ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Εισαγωγή στη Βιοπληροφορική - Προπτυχιακό
(CEID1047)

Πρώτο Σύνολο Ασκήσεων 2021-2022

Στεφανίδης Μάριος — 1067458
Μητροπούλου Αικατερίνα — 1067409

Πάτρα, Ιούνιος 2022

Η παρούσα αναφορά συντάσσεται με αφορμή την πρώτη εργασία στο μάθημα "Εισαγωγή στην Βιοπληροφορική" που διδάσκεται στο Πανεπιστήμιο Πατρών και παρακολουθείται από φοιτητές των τμημάτων Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών (ΗΜΤΥ) και Μηχανικών Υπολογιστών και Πληροφορικής (CEID). Κύριος στόχος είναι η εκτενής παρουσίαση και επεξήγηση των λύσεων από τις δοθέντες ασκήσεις που περιέχει η εργασία.

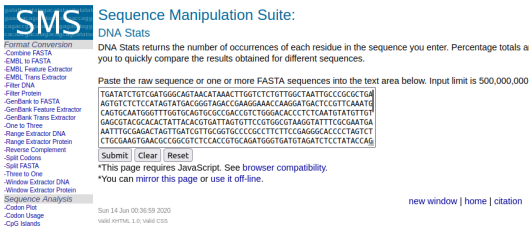
1 Άσκηση Πρώτη

Στόχος της συγκεκριμένης άσκησης είναι η διερεύνηση έτοιμων εργαλείων λογισμικού που αφορούν τον χειρισμό προβλημάτων του κλάδου της Βιοπληροφορικής. Θα γίνει δοκιμή κάποιων παραδειγμάτων χρήσης εργαλείων λογισμικού που καταγράφονται στη σελίδα **Rosalind Problems** ενώ ακόμα για το καθένα από αυτά θα γίνει μια μικρή αναφορά χρήσης και εμπειρίας με διάφορα δεδομένα [1].

1.1 Introduction to the Bioinformatics Armory

Σε αυτό το σημείο γίνεται αναφορά του εργαλείου DNA STATS του Sequence Manipulation Suite. Ουσιαστικά, μετράει τον αριθμό των εμφανίσεων κάθε νουκλεοτιδίου σε μια δεδομένη αλυσίδα DNA.

Πιο συγκεκριμένα, δίνεται ως είσοδος μια ακολουθία DNA (=S) που έχει μήκος το πολύ 1000-bp και επιστρέφονται 4 ακέραιοι αριθμοί (χωρισμένοι με κενά) που αντιπροσωπεύουν το πλήθος των φορών που εμφανίστηκαν τα σύμβολα 'A', 'C', 'G' και 'T' στο S.



Εικόνα 1: Είσοδος

DNA Stats results

Results for 828 residue sequence "Untitled" starting "TGATATCTGT"

Pattern:	Times found:	Percentage:
g	214	25.36
a	193	23.31
t	212	25.60
c	213	25.72
n	0	0.00
u	0	0.00
r	0	0.00
y	0	0.00
s	0	0.00
w	0	0.00

Εικόνα 2: Έξοδος

Όπως φαίνεται και παραπάνω, η χρήση του εργαλείου είναι ιδιαίτερα απλή. Αρκεί απλά να κατεβάσουμε το Sample Dataset και να αντιγράψουμε τα περιεχόμενά του στο πλαίσιο του κειμένου (εικόνα 1). Έπειτα, επιλέγουμε "submit" και αυτόματα επιστρέφεται η ανάλυση του dataset (εικόνα 2). Ειδικότερα:

[1] Αποφύγαμε να κατεβάσουμε τοπικά στους υπολογιστές μας τα λογισμικά των προτεινόμενων εργαλείων και προσπαθήσαμε να χρησιμοποιήσουμε αποκλειστικά τις online ηλεκτρονικές διεπαφές, όπου αυτές ήταν διαθέσιμες

Βάση	Πλήθος
Αδενίνη (A)	193
Θυμίνη (T)	212
Κυτοσίνη (C)	213
Γουανίνη (G)	210

1.2 GenBank Introduction

Με τη χρήση του συγκεκριμένου λογισμικού δίνεται η δυνατότητα να κάνουμε απλά ερωτήματα με χρήση περιορισμών στην GenBank.

Πιο συγκεκριμένα, μπορούμε να πραγματοποιήσουμε αναζήτηση στη βάση Δεδομένων Nucleotide με γενικά ερωτήματα κειμένου, η οποία θα παράγει στη συνέχεια τα πιο σχετικά αποτελέσματα. Δίνεται ακόμα η δυνατότητα να κάνουμε ένα απλό ερώτημα που βασίζεται σε όνομα πρωτεΐνης, όνομα γονιδίου ή σύμβολο γονιδίου. Ωστόσο, σε περίπτωση που επιθυμούμε να περιορίσουμε την αναζήτηση μας σε συγκεκριμένα είδη εγγραφών, μπορούμε να χρησιμοποιήσουμε την σελίδα Limits page της GenBank ή εναλλακτικά το πεδίο φιλτραρίσματος των αποτελεσμάτων για να επιλέξουμε συγκεκριμένες κατηγορίες εγγραφών.

Παραδείγματος χάριν, δίνεται στο λογισμικό ως είσοδος το όνομα γένους, ακολουθούμενο από δύο ημερομηνίες σε μορφή EEEE/M/H και επιστρέφεται ο αριθμός των εγγραφών Nucleotide GenBank για το συγκεκριμένο γένος που δημοσιεύτηκε μεταξύ των καθορισμένων ημερομηνιών.

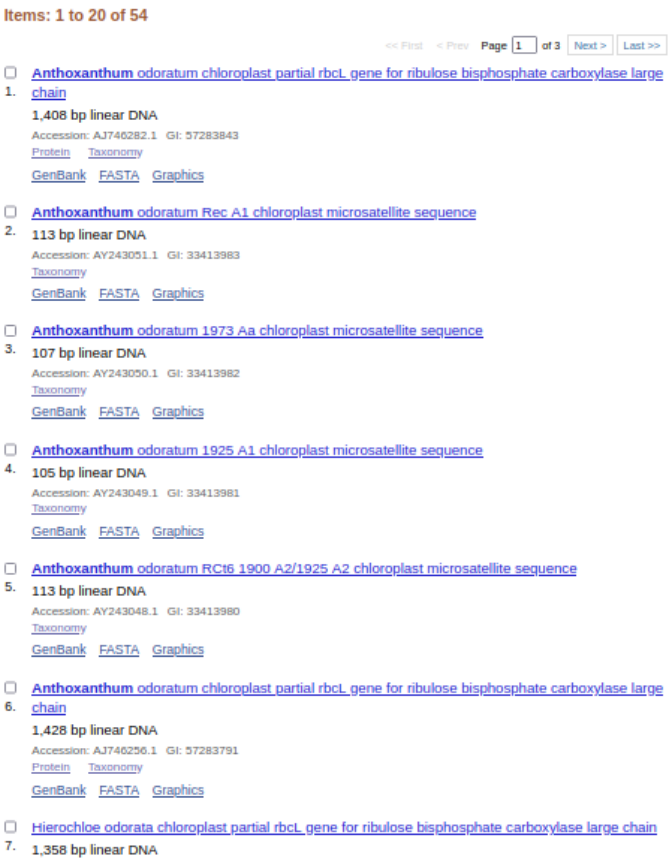
Εικόνα 3: Αναζήτηση

Όπως φαίνεται και παραπάνω, έχει χρησιμοποιηθεί το γραφικό περιβάλλον της GenBank και έχει πραγματοποιηθεί η αναζήτηση του Anthoxanthum μεταξύ των ημερομηνιών 2003/7/25 έως 2005/12/27 (εικόνα 3). Η παραπάνω αναζήτηση επιστρέφει 54 αποτελέσματα (εικόνα 4).

1.3 Data Formats

Στο συγκεκριμένο πρόβλημα δίνουμε περισσότερη έμφαση στο Format που χρησιμοποιούν οι απαντήσεις που επιστρέφονται από τα ερωτήματα στη GenBank (βλ. ενότητα 1.2). Ακόμα, ερχόμαστε σε επαφή με το εργαλείο GenBank to FASTA.

Δίνεται ως είσοδος μια συλλογή από n ($n \leq 10$) αναγνωριστικά εισόδου GenBank (εικόνα 5) και επιστρέφεται η συντομότερη από τις συμβολοσειρές που σχετίζονται με τα αναγνωριστικά σε μορφή FASTA (εικόνα 6).

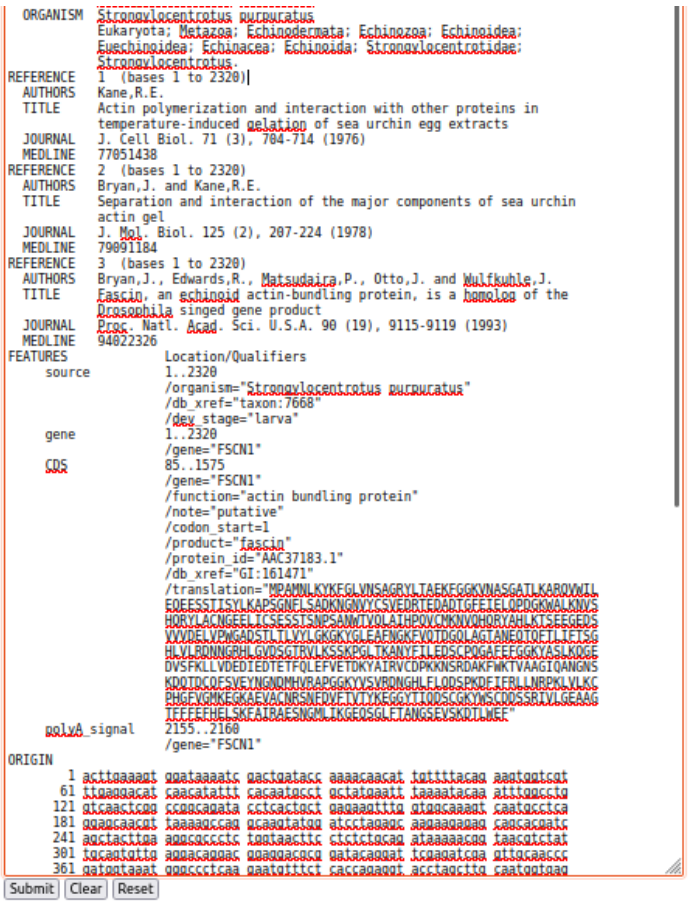


Εικόνα 4: Αποτελέσματα

1.4 New Motif Discovery

Με το συγκεκριμένο εργαλείο μπορούμε να βρίσκουμε μοτίβα μέσα σε συμβολοσειρές. Αφού δώσουμε την είσοδο (είτε σε μορφή αρχείου είτε συμπληρώνοντας το αντίστοιχο πλαίσιο κειμένου) (εικόνα 7), το MEME εκτελεί ένα "job". Στη συνέχεια, προκύπτουν τα αποτελέσματα, τα οποία μπορούμε να δούμε σε διαφορετικές μορφές (εικόνα 8).

Δίνεται ως είσοδος ένα σύνολο ακολουθιών πρωτεΐνης σε μορφή FASTA που μοιράζονται κάποιο μοτίβο με ελάχιστο μήκος 20 και επιστρέφεται μια κανονική έκφραση για το μοτίβο με την καλύτερη βαθμολογία. Ενδεικτικά, μπορείτε να δείτε τις μορφές MEME HTLM Output και MAST HTLM Ouput.



Εικόνα 5: Αναγνωριστικά Εισόδου GenBank

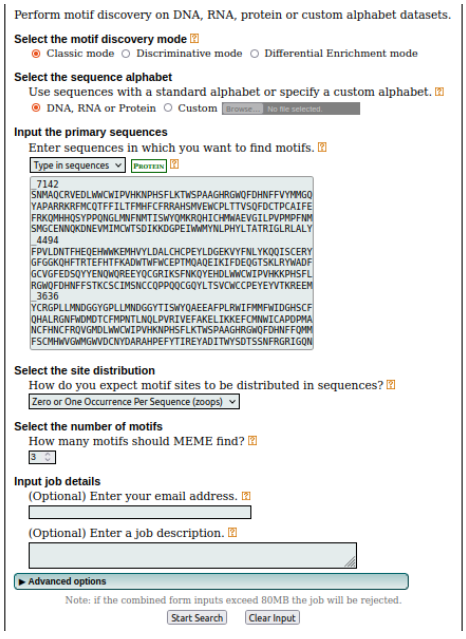
1.5 Pairwise Global Alignment

Σε αυτό το σημείο ασχολούμαστε με ένα εργαλείο ολικής στοίχισης ακολουθιών RNA και DNA, το Needle. Δέχεται ως είσοδο δύο ταυτότητες GenBank (εικόνα 9) και επιστρέφεται η μέγιστη συνολική βαθμολογία ευθυγράμμισης μεταξύ των συμβολοσειρών DNA που σχετίζονται με αυτά τα αναγνωριστικά (εικόνα 10).

GenBank to FASTA results

```
>Strongylocentrotus purpuratus fascin (FSCN1) mRNA, complete cds.
acttgaaagtggataaaatcgactgataccaaaacaacattgttttacagaagtggtcgt
ttgaggacatcaacatatttcacaatgcctgctatgaatttaaaatacaaatttggcctg
gtcaactcggccggcagatacctcactgctgagaagtttggtggcaaagtcaatgcctca
ggagcaacgttaaaagccaggcaagtatggatcctagagcaagaagagagcagcacgatc
agctacttgaaggcgccctctggtaacttcctctctgcagataaaaacggtaacgtctat
tgcagtgttgaggacaggacggaggacgcggatacaggattcgagatcgagttgcaaccc
gatggtaaatgggccctcaagaatgtttctcaccagaggtacctagcttgcaatggtgag
gagctgatctgcagtgaatccagcaccagcaaccctcagcaaactggactgtccagctg
gccatccatccacaggtctgcatgaagaacgtccagcaccaacgctacgcacatctcaaa
accagtgaggagggtgaagacagcgtggttgtagacgaattggttccctggggagctgat
tccacactcactcttgtctacctgggcaaaggaaagtacggccttgaggccttcaacgga
aagtttgtccaaaccgacggacagcttgctggcacagccaacgaacagacgcagttcaca
ctcatcttcacatccggtcacctggtactaagggacaacaatggacgtcacttaggagtg
gacagtggaaccagggtcttgaagtcctccaagcctggactgacgaaagccaattacttc
atcctagaggatagctgtccacaaggtgccttcgaatttggtggcaaatatgcatcgtta
aagcaaggcgaagatgtttcattcaagcttcttgttgacgaagatatcgaagacacagag
accttccagttggagttcgttgaaaccgacaagtatgccatcagggtatgtgaccccaag
aagaactccagagatgctaagttctggaagaccgtcgctgctggtatccaggctaacggc
aactcaaaggaccagacggactgtcaattctctgtcgaatacaacggcaacgacatgcat
gtgcgtgctccaggaggcaagtatgttagtgtccgtgacaacggccatctcttccttcag
gattcacccaaagacttcatcttccgtctgctcaaccgacccaagctggtgctcaagtgc
cctcatggattcgtgggcatgaaggagggcaaggctgaggtcgcctgcaaccgatcaaac
tttgatgtcttcactgtcacctacaaggaaggcggatacactatccaagactcctgtggc
aagtactggtcttgtgatgacagtagccgcatcgttcttggagaggcagcaggtactttc
ttcttcgagttccatgagctctccaagtttgctatccgagcagaaagcaacggcatgttg
atcaaggcgagcagagtggcttgtttaccgccaatggttccgaggtctcaaaggacaca
ctgtgggaattctaaacaaattgggcttgaaagaagccaaatccaaatcagaagtagagt
agctgacaagccagccactctatctattatatcaattgcaaatattgtacatttttttaa
tacaaaaatattttcaaaggtgcataataattatttcatactctggtgggatctttagga
tcatattttctcattgccttggcatactgacttccattatccctcatttttaaaaggta
aattgatcacttaatatcaactgaaaacgaaatggaagtaggtctctggaaatttagaag
aatagatgactatccagtatattcaaatatttgttgaacctgtctcaaaaaaaccccata
aaaaaccctctgttttgtgctgtctcatagccataaatagagatcaattctggtggtata
tgctacttaaaatcaggcttgaaatagaataaatggaatggaatggattttcaaaaagat
ttggaattttaaatttgcagccagtgtccgtgacaactcttgcataccagaagcactaaa
cagtctctgccgcaccgctcgcccaggtgtattgttgtgactgtacttttgaacaataaa
acagatcttttcgtcaagtttggatataaaagtggattgaaatgcacgaatagattcgac
ttgtatagggcatggtggacattgattttacagatactttcaatataccggtaaaaatca
atcatatagaaaatgaaaacagggtgtaatatctcaaata
```

Εικόνα 6: Συντομότερη Συμβολοσειρά



Εικόνα 7: Είσοδος σε μορφή FASTA

Your MEME job is complete. The results should be displayed below.

Job Details ...

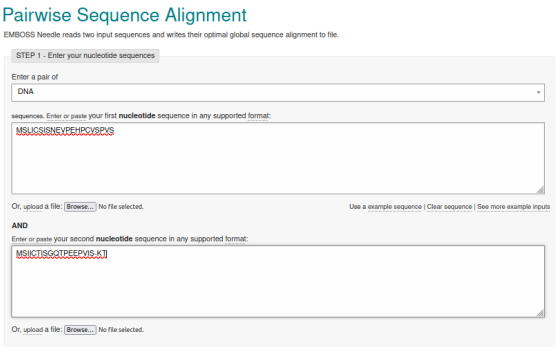
Results

- MEME HTML output
- MEME XML output
- MEME text output
- MAST HTML output
- MAST XML output
- MAST text output
- (Primary) Sequences

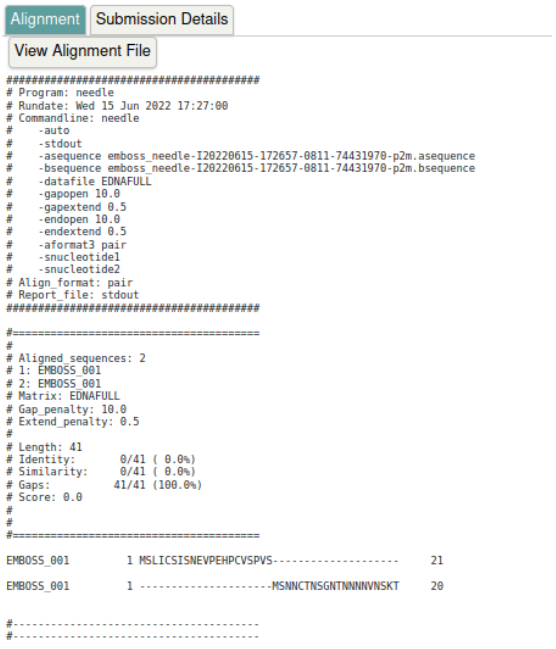
Status Messages

- Arguments ok
- Starting meme
  `meme sequences.fa -protein -oc . -nostatus -time 14400 -mod zoops -nmotifs 3 -minw 6 -maxw 50 -objfun classic -markov_order 0`
- meme ran successfully in 1.07 seconds
- Starting mast
  `mast meme.xml sequences.fa -oc . -nostatus`
- mast ran successfully in 0.18 seconds
- Done

Εικόνα 8: Μορφές αποτελέσματος



Εικόνα 9: Είσοδος ακολουθιών DNA



Εικόνα 10: Μέγιστη βαθμολογία ευθυγράμμισης

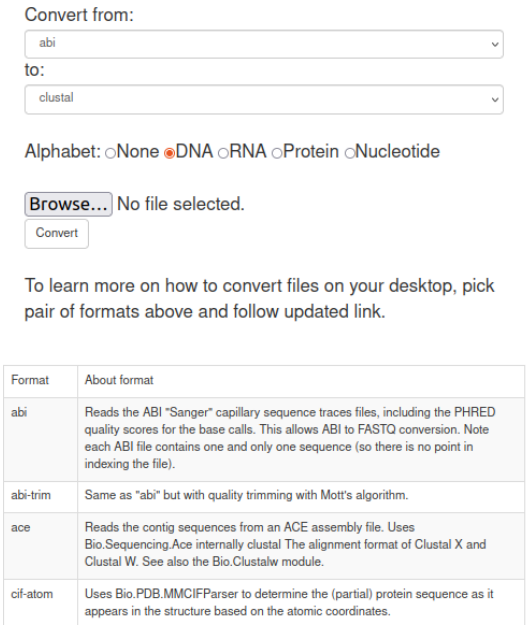
1.6 FASTQ format introduction

Το FASTQ είναι άλλο ένα format για αναπαράσταση βιολογικών ακολουθιών. Στη συγκεκριμένη περίπτωση, ερχόμαστε σε επαφή με εργαλεία που αφορούν στην μετατροπή του FASRQ σε FASTA.

Τέτοια εργαλεία για παράδειγμα είναι τα εξής:

- Online sequence conversion tool (εικόνα 11)
- Galaxy
- BlastStation

Όπως γίνεται προφανές, δέχεται ως είσοδο ένα αρχείο FASTQ και επιστρέφει τις αντίστοιχες εγγραφές σε μορφή FASTA.



Εικόνα 11: Sequence conversion tool

1.7 Read Quality Distribution

Σε αυτό το σημείο ασχολούμαστε με τον έλεγχο ποιότητας. Πιο συγκεκριμένα, υπάρχουν εργαλεία, τα οποία ελέγχουν την ποιότητα της ακολουθίας που έχει δοθεί ως είσοδο, προκειμένου να δούμε αν θα εξάγουμε από την παραπάνω σωστά δεδομένα.

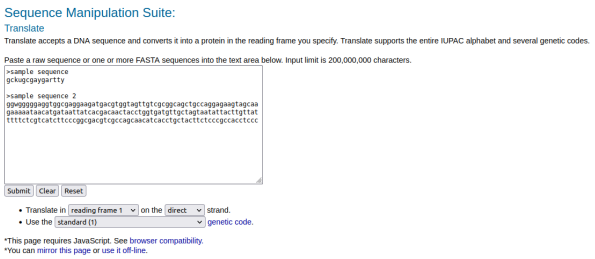
Δίνεται ως είσοδος ένα όριο ποιότητας μαζί με καταχωρήσεις FASTQ για πολλαπλές αναγνώσεις και επιστρέφεται ο αριθμός των αναγνώσεων, των οποίων η μέση ποιότητα είναι κάτω από το όριο. Υπάρχουν οι εξής επιλογές:

- Μια έκδοση του FastQC μπορεί να ληφθεί από το Babraham Bioinformatics και να εκτελεστεί τοπικά σε οποιοδήποτε λειτουργικό σύστημα με εγκατεστημένο κατάλληλο Java Runtime Environment (JRE).
- Μια ηλεκτρονική έκδοση του FastQC είναι επίσης διαθέσιμη στο Galaxy

1.8 Protein Translation

Το πακέτο SMS 2 (έχει αναφερθεί και παραπάνω) διαθέτει ένα εργαλείο μετάφρασης - **Translate** -, το οποίο μπορεί να μετατρέψει μια αλληλουχία βάσεων σε αλληλουχία αμινοξέων.

Δέχεται ως είσοδο μια συμβολοσειρά DNA (=S) μήκους το πολύ 10 kbp και μια συμβολοσειρά πρωτεΐνης μεταφρασμένη από την S και επιστρέφει τον δείκτη παραλλαγής γενετικού κώδικα που χρησιμοποιήθηκε για την μετάφραση.



Εικόνα 12: Είσοδος στο translate

Εικόνα 13: Αποτέλεσμα

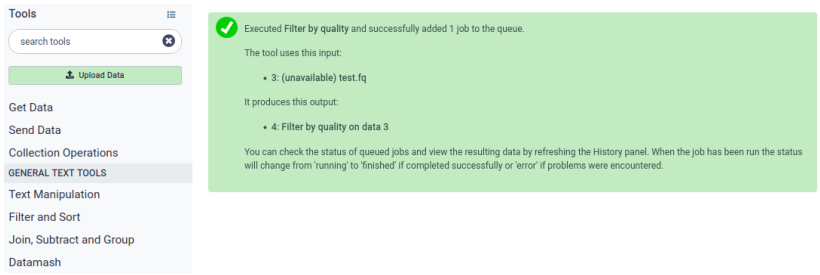
1.9 Read Filtration by Quality

Όταν διαθέτουμε κακή ποιότητα αλληλουχιών, μας δίνεται η δυνατότητα μέσω του εργαλείου FASTQ Quality Filter από το σύνολο εργαλειών του FASTX, να φιλτράρουμε τα δεδομένα προκειμένου να κρατήσουμε μόνο τα αξιόλογα.

Δίνεται ως είσοδος μια τιμή κατωφλίου q, ένα ποσοστό βάσεων p και ένα σύνολο καταχωρήσεων FASTQ (εικόνα 14) και επιστρέφεται ο αριθμός αναγνώσεων σε φιλτραρισμένες καταχωρήσεις τύπου FASTQ (εικόνα 16).

Έχει χρησιμοποιηθεί η ηλεκτρονική διεπαφή για το φίλτρο ποιότητας FASTQ, η οποία είναι διαθέσιμη μέσω της πλατφόρμας Web Galaxy (εικόνα 15).

Εικόνα 14: Είσοδος στο FASTQ



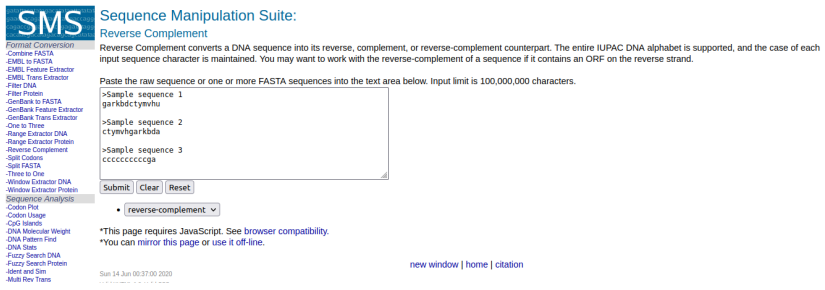
Εικόνα 15: Μήνυμα Επιτυχίας

Εικόνα 16: Έξοδος/Αποτέλεσμα

1.10 Complementing a Strand of DNA

Γνωρίζουμε πως σε μια ακολουθία DNA (έστω S), τα 'A' και 'T' είναι συμπληρωματικά μεταξύ τους, όπως και τα 'C' και 'G'. Το εργαλείο Reverse Complement του SMS 2 πακέτου λοιπόν υπολογίζει το αντίστροφο συμπλήρωμα μιας αλυσίδας (π.χ., το αντίστροφο συμπλήρωμα του "GTCA" είναι "TGAC").

Δίνεται ως είσοδος μια συλλογή από n ($n \leq 10$) συμβολοσειρές DNA και επιστρέφεται ο αριθμός των συμβολοσειρών που δόθηκαν ως είσοδος που ταιριάζουν με τα αντίστροφα συμπληρώματά τους (εικόνα 18). Έχει χρησιμοποιηθεί η online ηλεκτρονική διεπαφή του προγράμματος Reverse Complement (εικόνα 17).

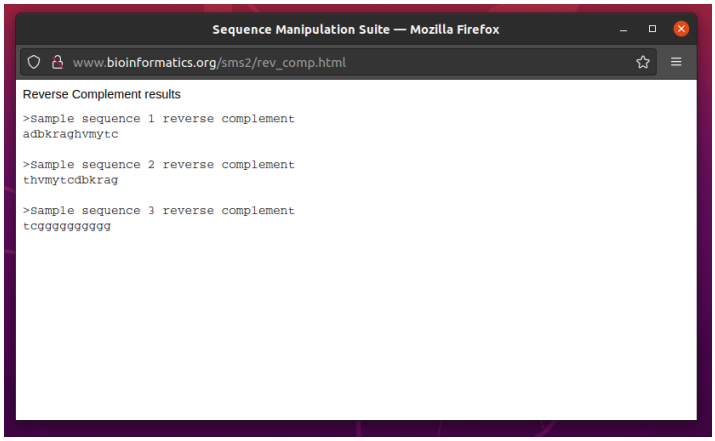


Εικόνα 17: Είσοδος στο Reverse Complement

1.11 Suboptimal Local Alignment

Μέσω του εργαλείου Lalign μπορούμε να βρούμε, μεταξύ 2 ακολουθιών, τα πολλαπλά εναλλακτικά τοπικά ταιριάσματα τους.

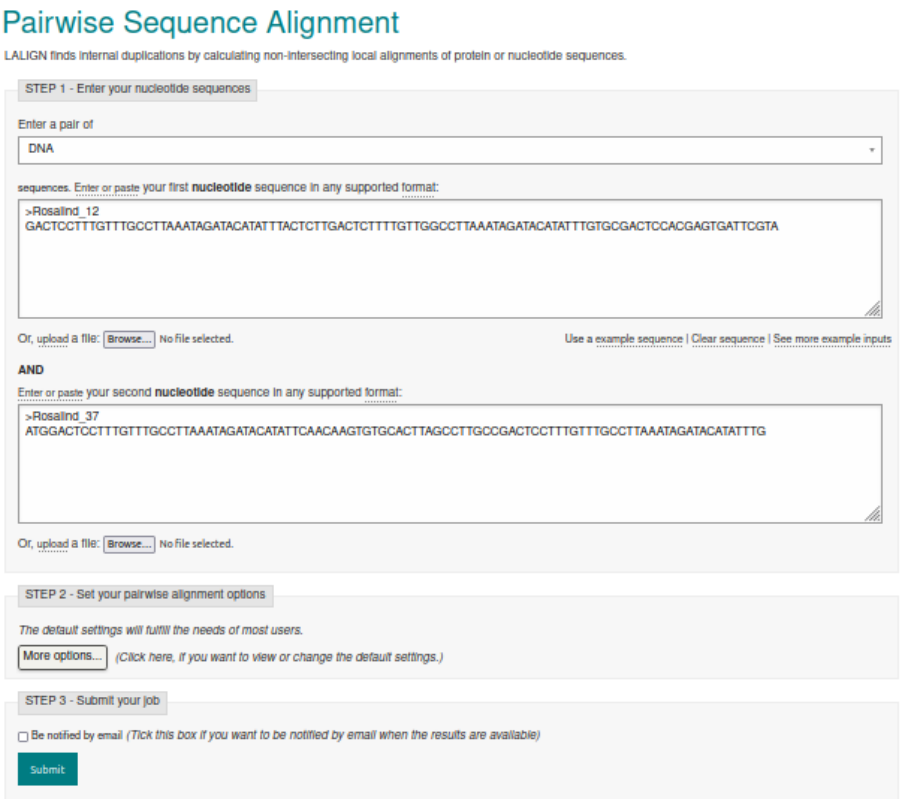
Δίνονται ως είσοδο 2 συμβολοσειρές DNA S και T σε μορφή FASTA που μοιράζονται κάποια σύντομη ανακριβή επανάληψη r των 32-40 bp (με τον όρο ανακριβή εννοούμε πως το r μπορεί να εμφανίζεται με



Εικόνα 18: Μήνυμα Αποτελέσματος

μικρές τροποποιήσεις) (εικόνα 19) και επιστρέφεται ο συνολικός αριθμός εμφανίσεων του r ως υποσυμβολοσειρά του s, ακολουθούμενη από τον συνολικό αριθμό εμφανίσεων του r ως υποσυμβολοσειρά του t.

Χρησιμοποιείται η online διαθέσιμη ηλεκτρονική διεπαφή. Μπορείτε να βρείτε εδώ τα αποτελέσματα.



Εικόνα 19: Είσοδος 2 συμβολοσειρών DNA

1.12 Base Quality Distribution

Η ποιότητα των βάσεων μπορεί να ποικίλλει ανάλογα με τη θέση κατά την διάρκεια της ανάγνωσης εξαιτίας της φύσης της διαδικασίας αλληλουχίας. Δίνεται λοιπόν η δυνατότητα να ελέγξει κάποιος αυτήν την κατανομή ποιότητας χρησιμοποιώντας τη μονάδα "Per Base Sequence Quality" του προγράμματος FastQC. Οι μέσες αποδεκτές τιμές ποιότητας είναι 10 για το κατώτερο τεταρτημόριο και 25 για το μεσαίο.

Εάν οι τιμές πέσουν κάτω από αυτό το όριο, τότε η μονάδα επιστρέφει μια προειδοποίηση. Σημειώνεται πως για τις αναγνώσεις >50bp το FastQC θα ομαδοποιήσει τις βάσεις.

Δίνεται ως είσοδος ένα αρχείο FASTQ με όριο ποιότητας q και επιστρέφεται ο αριθμός θέσεων όπου η μέση ποιότητα βάσης πέφτει κάτω από το παραπάνω όριο.

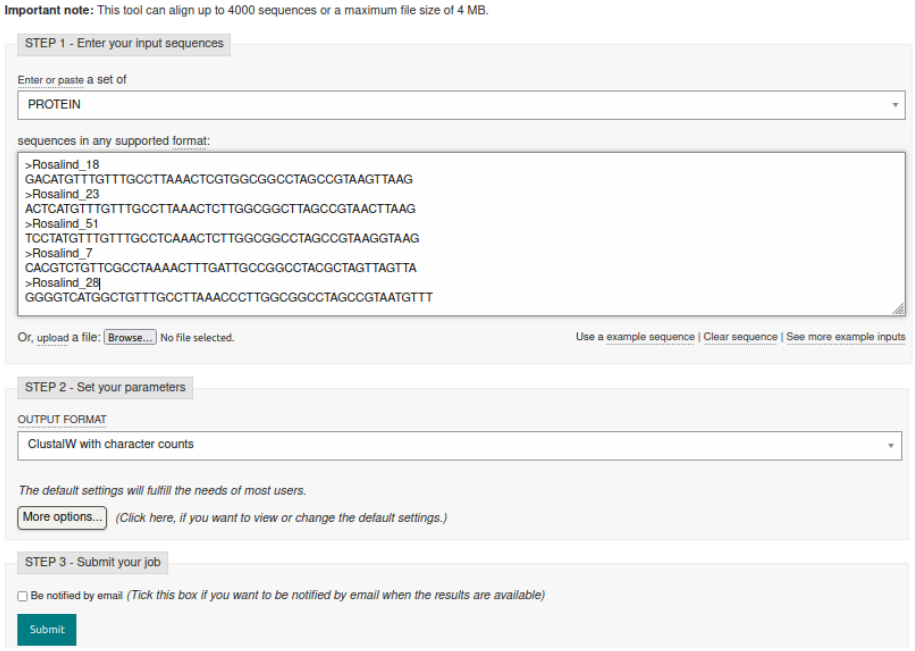
1.13 Global Multiple Alignment

Σε αυτό το σημείο, χρησιμοποιείται το εργαλείο Clustal Omega για Mutliple Sequence ALignment, του οποίου η λειτουργία φαίνεται παρακάτω.

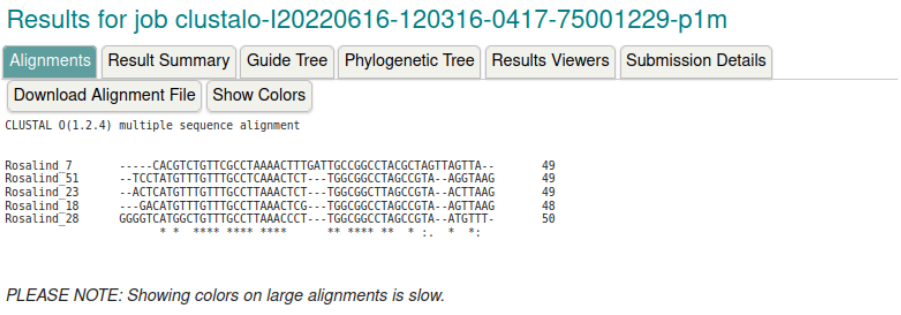
Πιο συγκεκριμένα, γίνεται η επιλογή της λειτουργίας "Protein" ή "DNA" και, στη συνέχεια, η επικόλληση της ακολουθίας σε μια από τις αναφερόμενες μορφές (δίνεται και η δυνατότητα ανεβάσματος απευθείας του κατάλληλου αρχείου). Για την απόκτηση καλύτερης στοίχισης, κατάλληλη είναι η αργή επιλογή. Εάν το Clustal εκτελεστεί μόνο σε 2 ακολουθίες, τότε οι επιλογές παραμέτρων αντιστοιχούν με εκείνες του Needle (βλ. ενότητα 1.5).

Δίνεται ως είσοδος ένα σύνολο νουκλεοτιδικών ακολουθιών σε μορφή FASTA και επιστρέφεται το αναγνωριστικό της συμβολοσειράς που είναι πιο διαφορετικό από τις υπόλοιπες.

Χρησιμοποιείται η online διαθέσιμη ηλεκτρονική διεπαφή (εικόνα 20).



Εικόνα 20: Είσοδος στο Clustal Omega



Εικόνα 21: Αποτελέσματα

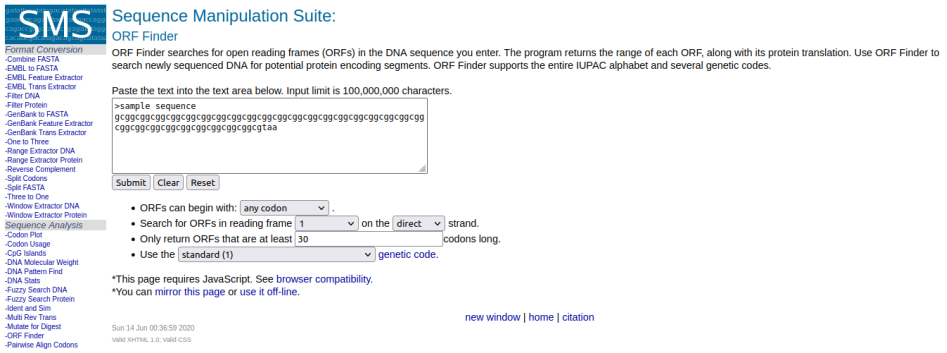
Όπως βλέπουμε, παίρνοντας τις ακολουθίες που δίνονται ως παραδείγματα από τη Rosalind και τοποθετώντας στο Clustal Omega, επιστρέφεται η πολλαπλή στοίχιση που προκύπτει, σύμφωνα με την είσοδο (εικόνα 21).

1.14 Finding Genes with ORFs

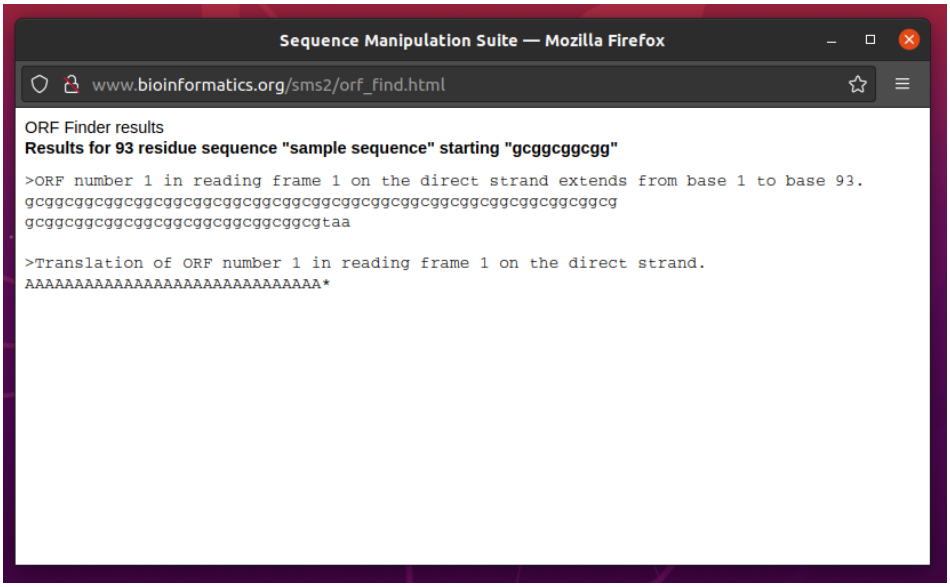
Το εργαλείο ORF finder από το πακέτο SMS 2 ανιχνεύει κωδικόνια από μία αλληλουχία βάσεων DNA, προκειμένου να προκύψει τελικά η πρωτεΐνη.

Δίνεται ως είσοδος μια συμβολοσειρά DNA s μήκους το πολύ 1 kbp και επιστρέφεται η μεγαλύτερη συμβολοσειρά πρωτεΐνης που μπορεί να μεταφραστεί από ένα ORF του s. Εάν υπάρχουν περισσότερες από μία πρωτεϊνικές συμβολοσειρές μέγιστου μήκους, τότε μπορείτε να εξαχθεί οποιαδήποτε λύση.

Χρησιμοποιείται η online διαθέσιμη ηλεκτρονική διεπαφή (εικόνα 22). Παρατηρούμε τελικά τα κωδικόνια που ανιχνεύτηκαν σύμφωνα με τα δεδομένα εισόδου (εικόνα 23).



Εικόνα 22: Είσοδος στο ORF finder



Εικόνα 23: Αποτελέσματα

1.15 Base Filtration by Quality

Σε αυτό το σημείο χρησιμοποιούμε το εργαλείο FASTQ Quality Trimmer, το οποίο "τριμμάρει" μια ακολουθία σύμφωνα με παραμέτρους που αφορούν στην ποιότητα. Πιο συγκεκριμένα, οι βάσεις κακής ποιότητας μπορούν εύκολα να περικοπούν χρησιμοποιώντας συγκεκριμένο όριο (που ορίζεται από γραφική

παράσταση ποιότητας παρόμοιας με αυτή που αναφερθήκαμε στην ενότητα 1.12). Υπάρχουν πολλά τέτοια εργαλεία, κάποια από αυτά ενδεικτικά είναι:

- Το FASTQ Quality Trimmer του Galaxy. Χρησιμοποιεί μια προσέγγιση "συρόμενου παραθύρου", επομένως για ένα απλό κόψιμο των άκρων θα πρέπει να οριστεί το μέγεθος του παραθύρου σε 1
- Το Trimmomatic. Είναι ένα εργαλείο που βασίζεται στη γραμμή εντολών σε Java. Για ένα απλό κόψιμο και από τα δύο άκρα θα πρέπει να καθοριστούν κατάλληλα οι παράμετροι LEADING και TRAILING.

Δίνεται ως είσοδος ένα αρχείο FASTQ με μια τιμή αποκοπής ποιότητας q και επιστρέφεται το ίδιο αρχείο κομμένο και από τα δύο άκρα (αφαιρέθηκαν οι leading και trailing βάσεις με ποιότητα χαμηλότερη από q).

2 Άσκηση Δεύτερη

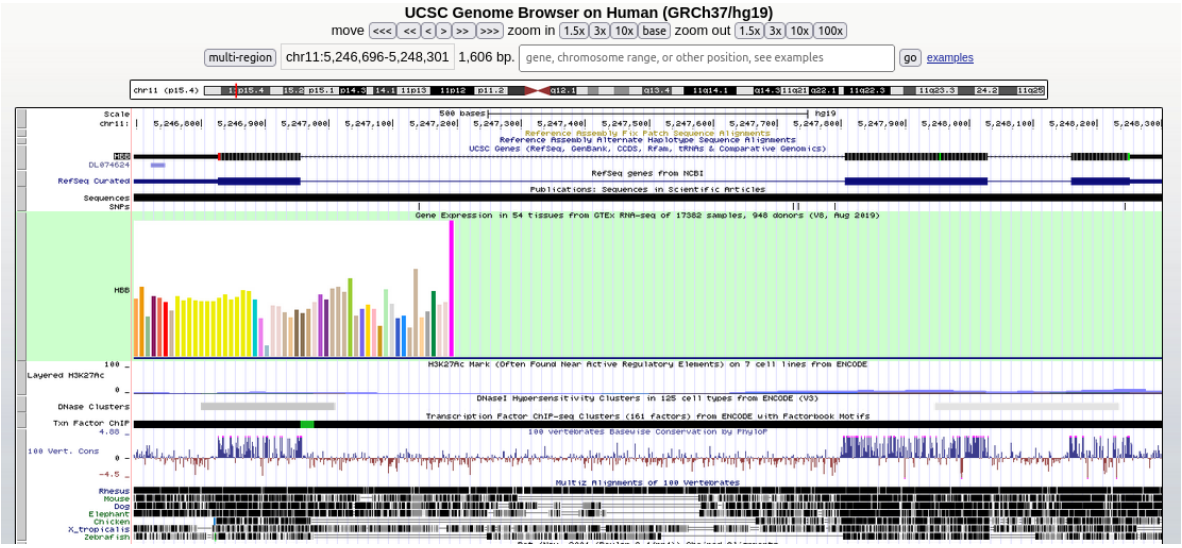
Σκοπός της συγκεκριμένης άσκησης είναι η στοίχιση ακολουθιών με χρήση του δικτυακού τόπου UCSC και του Genome Browser.

Το UCSC Genome Browser είναι ένα διαδικτυακό εργαλείο που χρησιμοποιείται σαν μικροσκόπιο πολλαπλής ισχύος, το οποίο επιτρέπει στους ερευνητές να προβάλλουν και τα 23 χρωμοσώματα του ανθρώπινου γονιδιώματος σε οποιαδήποτε κλίμακα από ένα πλήρες χρωμόσωμα έως ένα μεμονωμένο νουκλεοτίδιο.

Όσον αφορά στην άσκηση, θα γίνει επιλογή του human genome hg19 σύμφωνα με το ερώτημα hbb, το οποίο θα οδηγήσει στο αποτέλεσμα chr11:5,246,696–5,248,301. Το τελευταίο αντιπροσωπεύει μια περιοχή από 1606 ζευγών βάσεων που ορίζουν το beta globin γονίδιο. Στη συνέχεια, θα γίνει σύγκριση με διάφορους συνδυασμούς ειδών (όπως φαίνεται στις εικόνες 24 έως 28) σύμφωνα με τα Nets και Chains.

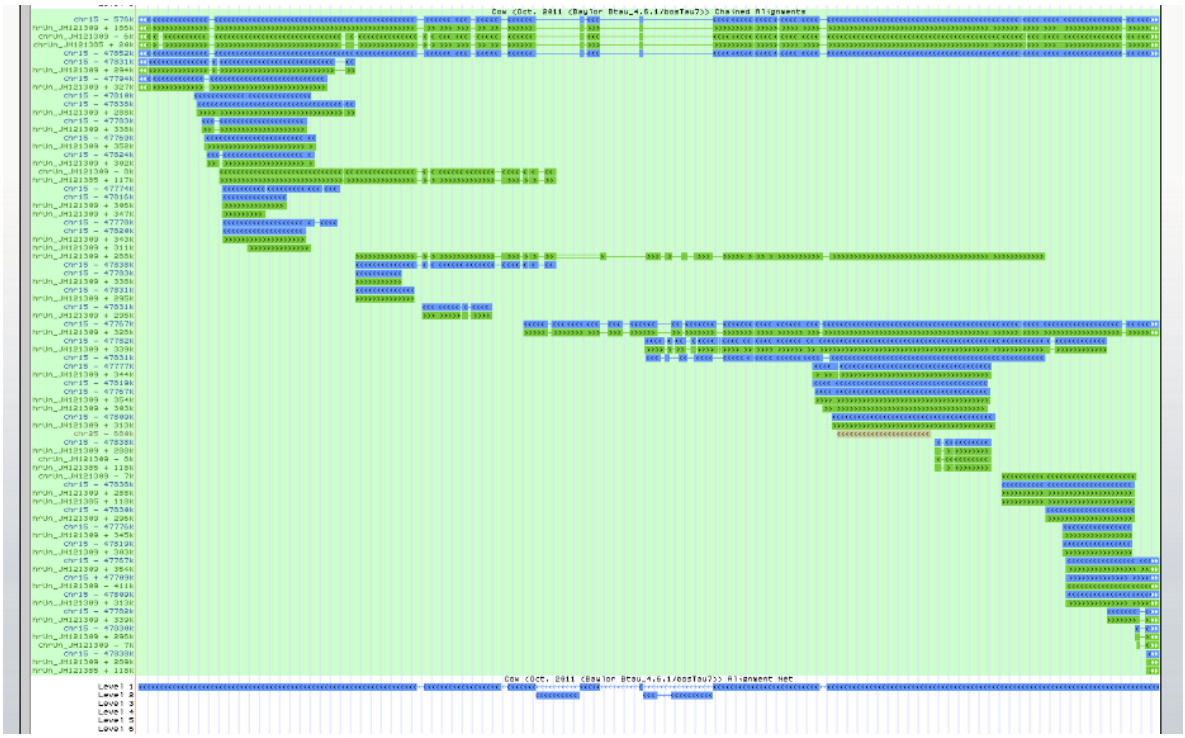
Τα Nets και Chains είναι συλλογές υψηλότερου επιπέδου βασικών ευθυγραμμίσεων ακολουθιών κατά ζεύγη. Τα Nets διασταυρούμενων ειδών χρησιμοποιούνται για τη δημιουργία μιας συλλογής μονής κάλυψης (single-coverage) - στο γονιδίωμα αναφοράς - από ζεύγη ευθυγραμμίσεων που αποτελούν τις βάσεις των ευθυγραμμίσεων πολλών ειδών Multiz στο κομμάτι της Διατήρησης. Οι αλγόριθμοι Nets και Chains, καθώς και τα αποτελέσματα από ευθυγραμμίσεις ανθρώπου - ποντικού δημιουργούνται από γονιδιωματικές τοπικές ευθυγραμμίσεις που υπολογίστηκαν από τον Blastz και τον Lastz, μετά την επεξεργασία από μια σειρά προγραμμάτων UCSC και κυρίως των axtChain, chainNet και netFilter.

Στις παρακάτω εικόνες παρουσιάζονται τα αποτελέσματα στοίχισης της ακολουθίας ζευγών βάσεων που το αποτελούν το γονίδιο HBB στον άνθρωπο, στο άλογο, στην αγελάδα, στο γουρούνι και στον αρουραίο. Όπως γίνεται προφανές υπάρχει μεγαλύτερη αντιστοιχία με την αγελάδα.

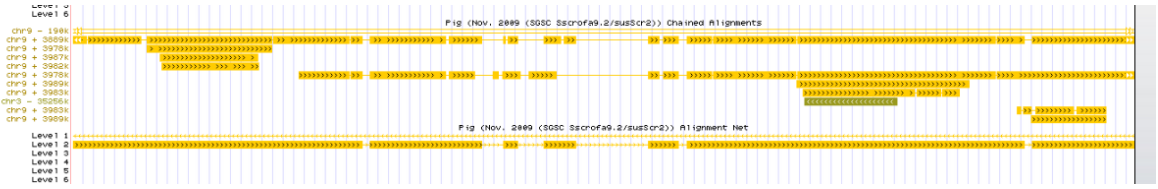


Εικόνα 24: Human genome hg19 με ερώτημα hbb

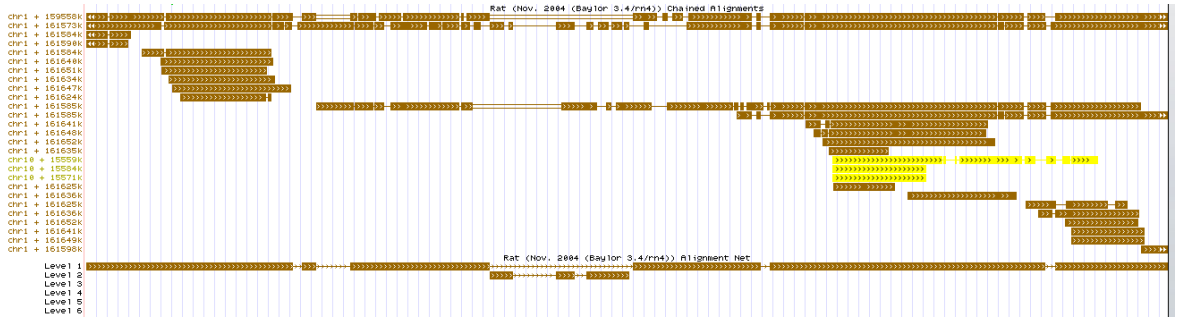
Εικόνα 25: Horse sample



Εικόνα 26: Cow sample



Εικόνα 27: Pig sample



Εικόνα 28: Rat sample

3 Άσκηση Τρίτη

3.1 Ερώτημα Α

Αρχικά, υλοποιείται το δέντρο επιθεμάτων (suffix tree) της συμβολοσειράς κειμένου T μήκους n, σύμφωνα με τον αλγόριθμο του Ukkonen. Στη συνέχεια, εφαρμόζεται ο αλγόριθμος DFS (Depth First Search - Αναζήτηση Κατά Βάθος) στο παραπάνω δέντρο, προκειμένου να ανατεθεί σε κάθε κόμβο μια ετικέτα (label), η οποία αντιπροσωπεύει τον ελάχιστο δείκτη επιθεμάτων που εμφανίζονται στους απογόνους του καθώς και την αντίστοιχη θέση του. Έπειτα, δεδομένης μιας συμβολοσειράς P, το δέντρο προσπελαύνεται έως ότου βρεθεί ο κόμβος που αντιστοιχεί στο P - ή κάποια προέκτασή του. Σε αυτό το σημείο λοιπόν ελέγχεται, αν η ετικέτα του συγκεκριμένου κόμβου υποδεικνύει εμφάνιση του P πριν από τη θέση k.

3.2 Ερώτημα Β

Οι Ilie και Smyth[2] εισήγαγαν έναν αποδοτικό αλγόριθμο, ο οποίος εντοπίζει τη μικρότερη μη επαναλαμβανόμενη συμβολοσειρά σε ένα κείμενο βασιζόμενοι σε μια απλή παρατήρηση. Η παραπάνω παρατήρηση αναφέρει πως κάθε ελάχιστη μοναδική υποσυμβολοσειρά [i...j] είναι η μικρότερη μεταξύ όλων των των μοναδικών επιθεμάτων (prefixes) που καταλήγουν στο j για όλα τα επιθέματα (suffixes) μιας λέξης w. Σύμφωνα με το παραπάνω λοιπόν εισήγαγαν έναν καινούργιο αλγόριθμο, στο τέλους του οποίου η μικρότερη μη-επαναλαμβανόμενη συμβολοσειρά [i...j] θα αποθηκευτεί, όταν MinUnique[j] = i.

Πιο συγκεκριμένα, στο βήμα 1 υπολογίζονται οι πίνακας SA (Suffix Array - περιέχει τις θέσεις 1, 2, ..., n ταξινομημένες σε αύξουσα λεξικογραφική σειρά των αντίστοιχων επιθεμάτων suf[i], i = 1, 2, ...) και LCP (Longest Common Prefix Array - περιέχει στη θέση i το μήκος του μακρύτερου κοινού προθέματος των suf[SA[i]] και suf[SA[i-1]]). Έπειτα, στο βήμα 3 αρχικοποιείται ο πίνακας MINUNIQUE με τιμές μικρότερες από οποιεσδήποτε έγκυρες τιμές μπορεί να του ανατεθούν. Στο βήμα 5, υπολογίζεται το μεγαλύτερο επαναλαμβανόμενο πρόθεμα κάθε επιθέματος. Η μικρότερη μη-επαναλαμβανόμενη συμβολοσειρά βρίσκεται μια θέση μετά, εκτός της περίπτωσης που το μεγαλύτερο επαναλαμβανόμενο πρόθεμα φτάνει στο τέλος του w. Προκειμένου να αποφευχθεί ο παραπάνω έλεγχος σε κάθε βήμα, επιτρέπεται στο MINUNIQUE να έχει n+1 στοιχεία, εκ των οποίων το τελευταίο θα αγνοείται. Τέλος, στο βήμα 6, ανανεώνεται ο πίνακας MINUNIQUE, αν βρεθεί μικρότερη συμβολοσειρά.

Κρίνεται απαραίτητο να σημειωθεί πως ο συγκεκριμένος αλγόριθμος έχει γραμμική χρονική πολυπλοκότητα.

[2]Ilie, Lucian & Smyth, Wf. (2011). Minimum Unique Substrings and Maximum Repeats

MINUNIQUE(w)

1. compute SA, LCP
2. **for** i **from** 1 **to** n **do**
3. MINUNIQUE[i] $\leftarrow 0$
4. **for** i **from** 1 **to** n **do**
5. $lcp \leftarrow \max(\mathsf{LCP}[i], \mathsf{LCP}[i+1])$
6. MINUNIQUE[SA[i] + lcp] $\leftarrow \max($MINUNIQUE[SA[i] + lcp], SA[i])
7. **return** MINUNIQUE

Εικόνα 29: Αλγόριθμος MINUNIQUE

4 Άσκηση Τέταρτη

Οι Ho-Leung Chan, Wing-Kai Hon, Tak-Wah Lam και Kunihiko Sadakane [3] εισήγαγαν μια καινούργια έννοια, το Compressed Suffix Tree. Έστω C = $T_1, T_2, ..., T_k$ είναι μια συλλογή κειμένων με συνολικό μήκος n. Μπορούμε να διατηρήσουμε ένα συμπιεσμένο δέντρο επιθεμάτων για το C, το οποίο χρησιμοποιεί χώρο O(n)-bit και υποστηρίζει τα ακόλουθα ερωτήματα σχετικά με το δέντρο επιθεμάτων για το C: εύρεση της ρίζας σε χρόνο O(1), και εύρεση του γονέα, του αριστερού παιδιού, του αριστερού αδελφού, του δεξιού αδελφού και του επιθέματος συνδέσμου ενός κόμβου σε χρόνο O(logn). Η ετικέτα ακμής και το φύλλο μπορούν να υπολογιστούν σε χρόνο O($\log_2$n). Η εισαγωγή ή διαγραφή ενός κειμένου T στο C μπορεί να γίνει σε O($|T|\log_2$n) χρόνο.

Ακόμα, η δενδροειδής δομή ενός suffix tree αναπαριστάται από μια λίστα παρενθέσεων ως εξής: διασχίζοντας το suffix tree με τον αλγόριθμο DFS, την πρώτη φορά που επισκέπτεσαι έναν κόμβο, προσθέτεις ένα "(" στη λίστα και στη τελευταία φορά που επισκέπτεσαι έναν, προσθέτεις ένα ")" αντίστοιχα στην λίστα. Σημειώνεται πως η λίστα των παρενθέσεων είναι ισορροπημένη και κάθε κόμβος στο δένδρο επιθεμάτων αντιπροσωπεύεται από ένα ζεύγος ταιριασμένων παρενθέσεων. Επομένως, μπορούμε να καθορίσουμε έναν κόμβο u στο δέντρο επιθεμάτων χρησιμοποιώντας τη θέση ανοικτής παρένθεσης που αντιπροσωπεύει το u.

Η παραπάνω λίστα των ισορροπημένων παρενθέσεων υποστηρίζει διάφορα ερωτήματα σχετικά με το suffix tree. Συγκεκριμένα, ο χαμηλότερος κοινός πρόγονος δύο κόμβων u και v είναι το double-enclose(u, v). Η τάξη ενός φύλλου u, η οποία είναι η λεξικογραφική σειρά του επιθέματος που αντιστοιχεί σε αυτό, είναι rank-leaf(u). Το i-οστό φύλλο, το οποίο είναι αυτό που αντιστοιχεί στο λεξικογραφικά i-οστό επίθεμα, δίνεται από το select-leaf(i). Το αριστερότερο φύλλο και το δεξιότερο φύλλο του υπό-δένδρου με ρίζα στο u μπορούν να βρεθούν από τις σχέσεις rank-leaf(u-1)+1 και rank-leaf(find-match(u)), αντίστοιχα. Κάθε μία από τις παραπάνω λειτουργίες απαιτεί O(logn) χρόνο.

Όσον αφορά στις πράξεις που μπορούν να γίνουν ισχύουν τα παρακάτω. Ας υποθέσουμε ότι έχουμε τη λίστα των ισορροπημένων παρενθέσεων, CSA (Compressed Suffix Array), FM- index [4] και LCP [5] που αναπαριστούν το suffix tree για μια συλλογή κειμένων C. Για να εισάγουμε ένα νέο κείμενο T στο C, ενημερώνουμε τις δομές δεδομένων ώστε να αντικατοπτρίζουν την αλλαγή ότι όλα τα επιθέματα του T εισάγονται στο suffix tree. Στη συνέχεια, εκτελούμε την ενημέρωση σε γύρους $|T|$ έτσι ώστε στον i-οστό γύρο, το i-οστό συντομότερο επίθεμα T [$|T|$ - i + 1 .. $|T|$] εισάγεται ως νέο φύλλο στο suffix tree. Κάθε γύρος περιλαμβάνει την ενημέρωση του καταλόγου των ισορροπημένων παρενθέσεων, CSA, FM-index και LCP. Έτσι, στο τέλος του i-οστού γύρου, οι δομές δεδομένων αναπαριστούν το συμπιεσμένο δένδρο επιθέτων για τη συλλογή $C \bigcup T[|T| - i + 1..|T|]$. Το βασικό μέλημα είναι η ενημέρωση του καταλόγου των ισορροπημένων παρενθέσεων και του LCP, η οποία γίνεται με τα ακόλουθα δύο βήματα: υπολογισμός των νέων πληροφοριών του suffix tree και την ενημέρωση των δομών δεδομένων σύμφωνα με το νέο suffix tree. Για το πρώτο βήμα, παρατηρούμε ότι το συμπιεσμένο δέντρο επιθεμάτων υποστηρίζει

[3] Dynamic Dictionary Matching and Compressed Suffix Trees

[4] CSA & FM-index: Αυτοί οι δείκτες είναι συμπιεσμένες εκδόσεις πινάκων επιθεμάτων που καταλαμβάνουν μόνο O(n) bits, αλλά υποστηρίζουν αποτελεσματική αναζήτηση προτύπων

[5] διατηρεί το μήκος του μεγαλύτερο κοινού προθέματος μεταξύ δύο γειτονικών φύλλων

τις λειτουργίες πλοήγησης στο κανονικό δέντρο επιθέτων, οπότε μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο του Weiner για να υπολογίσουμε τη θέση του νέου φύλλου.

5 Άσκηση Πέμπτη

Για την επίλυση του συγκεκριμένου ερωτήματος έχει δημιουργηθεί ο παρακάτω αλγόριθμος, ο οποίος ακολουθεί τα εξής βήματα:

- Δημιουργία ενός γενικευμένου δέντρου επιθεμάτων σύμφωνα με τον αλγόριθμο του Ukkonen (GST - Generalized Suff Tree) για μία συμβολοσειρά S και ένα πρότυπο P. Σε κάθε εσωτερικό κόμβο v του GST με 1 (2), σημειώνεται όταν υπάρχει φύλλο στο αντίστοιχο υπό-δέντρο του v, το οποίο αναπαριστά ένα suffix του S (P)
- Πέρασμα του GST από το πιο απομακρυσμένο φύλλο προς τα πάνω ψάχνοντας την αντίθετη συμβολοσειρά από αυτή που επιθυμούμε (π.χ. αν αναζητώ το πρότυπο P = aqra τότε θα αναζητήσω την συμβολοσειρά arqa)
- Σύγκριση κάθε φορά του υπό-μονοπατιού που βρισκόμαστε, προκειμένου να ταιριάζει με το πρότυπο που επιθυμούμε
- Αποθήκευση της θέσης/εσωτερικού κόμβου του υπό-μονοπατιού, στο οποίο εντοπίζεται το πρότυπο

Από τη στιγμή λοιπόν που τρέχω το δέντρο από κάτω προς τα πάνω το μονοπάτι, του οποίου το υπό-μονοπάτι του περιέχει το πρότυπο P, θα διέρχεται από τη ρίζα.

6 Άσκηση Έκτη

6.1 Ερώτημα Α

Το απλό πρόβλημα της καθολικής στοίχισης χωρίς συγγενική ποινή ασυμφωνίας θα λυθεί με χρήση τεχνικών δυναμικού προγραμματισμού.

Σε πρώτο επίπεδο εφαρμόζουμε ουσιαστικά τον αλγόριθμο των Needleman-Wunsch ως εξής. Κατασκευάζεται ένας γράφος οι κόμβοι του οποίου αντιπροσωπεύουν τα ζεύγη χαρακτήρων των δύο ακολουθιών. Καθένας από τους κόμβους διαθέτει ακμές που οδηγούν στον κάτω, κάτω δεξιά και δεξιά κόμβο του γράφου. Οι παραπάνω ακμές θα χαρακτηρίζονται από κάποια βάρη, τα οποία υπολογίζονται σύμφωνα με την πράξη που θα πραγματοποιείται κάθε φορά (αντικατάσταση ενός χαρακτήρα, προσθήκη κενού και αφαίρεση χαρακτήρα). Πρακτικά, λοιπόν, δημιουργείται ένας γράφος, από τον οποίο, ξεκινάμε από τον κάτω δεξιά κόμβο του και ακολουθούμε διαγώνια διαδρομή, όσο οι χαρακτήρες δεν διαφέρουν. Σε περίπτωση που υπάρξει ασυμφωνία, γίνεται επιλογή του ελάχιστου κόστους (αριστερά, πάνω ή πάνω αριστερά κόμβος) και εκτελείται η αντίστοιχη πράξη (προσθήκη κενού όσον αφορά στις δύο πρώτες περιπτώσεις ή ασυμφωνία χαρακτήρα στην στην τελευταία περίπτωση). Επιπλέον, πρέπει να αναφερθεί πως ο συγκεκριμένος αλγόριθμος υλοποιείται σε O(nm), όπου n, m τα μήκη των ακολουθιών.

Στη συγκεκριμένη περίπτωση όμως ο παραπάνω αλγόριθμος πρέπει να επεκταθεί, μιας και στον γράφο πρέπει να εισαχθούν νέες ακμές που θα ενώνουν διαγώνια κόμβους με μεγαλύτερες αποστάσεις. Γι' αυτό το λόγο, θα εφαρμόσουμε μια μέθοδο σαν κι αυτή των affine gap penalties. Αυτό σημαίνει πως θα δημιουργήσουμε έναν δεύτερο γράφο, σύμφωνα με τον οποίο θα αποθηκεύουμε τη πληροφορία των ασυμφωνιών. Κατά αυτό τον τρόπο, ορίζουμε τους παρακάτω κανόνες.

Σε περίπτωση που υπάρξει ανάγκη για προσθαφαίρεση, το κόστος ισούται με:

$$S_{i,j} = S_{i-1,j} - \rho$$

ή

$$S_{i,j} = S_{i,j-1} - \rho$$

Αν υπάρχει συμφωνία χαρακτήρων, το κόστος γίνεται:

$$S_{i,j} = S_{i-1,j-1} + 1$$

Όπως αναφέραμε και παραπάνω, σε περίπτωση ασυμφωνίας οδηγούμαστε στον δεύτερο γράφο και το κόστος γίνεται:

$$W_{i,j} = S_{i,j} - \rho$$

Σημειώνεται πως ο αλγόριθμος απαιτεί O(nm) για την κατασκευή των δύο γράφων, συνεπώς αυτή είναι και η τελική του πολυπλοκότητα.

6.2 Ερώτημα Β

Δημιουργήσαμε αλγόριθμο, ο οποίος υπολογίζει το μέγιστο κοινό πρόθεμα από ένα σύνολο συμβολοσειρών. Πιο συγκεκριμένα, χρησιμοποιήσαμε τη μέθοδο Divide and Conquer, μιας και η ιδέα είναι να διαιρέσουμε όλες τις συμβολοσειρές σε δύο μικρότερα σύνολα και στη συνέχεια να επεξεργαστούμε αναδρομικά αυτά τα σύνολα. Παρακάτω, παρουσιάζεται ο κώδικας:

```
# All possible pairs in List
# Using list comprehension + enumerate()
def getEachPair(list):

    for index, X in enumerate(list):
        for Y in list[index+1:]:
            res = LCP(X, Y)
            print("The longest common prefix between", X, "and", Y, "is", res)


# function to find the longest common prefix (LCP)
# between two strings
def LCP(X, Y):

    k = l = 0

    while k < len(X) and l < len(Y):
        if X[k] != Y[l]:
            break
        k = k + 1
        l = l + 1

    return X[:k]
```

7 Άσκηση Επτά

Ο Eugene W. Myers [6] προσπάθησε να αποδείξει ότι D(n,m)=m+n-2u ή u=(m+n-D(n,m))/2 βασιζόμενος στα εξής. Ένα edit script για δύο συμβολοσειρές A και B είναι ένα σύνολο εντολών και διαγραφής που μετατρέπουν το A σε B. Η διαγραφή "xD" (deletion) διαγράφει το σύμβολο a_x από το A. Η εντολή εισαγωγής "xIb$_1$, b$_2$, ... b_t" (insertion) εισάγει την ακολουθία συμβόλων b$_1$, ... b$_t$ αμέσως μετά το a_x. Κάθε ίχνος [7] αντιστοιχεί μοναδικά σε ένα edit script. Έστω (x_1, y_1)(x_2, y_2)...(x_L, y_L) είναι ίχνος με $y_0 = 0$ και $y_L + 1 = M + 1$. Το script αποτελείται από τις εντολές "xD" για x $\notin$ x_1, x_2, ..., x_L, και "x_kIb$_{y_k} + 1$, ..., b$_{y_k} - 1$" για k τέτοιο ώστε $Y_k + 1 < y_k + 1$. Το script διαγράφει σύμβολα N-L και εισάγει σύμβολα M-L. Συνεπώς, για κάθε ίχνος μήκους L υπάρχει ένα αντίστοιχο script μήκους D = N + M - 2L.

Η μέγιστη κοινή υπό-συμβολοσειρά μεταξύ δύο συμβολοσειρών S_1, S_2 μπορεί να βρεθεί σε γραμμικό χρόνο χρησιμοποιώντας ένα GST (generalized suffix tree). Πιο συγκεκριμένα,

[6] Eugene W. Myers, An O(ND) Difference Algorithm and Its Variations

[7] Ένα ίχνος μήκους L είναι μια ακολουθία σημείων αντιστοίχισης L, (x_1, y_1)(x_2, y_2)...(x_L, y_L), έτσι ώστε $x_i < x_i + l$ και $y_1 < y_i + l$ για διαδοχικά σημεία (x_i, y_i) και (x_{i+1}, y_{i+1} για i ∈ L[1, L-1)

1. Κατασκευή ενός Generalized Suffix Tree

2. Σημείωσε σε κάθε εσωτερικό κόμβο v του GST με 1 (2), όταν υπάρχει φύλλο στο αντίστοιχο υπό-δέντρο του v, το οποίο αναπαριστά ένα suffix του S_1 (S_2).

Κάθε εσωτερικός κόμβος που σημειώνεται ταυτόχρονα με 1 και 2, δηλώνει την ύπαρξη ενός κοινού substring μεταξύ των συμβολοσειρών S_1 και S_2. Ο κώδικας που υλοποιεί το ζητούμενο περιλαμβάνεται στο ίδιο Archive με αυτό το pdf και όνομα suffix_trees.

8 Άσκηση Οκτώ

Η ακολουθία της spike (ακίδα) πρωτεΐνης του κορονοϊού έχει ληφθεί από εδώ και βρίσκεται στο φάκελο MerosA_Askisi8 με όνομα spike ενώ αντίστοιχα για τον συναφές κορονοϊό Bat-RaTG13 έχει ληφθεί από εδώ και βρίσκεται στο φάκελο MerosA_Askisi8 με όνομα bat. Στον φάκελο υπάρχει το αρχείο main στο οποίο εκτελείται ο ζητούμενος κώδικας και προκύπτουν τα αντίστοιχα αποτελέσματα.

9 Άσκηση Εννέα

9.1 Ερώτημα Α

Εφαρμόζουμε τον αλγόριθμο Needleman-Wunsch, για τον οποίο ισχύουν τα εξής:

$$D_{i,j} = max \begin{cases} D_{i-1,j-1} + s(v_i, w_j) \\ D_{i-1,j} + s(v_i, -) \\ D_{i,j-1} + s(-, w_j) \end{cases} = max \begin{cases} D_{i-1,j-1} + 1, v_i = w_j \\ D_{i-1,j-1} - 1, v_i \neq w_j \\ D_{i-1,j} - 1, w_i = - \\ D_{i,j-1} - 1, v_i = - \end{cases}$$

Σύμφωνα με τα παραπάνω προκύπτει ο εξής πίνακας:

$D_{i,j}$		G	A	T	C	G	T	G	A	A	T	T
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
G	-1	⇖1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
G	-2	0	⇖0	-1	-2	-1	-2	-3	-4	-5	-6	-7
T	-3	-1	-1	⇖1	0	-1	0	-1	-2	-3	-4	-5
T	-4	-2	-2	⇑0	0	-1	0	-1	-2	-3	-2	-3
C	-5	-3	-3	-1	⇖⇑1	0	-1	-1	-2	-3	-3	-3
G	-6	-4	-4	-2	0	⇖2	1	0	-1	-2	-3	-4
T	-7	-5	-5	-3	-1	1	⇖3	2	1	0	-1	-2
G	-8	-6	-6	-4	-2	0	2	⇖4	3	2	1	0
G	-9	-7	-7	-5	-3	-1	1	⇑3	3	2	1	0
A	-10	-8	-6	-6	-4	-2	0	2	⇖⇑4	⇐⇖4	⇐3	⇐2

Score = 2

G A T – C G T G – A - - -

G G T T C G T G G A A T T

9.2 Ερώτημα Β

Εφαρμόζουμε τον αλγόριθμο Smith - Waterman, για τον οποίο ισχύον τα εξής:

$$S_{i,j} = max \begin{cases} S_{i-1,j-1} + s(v_i, w_j) \\ S_{i-1,j} + s(v_i, -) \\ S_{i,j-1} + s(-, w_j) \\ 0 \end{cases} = max \begin{cases} S_{i-1,j-1} + 1, v_i = w_j \\ S_{i-1,j-1} - 1, v_i \neq w_j \\ S_{i-1,j} - 1, w_i = - \\ S_{i,j-1} - 1, v_i = - \\ 0 \end{cases}$$

Σύμφωνα με τα παραπάνω προκύπτει ο εξής πίνακας:

$D_{i,j}$		G	A	T	C	G	T	G	A	A	T	T
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	1	0	0	0	1	0	1	0	0	0	0
G	0	1	0	0	0	1	0	1	0	0	0	0
T	0	0	0	1	0	0	2	1	0	0	1	1
T	0	0	0	1	0	0	1	1	0	0	1	2
C	0	0	0	0	⇐⇑2	1	0	0	0	0	0	1
G	0	1	0	0	1	⇖3	2	1	0	0	0	0
T	0	0	0	1	0	2	⇖4	3	2	1	1	1
G	0	1	0	0	0	1	3	⇖5	4	3	2	1
G	0	1	0	0	0	1	2	⇑4	4	3	2	1
A	0	0	2	1	0	0	1	3	⇐⇑5	⇐5	4	3

Score = 5

C G T G - A -

C G T G G A A