synthetic construct	360	360	100%	4e-125	100.00%	176	AAP36762.1
hypothetical protein, partial [Homo sapiens]	Homo sapiens	361	361	100%	1e-124	100.00%	241	CAE11873.1
ferritin light subunit [Homo sapiens]	Homo sapiens	358	358	100%	2e-124	99.43%	175	AAA35831.1
FTL [Homo sapiens]	Homo sapiens	358	358	100%	2e-124	99.43%	175	CAG32996.1
FTL, partial [synthetic construct]	synthetic construct	358	358	100%	2e-124	99.43%	175	AKI70338.1
FTL, partial [synthetic construct]	synthetic construct	358	358	100%	2e-124	99.43%	175	AIC54405.1
FTL, partial [synthetic construct]	synthetic construct	358	358	100%	2e-124	99.43%	175	AKI70336.1
Chain A, Ferritin light chain [Homo sapiens]	Homo sapiens	358	358	99%	2e-124	100.00%	174	2FG4_A

iii)

Ο MSA γίνεται για να στοιχίσει σχετικές ακολουθίες και να αναγνωρίσει ποιες θέσεις είναι διατηρημένες (αμετάβλητες) και ποιες μεταβάλλονται. Με αυτόν τον τρόπο, μπορούμε να συμπεράνουμε σχέσεις μεταξύ των ακολουθιών. Τα εργαλεία MAFFT, MUSCLE και T-Coffee αυτοματοποιούν αυτή τη διαδικασία, παίρνοντας ένα σύνολο ακολουθιών σε μορφή FASTA και παράγοντας μια λίστα που δείχνει πού συμφωνούν οι θέσεις σε όλες τις εισερχόμενες ακολουθίες.

Το MAFFT στοιχίζει πολλαπλές ακολουθίες πολύ γρήγορα. Με απλά λόγια, βρίσκει τμήματα που μοιάζουν μεταξύ όλων των ακολουθιών και τα ράβει μαζί γρήγορα. Το MUSCLE επίσης στοιχίζει ακολουθίες αλλά χρησιμοποιεί μια διαδικασία όπου ξεκινά με μια πρόχειρη εκδοχή και μετά βελτιώνει αυτή την εκδοχή βήμα βήμα. Τέλος, το T-Coffee λειτουργεί συγκρίνοντας πρώτα κάθε ζευγάρι ακολουθιών και δημιουργώντας μια "βιβλιοθήκη" από αυτές τις διμερείς αντιστοιχίσεις. Στη συνέχεια, χρησιμοποιεί αυτή τη βιβλιοθήκη για να φτιάξει την τελική ευθυγράμμιση. Ουσιαστικά, αφιερώνει περισσότερο χρόνο για να βεβαιωθεί ότι όλα τα μικρά ζευγάρια συμφωνούν πριν ενώσει τα πάντα μαζί.

Προχωρώντας στα αποτελέσματα, οι τρεις μέθοδοι στοίχισης διαφέρουν τόσο στο συνολικό μήκος όσο και στον τρόπο τοποθέτησης των κενών. Το MAFFT παρήγαγε μία στοίχιση 267 στηλών με συνολικά 802 χαρακτήρες κενών κατανεμημένους στις δέκα αλληλουχίες. Το T-COFFEE επίσης δημιούργησε 267 στήλες και τον ίδιο συνολικό αριθμό κενών, αλλά εάν τα συγκρίνουμε στήλη προς στήλη θα διαπιστώσουμε ότι σε ορισμένα σημεία τα αμινοξέα/βάσεις και τα κενά («–») έχουν μετατοπιστεί ελαφρώς, γεγονός το οποίο σημαίνει ότι, παρόλο που συμφωνούν ως προς το πόσα κενά θα εισάγουν, δεν τα τοποθετούν πάντα στις ίδιες ακριβώς θέσεις. Αντίθετα, η

στοίχιση του MUSCLE έχει μήκος μόλις 261 στήλες (έξι λιγότερες) και περιλαμβάνει συνολικά 742 κενά, δηλαδή δίνει μία πιο σύντομη στοίχιση με λιγότερα κενά, αποκόπτοντας κάποιες αμφιλεγόμενες περιοχές.

Για να αποφασίσουμε ποια στοίχιση είναι πιο ακριβής, μπορουμε να βασιστούμε στο ενσωματωμένο αριθμητικό σκορ κάθε προγράμματος. Ο MAFFT αναφέρει έναν Sum-of-Pairs σκορ που τιμωρεί έντονα τις ασυμφωνίες, ο T-COFFEE δίνει ένα consistency score βάσει του πόσο καλά η τελική πολλαπλή στοίχιση συμφωνεί με τη βιβλιοθήκη των αρχικών δευτερευουσών στοιχίσεων και ο MUSCLE παράγει το δικό του επαναληπτικό «σκορ» μετά τη βελτίωση. Στα αποτελέσματά μας, τόσο ο MAFFT όσο και ο T-COFFEE πέτυχαν σχεδόν ίδια σκορ, επειδή εισήγαγαν τον ίδιο αριθμό κενών σε περίπου τις ίδιες περιοχές, ενώ το σκορ του MUSCLE είναι κάπως χαμηλότερο από σχεδιασμό (συγχωνεύει ασαφείς στήλες).

Ερώτημα 2

Έστω S_1 και S_2 δύο συμβολοσειρές μήκους n και mαντίστοιχα και συμβολίσουμε με D(n,m) τη μικρότερη δυνατή «απόσταση επεξεργασίας» (edit distance) όταν η εισαγωγή και η διαγραφή έχουν κόστος 1, η αντικατάσταση έχει κόστος 2 και το ταίριασμα έχει κόστος 0. Ορίζουμε επίσης υως το μήκος μιας μεγαλύτερης κοινής υποακολουθίας (LCS) των S_1 και S_2. Θα δείξουμε ότι:

$$\boldsymbol{D(n,m) = n + m - 2u}$$

Αρχικά, έστω ότι L είναι κάποια κοινή υποακολουθία μήκους u. Μπορούμε να μετατρέψουμε τη συμβολοσειρά S_1 στην L διαγράφοντας κάθε ένα από τα n−u στοιχεία που δεν ανήκουν στην υποακολουθία, με συνολικό κόστος (n−u)·1=n−u. Κατόπιν, για να πάμε από την L στη S_2 , εισάγουμε κάθε ένα από τα m−u στοιχεία που λείπουν, με κόστος (m−u)·1=m−u. Όλες οι ταυτίσεις στοιχείων κοστίζουν 0 και δεν κάνουμε καθόλου αντικαταστάσεις. Συνεπώς, το συνολικό κόστος αυτού του σεναρίου επεξεργασίας είναι:

$$\boldsymbol{(n-u) + (m-u) = n + m - 2u}$$

Αντίστροφα, θεωρούμε ένα βέλτιστο σενάριο επεξεργασίας που μετασχηματίζει S_1 σε S_2 με ελάχιστο κόστος D(n,m). Κάθε αντικατάσταση στοιχείου κοστίζει 2, το οποίο δεν είναι καλύτερο από μία διαγραφή και μία εισαγωγή (κόστος 1+1=2), άρα χωρίς βλάβη στη βέλτιστη τιμή μπορούμε να υποθέσουμε ότι το σενάριο περιέχει μόνο διαγραφές, εισαγωγές και ταυτίσεις. Έστω ότι γίνονται α διαγραφές και β εισαγωγές, οπότε α+β=D(n,m). Επιπλέον, κάθε διαγραφή αφαιρεί ένα από τα n αρχικά στοιχεία και κάθε εισαγωγή προσθέτει ένα από τα m τελικά, οπότε ο αριθμός των

ταυτίσεων (δηλαδή των στοιχείων που παραμένουν κοινά) είναι u′=n−α=m−β. Λύνουμε το σύστημα:

$$\boldsymbol{\alpha + \beta = D(n,m), a = n - u', \beta = m - u'}$$

και βρίσκουμε:

$$\boldsymbol{D(n,m) = (n - u') + (m - u') = n + m - 2u'}$$

οπότε:

$$\boldsymbol{u' = \frac{n + m - D(n,m)}{2}}$$

Δεδομένου ότι u' είναι το πλήθος των ταυτίσεων σε ένα βέλτιστο σενάριο, πρέπει να είναι το μέγιστο δυνατό, δηλαδή u′=u. Άρα:

$$\boldsymbol{D(n,m) \geq n + m - 2u}$$

και συνεπώς ισχύει η ισότητα.

Συνεχίζοντας, ο κώδικας υλοποιεί δύο σύντομες συναρτήσεις. Η πρώτη υπολογίζει το μήκος και ανακαλύπτει μία κοινή υποακολουθία (LCS) δύο συμβολοσειρών μέσω ενός δυναμικού πίνακα, ενώ η δεύτερη μετατρέπει αυτό το μήκος σε σταθμισμένη απόσταση επεξεργασίας με κόστος «εισαγωγή = διαγραφή = 1, αντικατάσταση = 2, ταύτιση = 0». Όταν τις εφαρμόζουμε στις συμβολοσειρές AGCAT και GAC, βρίσκουμε ότι η μεγαλύτερη κοινή υποακολουθία έχει μήκος 2, για παράδειγμα την ακολουθία “AC".

Εφόσον το n=|AGCAT|=5, m=|GAC|=3 και η LCS έχει μήκος u=2, ο τύπος:

$$\boldsymbol{D = n + m - 2u}$$

μας δίνει 4. Αυτό σημαίνει ότι το ελάχιστο κόστος για να μετασχηματίσουμε την AGCAT στην GAC, επιτρέποντας μόνο διαγραφές και εισαγωγές στο κόστος 1 (και θεωρώντας κάθε αντικατάσταση ως διαγραφή+εισαγωγή με συνολικό κόστος 2), απαιτεί 4 βασικές πράξεις.

```
String 1: AGCAT
String 2: GAC
LCS length u = 2, one LCS = 'AC'
Edit distance D = n + m - 2u = 5 + 3 - 2*2 = 4
```

Ερώτημα 3

i)

```
Mounted at /content/drive
SARS-CoV-2 Spike length: 1273 aa
MERS-CoV    Spike length: 1483 aa
LCS length u = 592
Weighted edit distance D = 1273 + 1483 - 2*592 = 1572
One LCS (first 50 aa): MVFLLLPVSSCVTTRPAGYPRSTQLFPHVSGTGTKFVNDVFANRIGTTST
One LCS (last 50 aa):  KALNESIDLELGYYKWPWYIWLGFIAGLAVILCCTCCLKCCCDEDEPVVH
```

Οι πρωτεΐνες spike του SARS-CoV-2 και του MERS-CoV μοιράζονται μια μέγιστη κοινή υποακολουθία 592 αμινοξέων. Αυτό μας λέει ότι σχεδόν το ήμισυ του μήκους της μικρότερης πρωτεΐνης (1273 αμινοξέα για τον SARS-CoV-2) αποτελείται από την ίδια ordered σειρά αμινοξέων. Αυτός ο μεγάλος κοινός κορμός προκύπτει επειδή και οι δύο ιοί βασίζονται στον ίδιο θεμελιώδη «μηχανισμό ένωσης» που βρίσκεται στην υπομονάδα S_2, δηλαδή μια σειρά από επαναλαμβανόμενες heptad repeats, κεντρικές έλικες και συνδετικές περιοχές, οι οποίες υφίστανται δραματική διαμόρφωση για να οδηγήσουν τη συγχώνευση των μεμβρανών.

Αντίθετα, οι υπομονάδες S_1 που αναγνωρίζουν τους υποδοχείς έχουν εξελιχθεί ξεχωριστά (ACE2 έναντι DPP4) και έτσι εμφανίζουν πολύ λιγότερες ταυτίσεις στην ακολουθία των αμινοξέων. Ο αλγόριθμος LCS εντοπίζει αποτελεσματικά αυτές τις διατηρημένες περιοχές, παραλείποντας εισαγωγές, διαγραφές ή αποκλίνοντα loops και το αποτέλεσμα (592 παρόμοιες θέσεις με τη σωστή σειρά) ποσοτικοποιεί πόσο μεγάλο μέρος του μηχανισμού συγχώνευσης παραμένει αναλλοίωτο ανάμεσα στους δύο κορονοϊούς.

Επεκτείνοντας, όταν μετατρέπουμε το LCS σε weighted edit distance, χρησιμοποιώντας κόστος 1 για κενά (εισαγωγές ή διαγραφές) και 2 για αντικαταστάσεις, λαμβάνουμε μια τιμή 1572 λειτουργιών που απαιτούνται για να μετατρέψουμε τη μία πρωτεΐνη spike στην άλλη. Αυτός ο μεγάλος αριθμός αντικατοπτρίζει τόσο το μήκος κάθε πρωτεΐνης, όσο και τις διαφορές στην ακολουθία που συγκεντρώνονται κυρίως στην περιοχή S_1.

Επιπλέον, τα αποκλίνοντα αμινοξέα (1273 – 592 = 681 μοναδικά για τον SARS-CoV-2 και τα 1483 – 592 = 891 μοναδικά για τον MERS-CoV), εξηγούν γιατί τα εμβόλια και τα αντισώματα που είναι ιδιαίτερα αποτελεσματικά κατά του ενός ιού έχουν περιορισμένη προστασία κατά του άλλου.

ii)

Επιλέγουμε τα καλυτέρα μοντέλα από τα 6 που μας βγάζει (αυτό με τις καλύτερες μετρικές) και τα βάζουμε στο dali Pairwise Structure Comparison.

Summary

No:	Chain	Z	rmsd	lali	nres	%id	PDB	Description
1:	t001-G	35.6	7.0	932	1149	28	PDB	

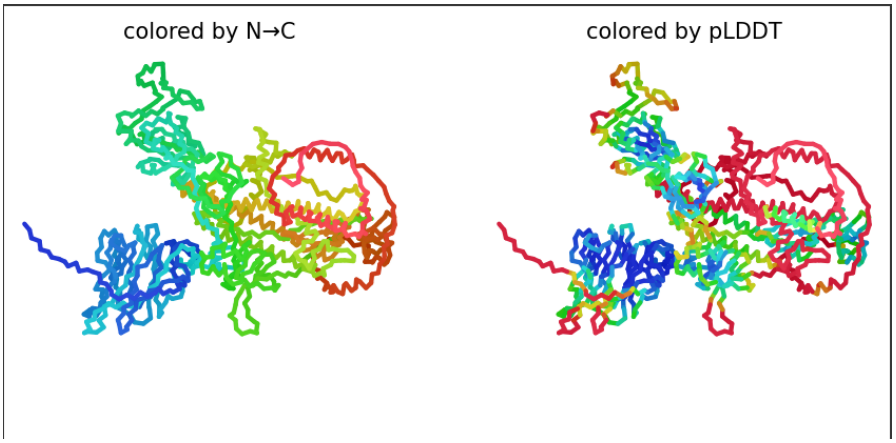
Η σύγκριση αποκαλύπτει μια εντυπωσιακή δομική σχέση ανάμεσα στις πρωτεΐνες, παρά την χαμηλή ομοιότητα στις αλληλουχίες τους. Πρώτον, το Z-score των 35,6 είναι εξαιρετικά υψηλό (πολύ πάνω από το όριο ~2), αποδεικνύοντας ότι οι δύο αυτές πρωτεΐνες μοιράζονται το ίδιο γενικό τρισδιάστατο τύπωμα (fold).

Δεύτερον, από τις περίπου 1 200–1 400 θέσεις αμινοξέων σε κάθε Spike, το DALI ευθυγράμμισε 932 αμινοξέα σε έναν κοινό δομικό πυρήνα. Αυτό το μήκος ευθυγράμμισης αντιστοιχεί κυρίως στον κεντρικό έλικα και στην υπομονάδα ένωσης, που πρέπει να διατηρούν το σχήμα τους για να επιτελέσουν τη συγχώνευση της μεμβράνης. Μέσα σε αυτόν τον πυρήνα, μόλις το 28% των θέσεων είναι παρόμοιες σε επίπεδο αλληλουχίας, υπογραμμίζοντας την εκτεταμένη διαφορά (εισαγωγές, διαγραφές και αντικαταστάσεις) γύρω από ένα διατηρημένο σκελετό. Το μεγάλο μήκος ευθυγράμμισης μαζί με τη χαμηλή ομοιότητα δείχνει πώς οι πρωτεΐνες μπορούν να διατηρούν κρίσιμες λειτουργικές αναδιπλώσεις ακόμα και όταν οι αλληλουχίες τους διαφοροποιούνται.

Τέλος, η ρίζα του μέσου τετραγώνου απόκλισης (RMSD) (7) στον ευθυγραμμισμένο πυρήνα αντανακλά μέτρια παραλλαγή του σκελετού.

iii)

Βάζουμε το alphafold στο google collab για ένα σύντομο χρονικό διάστημα καθώς απαιτεί πολύ χρόνο για την ολοκλήρωση και έχουμε:



Αντίστοιχα κάνουμε και για esmfold.

Ερώτημα 4

Πρώτη επιλογή

Σε πολλές εφαρμογές, λαμβάνουμε k ακολουθίες $S_1, S_2, ..., S_k$ και επιδιώκουμε να βρούμε κάθε μη κενή συμβολοσειρά X που εμφανίζεται τουλάχιστον δύο φορές σε καθεμία από τις S_i. Μια επιπλέον προϋπόθεση είναι ότι, μέσα σε κάθε S_i, οι δύο εμφανίσεις της X πρέπει να απέχουν το πολύ μια δοσμένη απόσταση "κενό" Δ_i (δηλαδή, η διαφορά του τέλους της πρώτης εμφάνισης από την έναρξη της δεύτερης δεν υπερβαίνει το Δ_i).

Στην αρχή, θα μπορούσαμε να δοκιμάσουμε την απευθείας καταμέτρηση όλων των υποσυμβολοσειρών μιας από τις ακολουθίες και να ελέγξουμε τη συχνότητά τους στις υπόλοιπες. Όμως, ο αριθμός τους είναι τετραγωνικός ως προς το μήκος μιας ακολουθίας, καθιστώντας αυτή την άμεση μέθοδο ακατάλληλη ακόμη και για μέτρια μεγέθη δεδομένων.

Μια πιο δομημένη λύση επεκτείνει την κλασική προσέγγιση του δένδρου επιθεμάτων σε πολλαπλές συμβολοσειρές. Κατασκευάζουμε ένα GST πάνω στην ένωση:

$$\boldsymbol{T = S_1\$_1 S_2\$_2 \dots S_k\$_k}$$

όπου κάθε $\$_i$ είναι μοναδικός τερματικός χαρακτήρας. Σε αυτό το δένδρο, κάθε εσωτερικός κόμβος αντιστοιχεί σε μια συμβολοσειρά X που εμφανίζεται σε μία ή περισσότερες από τις S_i. Αναθέτοντας σε κάθε φύλλο τον δείκτη της ακολουθίας και τη θέση εκκίνησης και προωθώντας αυτές τις

πληροφορίες προς τα πάνω, κάθε εσωτερικός κόμβος συγκεντρώνει για κάθε ακολουθία μια ταξινομημένη λίστα θέσεων όπου εμφανίζεται η *X*.

Φιλτράρουμε τους κόμβους ώστε σε κάθε λίστα θέσεων να υπάρχουν τουλάχιστον δύο καταχωρίσεις. Έτσι, βρίσκουμε όλες τις συμβολοσειρές που επαναλαμβάνονται σε κάθε ακολουθία. Για να εφαρμόσουμε και τον περιορισμό του "κενό" σε κάθε S_i, σαρώνουμε τη λίστα με ένα παράθυρο πλάτους $\Delta_i + |X|$ και αν υπάρχουν δύο θέσεις εντός αυτού του παραθύρου, η συνθήκη ικανοποιείται. Αναφερόμαστε μόνο στους κόμβους που περνούν τον έλεγχο για όλες τις k ακολουθίες. Η κατασκευή του GST γίνεται σε O(Ntotal) χρόνο (μέσω του αλγορίθμου Ukkonen ή McCreight), ενώ η σάρωση των λιστών θέσεων προσθέτει γραμμική πολυπλοκότητα ως προς τον συνολικό αριθμό εμφανίσεων.

Παρά την κομψότητά του, το GST μπορεί να απαιτεί μεγάλο χώρο μνήμης όταν k ή οι ακολουθίες είναι πολύ μεγάλες. Για την περίπτωση k = 2, οι Brodal, Lyngsø, Pedersen και Stoye (CPM 1999) παρουσίασαν έναν αλγόριθμο που βρίσκει μεγαλύτερα ζεύγη (maximal pairs) με περιορισμένο κενό σε πραγματικά γραμμικό χρόνο. Ενισχύουν το δέντρο επιθεμάτων των δύο συμβολοσειρών με λίστες διαστημάτων και δείκτες και εκτελούν μια κατώ-προς-άνω (bottom-up) διάσχιση που συγχωνεύει τις λίστες εμφανίσεων σε $O(n_1 + n_2 + occ)$ χρόνο, όπου occ είναι ο αριθμός των υποψήφιων επαναλήψεων. Σε κάθε κόμβο, ελέγχουν αποδοτικά αν υπάρχουν δύο εμφανίσεις σε κάθε συμβολοσειρά με απόσταση $\leq \Delta_i$. Αν ναι, η συμβολοσειρά αναφέρεται ως έγκυρη επανάληψη. Η ιδέα της k-δρομικής συγχώνευσης μπορεί να επεκταθεί σε k > 2 με κατάλληλη χρήση πολλαπλών δεικτών ή ουρών προτεραιότητας, διατηρώντας τη γραμμική αποδοτικότητα.

Συνεχίζοντας, οι Brodal & Pedersen (CPM 2000) μελέτησαν τις maximal quasiperiodicities, δηλαδή επαναλήψεις που επιτρέπεται να επικαλύπτονται υπό ελεγχόμενο τρόπο. Εισάγουν δομές δεδομένων που παρακολουθούν ταυτόχρονα τους περιορισμούς κενών και επικάλυψης. Μια υποψήφια συμβολοσειρά παραμένει "ενεργή" μόνο εφόσον μπορεί να επεκταθεί αριστερά ή δεξιά χωρίς να παραβιαστεί η ελάχιστη απόσταση. Διατηρώντας στοίβες κόμβων και στοίβες διαστημάτων θέσεων, διασφαλίζουν ότι στο τέλος της διάσχισης κάθε αναφερόμενη επανάληψη είναι μεγίστη (δεν μπορεί να επεκταθεί περαιτέρω σε όλες τις ακολουθίες ταυτοχρόνως).

Για γενικό k, οι Bakalis, Iliopoulos, Makris, Sioutas, Theodoridis, Tsakalidis & Tsichlas (Comput. J. 2007) προτείνουν δύο συμπληρωματικές μεθόδους. Πρώτον, εφαρμόζουν την προσέγγιση GST με k-δρομικές σαρώσεις παραθύρου που τερματίζουν νωρίς μόλις μια ακολουθία αποτύχει στον

έλεγχο. Αυτό αποδίδει αλγόριθμο O(Ntotal + occ · log N_max), όπου Nmax είναι το μήκος της μεγαλύτερης συμβολοσειράς. Δεύτερον, προτείνουν τεχνική κατακερματισμού k-mers, δηλαδή επιλέγοντας μήκος σπόρου ℓ, καταμετρούν όλα τα ℓ-mers κάθε S_i, κρατούν μόνο όσα εμφανίζονται ≥ 2 φορές σε κάθε ακολουθία και για κάθε τέτοιο σπόρο επιχειρούν αριστερή/δεξιά επέκταση χαρακτήρα-προς-χαρακτήρα όσο ικανοποιούνται οι περιορισμοί. Παρά την τετραγωνική χειρότερη περίπτωση, σε πρακτικά δεδομένα, όπως γονιδιωματικά, η μέθοδος αυτή ξεπερνά συχνά το GST λόγω μικρότερου αποτυπώματος μνήμης και λιγότερων επιζώντων σπόρων.

Δεύτερη επιλογή

α)

Λύνουμε το πρόβλημα της καθολικής στοίχισης ξεκινώντας με δύο πίνακες μεγέθους (n+1)×(m+1) των οποίων τα κελιά καταγράφουν το βέλτιστο σκορ για κάθε ζεύγος προθεμάτων των δύο αλληλουχιών. Ο πρώτος πίνακας, που θα τον ονομάσουμε «στοίχιση ή κενό», αποθηκεύει το καλύτερο σκορ όταν η τελευταία πράξη ήταν είτε ένα ταίριασμα χαρακτήρων είτε η εισαγωγή κενών. Ο δεύτερος πίνακας, που θα τον ονομάσουμε «τρέχουσα αλυσίδα ασυμφωνιών», αποθηκεύει το καλύτερο σκορ όταν η τελευταία πράξη ήταν μια ασυμφωνία που ανήκει σε μια συνεχόμενη ακολουθία. Με αυτόν τον τρόπο, επιβάλλουμε σωστά ότι μια μεμονωμένη ασυμφωνία κοστίζει ρ+σ και ότι κάθε επόμενη συνεχόμενη ασυμφωνία κοστίζει μόνο σ, ενώ ταυτόχρονα επιτρέπουμε κενά αμέσως μετά το κλείσιμο οποιασδήποτε τρέχουσας αλυσίδας ασυμφωνιών.

Προχωρώντας, αρχικοποιούμε στον πίνακα 1 το κελί πάνω αριστερά σε μηδέν και το αντίστοιχο κελί του πίνακα 2 σε μείον άπειρο (μη εφικτό). Στην κορυφαία γραμμή και στην αριστερή στήλη του πίνακα 1 καταγράφουμε το κόστος εισαγωγής μόνο κενών, δηλαδή −jρ ή -iρ. Ο πίνακας ασυμφωνιών παραμένει αδύνατος κατά μήκος αυτών των ορίων.

Συνεχίζοντας, καθ' όλη τη διάρκεια των εσωτερικών κελιών, όταν οι δύο υπό εξέταση χαρακτήρες διαφέρουν, δεν επιτρέπεται απλό ταίριασμα ούτε αντιμετώπιση ως μεμονωμένο κενό. Θα πρέπει να δημιουργήσουμε ή να συνεχίσουμε μια αλυσίδα ασυμφωνιών και για να ανοίξουμε νέα αλυσίδα, μεταφερόμαστε από το διαγώνιο κελί του πίνακα 1 και αφαιρούμε ρ+σ. Για να επεκτείνουμε τρέχουσα αλυσίδα, μεταφερόμαστε από το διαγώνιο κελί του πίνακα 2 και αφαιρούμε μόνο σ.

Τέλος, όταν οι χαρακτήρες ταιριάζουν, επιτρέπονται δύο διαγώνιες μεταβάσεις στον πίνακα 1. Συγκεκριμένα, η πρώτη από το αντίστοιχο κελί του ίδιου πίνακα, προσθέτοντας +1 για ταίριασμα και η δεύτερη από το

αντίστοιχο κελί του πίνακα ασυμφωνιών, επίσης με +1, κλείνοντας έτσι την τρέχουσα αλυσίδα ασυμφωνιών. Επιπλέον, σε κάθε κελί, ανεξαρτήτως ταίριασματος, επιτρέπονται εισαγωγή ή διαγραφή κενών με κόστος −ρ ανά σύμβολο. Αυτά τα βήματα κενών μπορούν να προέλθουν είτε από τον πίνακα 1 είτε από τον πίνακα 2, διότι πριν από το κενό η τρέχουσα αλυσίδα ασυμφωνιών κλείνει χωρίς επιπλέον ποινή πέραν της φυσιολογικής ποινής κενών.

β)

Λύνουμε το πρόβλημα της εύρεσης του μέγιστου κοινού προθέματος για κάθε ζεύγος από τις k ακολουθίες μήκους n κατασκευάζοντας ένα ενιαίο προθεματικό δέντρο που περιέχει όλες τις εισόδους και στη συνέχεια, διασχίζοντας αυτό το δέντρο ώστε να αναφέρουμε κάθε ζεύγος που το μέγιστο κοινό του πρόθεμα αντιστοιχεί σε έναν συγκεκριμένο κόμβο. Σε ένα trie, κάθε κόμβος αντιπροσωπεύει τη συμβολοσειρά που σχηματίζεται από τις ετικέτες των ακμών από τη ρίζα ως τον κόμβο αυτόν και το υποδέντρο του περιέχει ακριβώς εκείνες τις εισόδους που μοιράζονται αυτό το πρόθεμα. Κατά συνέπεια, για κάθε δύο ακολουθίες, ο βαθύτερος κοινός τους πρόγονος στο trie κωδικοποιεί το μέγιστο κοινό τους πρόθεμα.

Αναλυτικότερα, για την κατασκευή του trie σε χρόνο O(k n), ξεκινάμε από μια κενή ρίζα και εισάγουμε μία μία όλες τις k συμβολοσειρές. Καθεμιά από αυτές διασχίζεται χαρακτήρα-χαρακτήρα από τη ρίζα, ακολουθώντας υπάρχουσες ακμές ή δημιουργώντας νέους κόμβους όταν δεν υπάρχει ήδη η αντίστοιχη μετάβαση. Στο φύλλο όπου ολοκληρώνεται η συμβολοσειρά, καταγράφουμε το αναγνωριστικό της. Δεδομένου ότι κάθε χαρακτήρας κάθε συμβολοσειράς επεξεργάζεται ακριβώς μία φορά, ο συνολικός χρόνος κατασκευής είναι O(k n). Επιπλέον, σε κάθε κόμβο διατηρούμε δομές (π.χ. πίνακα ή χάρτη κατακερματισμού) για τη διαχείριση των παιδιών του.

Τέλος, αφού κατασκευαστεί το trie, εκτελούμε μία μεταπήδηση (post–order) για να συγκεντρώσουμε σε κάθε κόμβο τη λίστα όλων των αναγνωριστικών συμβολοσειρών στο υποδέντρο του και παράλληλα, να αναφέρουμε κάθε ζεύγος των οποίων το μέγιστο κοινό πρόθεμα είναι η συμβολοσειρά του κόμβου. Συγκεκριμένα, όταν ολοκληρώνονται οι κλήσεις DFS για τα παιδιά ενός κόμβου, συγχωνεύουμε τις λίστες αναγνωριστικών, επιλέγοντας πάντα ως βάση τη μεγαλύτερη λίστα. Κάθε φορά που εισάγουμε ένα νέο αναγνωριστικό στη συσσωρευμένη λίστα, το ζευγαρώνουμε με όλα τα ήδη υπάρχοντα και εκτυπώνουμε αμέσως αυτό το ζεύγος μαζί με το πρόθεμα του τρέχοντος κόμβου. Εφόσον παράγουμε ακριβώς α τέτοια ζεύγη και κάθε αναγνωριστικό μετακινείται το πολύ σε nεπίπεδα, ο συνολικός χρόνος για συγχώνευση και αναφορά είναι O(k n+α).

Ερώτημα 5

i)

Φανταζόμαστε αυτόν τον πίνακα με 12 γραμμές (μία για κάθε μήκος πρόθεμα του v, συμπεριλαμβανομένου του κενού) και 11 στήλες (μία για κάθε μήκος πρόθεμα του w). Η πρώτη γραμμή και η πρώτη στήλη αντιστοιχούν στη στοίχιση ενός προθέματος της μίας αλληλουχίας με πλήρη κενά της άλλης, καθώς κάθε κενό επιφέρει ποινή –1, οι τιμές στην πρώτη γραμμή μειώνονται από 0 έως –10 και στην πρώτη στήλη από 0 έως –11 όταν προχωράμε δεξιά ή κάτω, αντίστοιχα.

Αφού αρχικοποιηθεί ο πίνακας, κάθε εσωτερικό κελί D[i][j] που αντιπροσωπεύει τη βέλτιστη βαθμολογία στοίχισης των πρώτων I γραμμάτων του v με τα πρώτα j γράμματα του w, υπολογίζεται λαμβάνοντας υπόψη τρεις δυνατές "κινήσεις" από τα γειτονικά κελιά. Μια διαγώνια μετακίνηση από το D[i−1][j−1] αντιστοιχεί στη στοίχιση του vi με το wj και αν οι χαρακτήρες ταυτίζονται, προσθέτουμε +1, ενώ αν όχι, προσθέτουμε –1. Μια κατακόρυφη μετακίνηση από το D[i−1][j] σημαίνει στοίχιση του vi με κενό (ποινή –1), ενώ μια οριζόντια μετακίνηση από το D[i][j−1] στοίχιση του wj με κενό (επίσης –1).

Με τη συστηματική συμπλήρωση του πίνακα, σειρά προς σειρά, το κάτω-δεξί κελί D[11][10] δίνει την τελική βαθμολογία της βέλτιστης παγκόσμιας στοίχισης ολόκληρου του v με ολόκληρο το w. Στην περίπτωσή μας, η τιμή αυτή είναι +2, δείχνοντας ότι μπορούμε να στοιχίσουμε τις δύο αλληλουχίες έτσι ώστε η διαφορά ανάμεσα στους συνολικούς βαθμούς των ταυτίσεων και των ποινών (ασυμφωνίες και κενά) να ισούται με δύο.

Για να ανακτήσουμε μία ρητή στοίχιση που επιτυγχάνει αυτή τη βαθμολογία, ακολουθούμε τον πίνακα ανάποδα (traceback) από το κάτω-δεξιά κελί μέχρι το κελί D[0][0]. Σε κάθε βήμα επιλέγουμε το προηγούμενο κελί που συνδυαστικά με την αντίστοιχη ποινή (ταύτιση/ασυμφωνία ή κενό) αναπαράγει την τρέχουσα τιμή. Μια τέτοια βέλτιστη στοίχιση είναι:

$$\boldsymbol{v}: \boldsymbol{TTAGTTAAGTG}$$

$$\boldsymbol{w}: \boldsymbol{-TTAG-TGAA-TT}$$

ii)

Φανταζόμαστε ένα πλέγμα όπου οι γραμμές αντιστοιχούν στα προθέματα της αλληλουχίας v (συμπεριλαμβανομένου του κεντρικού προθέματος) και οι στήλες στα προθέματα της αλληλουχίας w. Στην τοπική στοίχιση (Smith–Waterman), κάθε κελί αρχικοποιείται στο 0, ώστε να μην επιβάλλεται ποινή

για κενά πριν από την έναρξη της βέλτιστης υποαλυλουχίας. Έτσι η πρώτη γραμμή και η πρώτη στήλη γεμίζουν με μηδενικά.

Για κάθε εσωτερικό κελί (i,j), που αντιπροσωπεύει την καλύτερη βαθμολογία για υποαλυλουχίες που τελειώνουν στα vi και wj, εξετάζουμε τρεις τρόπους επέκτασης, δηλαδή ταίριασμα ή λάθος ταίριασμα των δύο χαρακτήρων (προσθέτουμε +1 ή –1) και εισαγωγή κενού σε μία από τις αλληλουχίες (–1). Κρίσιμο στοιχείο είναι ότι αν όλοι οι τρεις υπολογισμοί δώσουν αρνητικό αποτέλεσμα, επιλέγουμε το 0, επιτρέποντας στην τοπική στοίχιση να ξεκινήσει ξανά χωρίς «βαρύ» προβάδισμα προηγούμενων κακών τμημάτων.

Μετά τη συμπλήρωση όλων των κελιών με τον παραπάνω κανόνα, αναζητούμε το μέγιστο στο σύνολο των τιμών. Στην περίπτωσή μας, η μέγιστη τιμή είναι 4, που σημαίνει ότι υπάρχουν υποαλυλουχίες των v και w των οποίων η στοίχιση αποδίδει τέσσερα περισσότερα ορθά ταυτίσματα από τα συνολικά λάθη και τα κενά. Για να βρούμε μια βέλτιστη τοπική ευθυγράμμιση, κάνουμε οπισθοδρόμηση από οποιοδήποτε κελί με τιμή 4, ακολουθώντας τις ενδείξεις αν προήλθε από διαγώνια κίνηση (ταίριασμα/λάθος ταίριασμα) ή από κενό, μέχρι να φτάσουμε σε κελί με τιμή 0. Ένα παράδειγμα τέτοιας στοίχισης φέρνει την υποαλυλουχία “TTAAG” από τη θέση 6–10 του v και την “TTGAA” από τη θέση 2–6 του w, με τέσσερις ακριβείς ταυτίσεις και κανένα λάθος ή κενό, αποδίδοντας έτσι συνολικό σκορ 4.

Ερώτημα 6

Άσκηση α

Ξεκινάμε με την εύρεση της ανθρώπινης πρωτεΐνης P61626 και την εκτέλεση BLASTP εναντίον της βάσης Swiss-Prot. Ζητώντας περισσότερα αποτελέσματα και αφαιρώντας το ακριβές ανθρώπινο ταίρι, εντοπίστηκαν με αξιοπιστία οκτώ ομόλογες ακολουθίες από άλλους οργανισμούς.

Σε κάθε ένα από τα οκτώ αρχεία pairwise .aln5, βλέπουμε την ανθρώπινη αλληλουχία στοιχισμένη σε Clustal format. Για παράδειγμα στο pairwise_1.aln5, στοιχίζεται η ανθρώπινη λυσοζύμη με την ομόλογη του γορίλλα. Παρατηρείται σχεδόν τέλεια ταυτότητα σε ολόκληρη τη λειτουργική περιοχή της πρωτεΐνης, ειδικά γύρω από τα καταλυτικά υπολείμματα (σημειωμένα με “*”). Μικρές διαφορές εντοπίζονται μόνο σε λίγες επιφανειακές θέσεις.

Το τελικό αρχείο συνενώνει όλες τις αλληλουχίες σε μία πολλαπλή στοίχιση. και η απόλυτη συντήρηση των καταλυτικών υπολειμμάτων ξεχωρίζει σε

κάθε στήλη. Οι θηλαστικοί ομόλογοι (άνθρωπος, γορίλλας, γουρούνι, άλογο, αγελάδα) ομαδοποιούνται στενά με ελάχιστα κενά, ενώ οι αμφίβιοι εισάγουν πιο εκτεταμένες διαγραφές, ορατές ως μπλοκ παυλών σε βρόχους.

Άσκηση β

https://pubmed.ncbi.nlm.nih.gov/17431180/

Στην πρωτοποριακή μελέτη τους του 2007, χρησιμοποίησαν τη μέθοδο «shotgun» υγρής χρωματογραφίας σε συνδυασμό με φασματομετρία μάζας δευτέρου σταδίου (LC-MS/MS) για να απομονώσουν και να αλληλουχίσουν πεπτιδικά θραύσματα από απολιθωμένο οστό του Tyrannosaurus rex. Συγκρίνοντας τα παραγόμενα φάσματα με γνωστές βάσεις δεδομένων πρωτεϊνών, εντόπισαν με βεβαιότητα πολλαπλά πεπτίδια που αντιστοιχούνται σε περιοχές της κολλαγόνου τύπου I, της κύριας δομικής πρωτεΐνης του οστού. Η επιτυχία αυτής της προσέγγισης απέδειξε ότι ακόμη και σε ένα δείγμα ηλικίας 68 εκατομμυρίων ετών, διακριτικά μοριακά σήματα μπορούν να επιβιώσουν και να ανιχνευθούν με σύγχρονες τεχνικές.

Η παρουσία ανέπαφων θραυσμάτων κολλαγόνου σε τόσο αρχαίο υλικό ήταν αναπάντεχη, δεδομένων των γρήγορων ρυθμών διάσπασής τους που προβλέπονταν από συμβατικά μοντέλα αποδόμησης πρωτεϊνών. Έδειξαν ότι ορισμένοι τομείς του μορίου κολλαγόνου διαθέτουν εξαιρετικά μεγάλη χημική σταθερότητα, επιτρέποντας τους να αντιστέκονται στην υδρόλυση και την μικροβιακή διάσπαση για δεκάδες εκατομμύρια χρόνια. Αυτό το εύρημα αμφισβήτησε τις προηγούμενες υποθέσεις σχετικά με την επιβίωση βιομορίων σε απολιθώματα βαθιάς γεωλογικής κλίμακας και υπογράμμισε τη σημασία των εγγενούς δομικών χαρακτηριστικών στην επιμήκυνση της διάρκειας ζωής των πρωτεϊνών.

Πέρα από την απλή ανίχνευση, τα δεδομένα αλληλουχίας παρείχαν νέα φυλογενετικά στοιχεία, δηλαδή τα ανακτώμενα πεπτίδια κολλαγόνου του T. rex εμφάνιζαν μεγαλύτερη ομοιότητα με αυτά των σύγχρονων πτηνών, όπως το κοτόπουλο, παρά με ερπετά ή αμφίβια.

Για να εξηγήσουν πώς αυτά τα πεπτίδια διατηρήθηκαν για τόσο μεγάλο χρονικό διάστημα, οι συγγραφείς πρότειναν ότι ο σίδηρος που απελευθερώνεται κατά την αποδόμηση της αιμοσφαιρίνης και άλλων σιδηρούχων μορίων μπορεί να καταλύει αντιδράσεις διασταυρούμενων δεσμών (cross-linking) στο μήτρα του κολλαγόνου. Μια τέτοια σιδηροεξαρτώμενη σταθεροποίηση θα «σταθεροποιούσε» το μόριο, καθιστώντας το λιγότερο επιρρεπές σε ενζυματική και χημική διάσπαση.