



Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

ΕΙΣΑΓΩΓΗ ΣΤΗ ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ

Πρώτο Σύνολο Ασκήσεων 2023-2024

Νίκος Αυγέρης
A.M. 1067508

Ασπασία Τζερμιά
A.M. 1067455

Πάτρα, 2024

Ερώτημα 1

(i)

- Το πρώτο πρόβλημα είναι δεδομένης μίας αλληλουχίας DNA να μετρήσουμε τις εμφανίσεις του κάθε αμινοξέος.

Ας δούμε πρώτα με χρήση biopython:

```
1  from Bio.Seq import Seq
2
3  my_seq = Seq("AGCTTTCAATTCTGACTGCAACGGGCAATATGTCTCTGTGATTAAAAAAAGAGTGTCTGATAGCAGC")
4
5  print("Adenine :" + str(my_seq.count("A")))
6  print("Cytosine :" + str(my_seq.count("C")))
7  print("Guanine :" + str(my_seq.count("G")))
8  print("Thymine :" + str(my_seq.count("T")))
```

Adenine :20
Cytosine :12
Guanine :17
Thymine :21

Το Seq είναι μια δομή που βολεύει όταν διαχειρίζομαστε αλληλουχίες διότι μας επιτρέπει να χρησιμοποιήσουμε έτοιμες συναρτήσεις. Η str() είναι απαραίτητη διότι έχουμε δύο διαφορετικού τύπου αντικείμενα.

Το DNA stats είναι online εργαλείο της σουίτας Sequence Manipulation Suite. Όπως βλέπουμε μας δείχνει και τα ποσοστά ύπαρξης του κάθε αμινοξέος.

Pattern:	Times found:	Percentage:
g	17	24.29
a	20	28.57
t	21	30.00
c	12	17.14
n	0	0.00
u	0	0.00
r	0	0.00
y	0	0.00
s	0	0.00
w	0	0.00
k	0	0.00
m	0	0.00
b	0	0.00
d	0	0.00
h	0	0.00

- Το δεύτερο πρόβλημα είναι να αναζητήσουμε όλες τις καταχωρίσεις ενός γένους που έγιναν μεταξύ 2 ημερομηνιών, στην GenBank του NCBI.

Βάλαμε φίλτρο για αναζήτηση στην GenBank, με το “nucleotide”.

Η συνάρτηση **Bio.Entrez.esearch()** της biopython μπορεί να ψάξει όλες τις βάσεις δεδομένων του NCBI.

```

1
2
3 from Bio import Entrez
4 Entrez.email = "up1067508@upnet.gr"
5 handle = Entrez.esearch(db="nucleotide", term="Anthoxanthum" + "[Organism] AND (2003/7/25 :
2005/12/27 [Publication Date])")
6 record = Entrez.read(handle)
7 print("\n[GenBank gene database]:", record["Count"])

```

- Το επόμενο πρόβλημα είναι να δώσουμε στην genBank 3 id και να μας επιστραφεί η μικρότερη αλληλουχία σε FASTA format. Θα ξεκινήσουμε με την Biopython. Στην αρχή δοκιμάσαμε τον παρακάτω κώδικα ο οποίος δεν χρησιμοποιεί όλα τα εργαλεία που προτείνονται στο Rosalind. Παρόλο που δίνει σωστό αποτέλεσμα αυτός ο κώδικας μετράει και το μέγεθος της περιγραφής μαζί με αυτό της αλληλουχίας.

```

1 from Bio import SeqIO
2 from Bio import Entrez
3
4
5 def fetch_and_convert(ids):
6     fasta_records = []
7
8     for id in ids:
9         handle = Entrez.efetch(db="nucleotide", id=id, rettype="gb", retmode="text")
10    record = SeqIO.read(handle, "genbank")
11    handle.close()
12
13    # format to FASTA
14    fasta_record = record.format("fasta")
15    fasta_records.append(fasta_record)
16
17    # return shorter
18    return min(fasta_records, key=len)
19
20
21 Entrez.email = "up1067508@upnet.gr"
22 ids = ["FJ817486", "JX069768", "JX469983"]
23 print(fetch_and_convert(ids))

```

>JX469983.1 Zea mays subsp. mays clone UT3343 G2-like transcription factor mRNA, parti
al_ids:
ATGATGTATCATGGAAAGATTTCCTG6CCCTTGGCTCGCAGAGGGCACAGGATAAT
GAGCATGCAAGTAAATTGAGGTTATTGGTGACCCCCAACATAGAACCTCTGTAATCT
GTAGGAAAGTGGGAAACAGGGCTACGGGTGACATCGGATCTTATGCTTTGGAT
GCCATGCGCAGCTTGGTGAC CAGACAGAGCTACACTAAAGGGTTCTACTGGATG
GGTGTACCGAGGATCACAAATTATCATGTGAAGAGGCATCGAGAAGTATGCGCTTGCA
AAGTATGCGCTTGGTGAC CAGACAGAGCTACACTAAAGGGTTCTACTGGATG
ATGGTGATGCGCTTGGTGAC CAGACAGAGCTACACTAAAGGGTTCTACTGGATG
ATGGAGGTTCTGAGACGGCTACATGAGCAGCTGAGGTTMAAGGAACTGACAATGAA
ATTTGAAAGCACAAGGGAGATCTTGAGATGATCATTTGAGGAGACAACAAGTTGGTG
TCAAATTAAAGGCTTCTGAGGGATCAGAAGCTTTCTGATTTCTCACTTCAGCTGATGACTAC
CCAGAGAGCATGCAACCTTCTCCAAGAAACAGGATAGACGGCTTACCCAGATTC
GAGCGCGATACAAACACACTGAAATTGAACTTGTGCGTGGGATCATGGC
ATTGCGTTCAGGCGCTTGTGAGGAGTTCAAAGCAGGCCCTGATGAGCAAAGT

Άρα αλλάζουμε τον κώδικα σε:

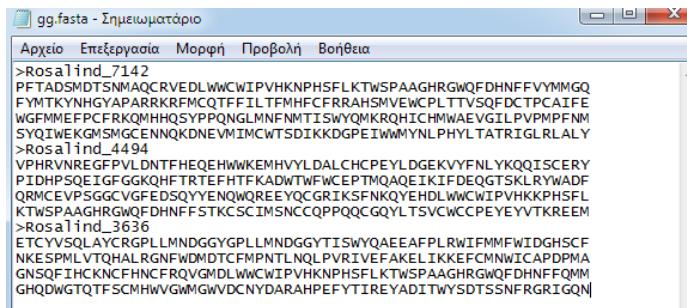
```

1
2
3 from Bio import Entrez
4 from Bio import SeqIO
5
6 Entrez.email = "up106750@upnet.gr"
7 handle = Entrez.efetch(db="nucleotide", id=["JX1817486, JX069768, JX469983"], rettype="fasta")
8 records = list(SeqIO.parse(handle, "fasta")) # Get the list of SeqIO objects in FASTA format
9
10 length = [0, 0, 0]
11 length[0] = len(records[0].seq) # First record ID
12 length[1] = len(records[1].seq)
13 length[2] = len(records[2].seq)
14
15 last = min(length)
16 if last == length[0]:
17     print(records[0])
18 elif last == length[1]:
19     print(records[1])
20 elif last == length[2]:
21     print(records[2])

```

ID: JX469983.1
Name: JX469983.1
Description: JX469983.1 Zea mays subsp. mays clone UT3343 G2-like transcription factor mRNA, partial cds
Number of features: 0
Seq('ATGATGTATCATGCGAAGAATTCTGTCCTTGCCTCGAGAGGGCACAG...TCA')

Και πλέον ο κώδικας είναι σωστός διότι μετράει μόνο το μήκος της αλληλουχίας.
Πρώτα φτιάχνουμε ένα fasta αρχείο με τις ακολουθίες μας.



```

>Rosalind_7142
PFTADSMDSNTMAQCRVEDLWWCWIPIVHKNPHSFLKTWSPAAGHRGWQFDHNFFVYMMQ
FYMTKYNHGYAPARRKRCMTCOTFFILTFMHFCRRAHSMVEWCPITTVSQFDCTPCAIFE
WGFMMEFPCFRKQMHQSYPQNLMLNFNTMISWYQMKRQHICHMWAEVGILPVPMFNM
SYQIWEKGMSMGCCENNQKDNEVMICWTSDIKKDGEPEIWMYLPHYLTATRIGRLALY
>Rosalind_4494
VPHNVNREGFPVLNTFHEQEHWKEMHVYLDALCHCPEYLDEKEVYFNLYKQQISCERY
P1DHP5QE1GPGQHFTRTEHTFKADWTFWCEPTMQAQEETKIFDEQGTSKLRYWADF
QRMCCEVPSGGCVGFEDSQYYENQWQREEEYQCGRIKSFNKQYEHDLMWCWIPIVHKPHSFL
KTMSPALGHRGWQFDHNFFSTKCSCIMSNCCQPQQCGQYLTSCWCPCPEYEVYTKREEM
>Rosalind_3636
ETCYVSQLAYCRGPLLMNDGGYGPILLMNDGGYTISWYQAEAFPLRWIFMMFWIDGHSCF
NKEPSMLVTQHALRGNFwMDTCFMPNTLNQLPVRIVEFAKELIKKECMNWICAPDPM
GSNQF1HCKNCFHNCFRQVGMDLWWCWIPIVHKNPHSFLKTWSPAAGHRGWQFDHNFFQMM
GHQDWGTTFSCHMHWVGWVGDCNYDARAHPEFYTIREYADITWSDTSSNFRGRIGQN

```



MEME
Multiple Em for Motif Elicitation

Version 5.5.5

Data Submission Form

Perform motif discovery on DNA, RNA, protein or custom alphabet datasets.

Select the motif discovery mode [?]

Classic mode Discriminative mode Differential Enrichment mode

Select the sequence alphabet

Use sequences with a standard alphabet or specify a custom alphabet. [?]

DNA, RNA or Protein Custom Δεν Επιλέγεται κάποια αρχείο

Input the primary sequences

Enter sequences in which you want to find motifs. [?]

Επιλογή αρχείου | gg.fasta [?]

Select the site distribution

How do you expect motif sites to be distributed in sequences? [?]

Zero or One Occurrence Per Sequence (zoops)

Select the number of motifs

How many motifs should MEME find? [?]

Input job details

(Optional) Enter your email address. [?]

(Optional) Enter a job description. [?]

Advanced options

Note: if the combined form inputs exceed 50MB the job will be rejected.

Version 5.5.5 Please send comments and questions to: meme-suite@ucv.edu Powered by Opal

- Στην επόμενη άσκηση ψάχνουμε να βρούμε αν 2 αλληλουχίες έχουν κοινούς προγόνους. Έτσι θέλουμε να κάνουμε ευθυγράμμιση και να δούμε τι σκορ λαμβάνουν. Θα χρησιμοποιήσουμε τα εργαλεία Needle και Stretcher. Μια διαφορά τους είναι ότι το Stretcher χρησιμοποιεί διαφορετική βαθμολόγηση όταν τελειώνουν τα κενά ενώ το Needle δεν το έχει ως προεπιλογή. Επίσης στις επιλογές που μας δίνουν τα δύο εργαλεία έχουν

διαφορετικά νούμερα. Πχ το Needle έχει στην αναζήτησή μας GAP EXTEND=0.5 ενώ το Stretcher έχει επιλογές 1,2,3 κτλ.

Με το Needle, βρίσκουμε το Fasta Format των ID που θέλουμε να εισάγουμε.

Input sequence ⓘ Sequence type
 Protein DNA

Paste your sequence here - or use the example sequence

```
>JX205496.1 Orchis italica C-class MADS-box-like protein (AG) mRNA, complete cds
ATTCTTCCGTTCCTCGCCCTCCCCTCTGCCCTCCCACCATCTCTCTCTTTATTTGAACCAAAT
GGCAATGATTGAGGTAAAGAACCAAGCGCTCCCTCTGGCGCTTCATCAGAAAGAACAGCAG
AGAACAGCATCGAACCTATCAGCTTAGCGGCCAACAAATTCAAGTCACAGAGCAACAGAGGAGGACAT
GATGGAGCCGAAGGAAAAGATGGAAAGGGAAAGATAGAGATAAGAGGATAGAAAATACCACGAACAGG
CAAGTCACCTTTGTAAGCGCCCAATGGCCTCTCCTCCTCCTGCCCCCTACGGAGTATGCCAATAACAGTGA
CCGAGGTGCGCCCTGTCATCTCTCACCCGTTGGCCGCTTACGGAGTATGCCAATAACAGTGAAGGG
```

Επιλογή αρχείου Λενε επιλέχθηκε κανένα αρχείο.

Paste your sequence here - or use the example sequence

```
>JX469991.1 Zea mays subsp., mays clone UT3351 ABI3VP1-type transcription factor mRNA, partial cds
ATGGAAAGCCTCCGCCGGCTCGGCCACCGCACTCCAAAGAGAACCCGCCGGACGACAGCTGGCACATGG
GAGGGGGCCCCGCCGGAGGAGATCGAGGGGGAGGCCGGGGATGATTCTATGTTCGCTGAAGACACGTTCCC
CTCCCTCCCGGACTTCTTGCCCTGCCTGCGCTCAGCTCCACCTTCGTCAGCTTCAACTCTCGTCAAC
TCCTCAGGGCTACACCAACACGGCAGGAAGAGCCGGAGGAGCTCGCTGCGCTTCCATGCCGTGGGACTC
AAGGGTTTGTGCGCTGATGACATGACCGACTCTCGACTTCCATGCCGTGGGACTC
CGAGCGCTTCCGGGGGGTAGCATGATGCTAGAGAACGCCATGTCGGCGCCGCCAGCCGGTGGCGAC
```

Επιλογή αρχείου Λενε επιλέχθηκε κανένα αρχείο.

[Use the example](#) [Clear sequence](#) [More ex](#)

C https://www.ebi.ac.uk/j.dispatcher/psa/emboss_needle/summary?jobId=emboss_needle-I20240501-081009-0580-24884231-p1m

```

# -stdout
# -asequence emboss_needle-I20240501-081009-0580-24884231-p1m.asequence
# -bsequence emboss_needle-I20240501-081009-0580-24884231-p1m.bsequence
# -datafile EDNAFULL
# -gapopen 10.0
# -gapextend 0.5
# -endopen 10.0
# -endextendl 0.5
# -aformat3 pair
# -snucleotide1
# -snucleotide2
# Align_format: pair
# Report_file: stdout
#####
=====#
# Aligned_sequences: 2
# 1: JX205496.1
# 2: JX469991.1
# Matrix: EDNAFULL
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 2325
# Identity: 706/2325 (30.4%)
# Similarity: 706/2325 (30.4%)
# Gaps: 1409/2325 (60.6%)
# Score: 935.0
#
=====#

```

JX205496.1	1	-----	0
JX469991.1	1	ATGGAAGCCTCCGCCGGCTCGCCACCCGACTCCAAAGAGAACCCGCC	50
...

Έπειτα πάμε στο Stretcher.

https://www.ebi.ac.uk/jdispatcher/psa/emboss_stretcher/summary?jobId=emboss_stretcher-l20240501-082500-0799-9088751-p1m

```

# Report_file: stdout
#####
=====
#
# Aligned_sequences: 2
# 1: JX205496.1
# 2: JX469991.1
# Matrix: EDNAFULL
# Gap_penalty: 10
# Extend_penalty: 1
#
# Length: 2200
# Identity: 782/2200 (35.5%)
# Similarity: 782/2200 (35.5%)
# Gaps: 1159/2200 (52.7%)
# Score: 257
#
=====
JX205496.1      1 AT-----TC-----        4
                ||| |
JX469991.1      1 ATGGAAGCCTCCGGCGCTGTCGCCACCGCACTCCAAAGAGAACCGCC 50
JX205496.1      5 -----        4

```

Εδώ βρήκαμε 35.5% ομοιότητα. Άρα βρήκαμε μεγαλύτερη επιτυχία παρόλο που έχοντας λιγότερες παραμέτρους.

Το επόμενο ζητούμενο είναι να μετατρέψουμε ένα FASTQ αρχείο σε FASTA μορφή.

Χρησιμοποιούμε πρώτα το εργαλείο [Sequence conversion website](#) και βλέπουμε ότι μας δίνει το επιθυμητό output.

https://sequenceconversion.bugaco.com/converter/biology/sequences/fastq_to_fasta.php

Fastq to Fasta Sequence Converter

Provided by [bugaco.com](#)

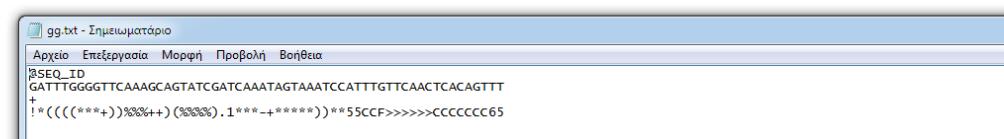
Convert file from:

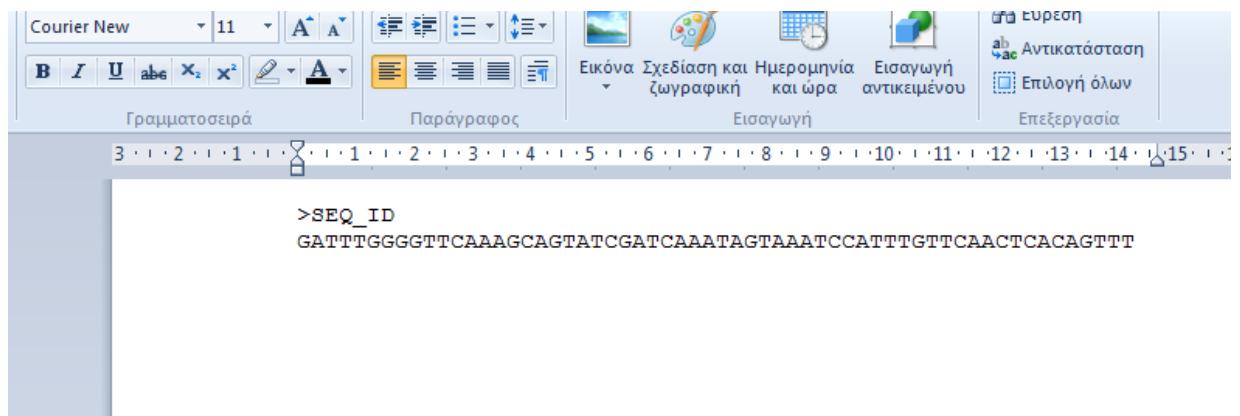
to

Alphabet: None DNA RNA Protein Nucleotide

Επιλογή αρχείου gg.txt

Your file will automatically download when conversion is finished.





Αποφεύγουμε να χρησιμοποιήσουμε το BlastStation διότι απαιτεί λήψη και αγορά.

Θα δοκιμάσουμε όμως με galaxy.

Βλέπουμε ότι δίνει το ίδιο ακριβώς αποτέλεσμα.

Started tool **FASTQ to FASTA** and successfully added 1 job to the queue.
It produces this output:
• 2: FASTQ to FASTA on data 1: FASTA
You can check the status of queued jobs and view the resulting data by refreshing the History panel. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

Started tool **FASTQ to FASTA** and successfully added 1 job to the queue.
It produces this output:
• 2: FASTQ to FASTA on data 1: FASTA
You can check the status of queued jobs and view the resulting data by refreshing the History panel. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

History + ⇔ ⚙

search datasets

Unnamed history

196 B 2

2: FASTQ to FASTA on data 1:

FASTA

Add Tags

1 sequences

format **fasta**, database ?

Input: 1 reads.
Output: 1 reads.
discarded 0 (0%) low-quality reads.

?

```
>1  
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTG
```

1: gg.txt

Add Tags

1 sequences

format **fastqsanger**, database ?

uploaded fastqsanger file

Επίσης θα δοκιμάσουμε και σε Biopython.

Input format: **fastq** FASTQ files are a bit like FASTA files but also include sequencing qualities. In Biopython, 'fastq' refers to Sanger style FASTQ files which encode PHRED qualities using an ASCII offset of 33. See also the incompatible 'fastq-solexa' and 'fastq-illumina' variants.

Output format: **fasta** This refers to the input FASTA file format introduced for Bill Pearson's FASTA tool, where each record starts with a '>' line. Resulting sequences have a generic alphabet by default.

How to convert from fastq to fasta ?

You can also convert between these formats by using command line tools.

- On Windows install [WSL](#), on Mac or Linux start terminal
- Install [BioPython](#)
- Run following script:

```
from Bio import SeqIO

records = SeqIO.parse("THIS_IS_YOUR_INPUT_FILE.fastq", "fastq")
count = SeqIO.write(records, "THIS_IS_YOUR_OUTPUT_FILE.fasta", "fasta")
print("Converted %i records" % count)
```

Or you can use this site as online fastq to fasta converter by selecting your formats & file.

[Sequence Converter Home page](#)

The screenshot shows a code editor interface with two tabs: 'main.py' and 'ggcopy.fasta'. The 'main.py' tab contains the following Python code:

```
from Bio import Entrez
from Bio import SeqIO
from Bio import SeqIO
records = SeqIO.parse("gg.fastq", "fastq")
count = SeqIO.write(records, "ggcopy.fasta", "fasta")
print("Converted %i records" % count)
```

The 'ggcopy.fasta' tab shows the converted FASTA sequence:

```
>SEQ_ID
GATTTGGGTTCAAAGCAGTATCGATAAATCCATTGTTCAACTCACAGTTT
```

The status bar at the bottom right indicates 'Converted 1 records'.

Θα χρησιμοποιήσουμε το εργαλείο fastqc που βρίσκεται online στο site galaxy. Στόχος η ανάλυση ποιότητας ενός FASTQ αρχείου.

The screenshot shows a terminal window with a FASTQ sequence. The sequence starts with '@Rosalind_0041' and includes quality scores represented by ASCII characters. The sequence is:

```
@Rosalind_0041
GGCCGGTCTATTTACGTTCTCACCCGACGTGACGTACGGTCC
+
6.3536354;.151<211/0?:6/-2051)-*"40/.,+%
@Rosalind_0041
TCGTATGCGTAGCAGGAAGTGAACATCCAGGAT
+
AH@FGGGJ<GB<<9:GD=D@GG9=?A@DC=;:?:>839/4856
@Rosalind_0041
ATTCCGGTAATTGGCGTGAATCTGTTCTGACTGATAGAGACAA
+
@DJEJEA?JHJ@8?F?IA3=;8@c95=;=?;>D/:;74792.
```

Δυστυχώς με αυτό το εργαλείο δεν καταφέραμε να δούμε πολλά.

The screenshot shows a software interface for analyzing sequencing data. It consists of two main vertical panels.

Top Panel (FASTQC Results):

- Header: "Add Tags" with a tag icon.
- Text: "320 lines", "format txt, database ?".
- Text area:

```
null
Picked up _JAVA_OPTIONS: -
```
- Control icons: magnifying glass, refresh, info, save, etc.
- Table:

Measure	Value
Filename	g1g_fastq
File type	Conventional base calls

Bottom Panel (FASTQ File View):

- Header: "3: g1g.fastq" with edit and delete icons.
- Header: "Add Tags" with a tag icon.
- Text: "309 bytes version=1.0", "format fastg, database ?".
- Text area: "uploaded fastg file".
- Control icons: magnifying glass, refresh, info, save, etc.
- Text area:

```
@Rosalind_0041
GGCCGGTCTATTTACGTTCTCACCCGACGTGACGTACGGTCC
+
6.3536354;.151<211/0?::6/-2051)-+"40/.,,+%
@Rosalind_0041
```

Οπότε θα ψάξουμε και στο Filter FASTQ.

Define Base Offsets as

Absolute Values

Use Absolute for fixed length reads (Illumina, SOLiD)

Use Percentage for variable length reads (Roche/454)

Offset from 5' end *

0

Values start at 0, increasing from the left

Offset from 3' end *

0

Values start at 0, increasing from the right

Aggregate read score for specified range *

mean of scores

Keep read when aggregate score is *

<=

Quality score *

28,0

+ Insert Quality Filter on a Range of Bases

▶ Run Tool

- Help



Εδώ βλέπουμε το επιθυμητό αποτέλεσμα.

Τέλος δοκιμάζουμε με Biopython.

```

from Bio import SeqIO
def count_low_quality_reads(fastq_file, quality_threshold):
    count = 0
    for record in SeqIO.parse(fastq_file, "fastq"):
        avg_quality = sum(record.letter_annotations["phred_quality"]) / len(record.letter_annotations["phred_quality"])
        if avg_quality < quality_threshold:
            count += 1
    return count
# Test the function with your data
fastq_data = """@Rosalind_0041
GCGCGGTCTATTACGTTCTACCCGACGTGACGTACGGTCC
+
6.3536354;.151<211/0?:;6/-2051)-*\"40/.,+%
@Rosalind_0041
TCGATGCGTAGCACTTGGTACAGGAAGTGAAACATCCAGGAT
+
AH@F0GGJ<GB<<9:G0=D@GG9=?A@DC=;:?>839/4856
@Rosalind_0041
ATTCGGTAATTGGCGTGAATCTGACTGATAGAGACAA
+
@DJEJA?JHJ@?F?IA3=;@C95=;=?;D@/:;74792.===
with open("fastq_data.fastq", "w") as f:
    f.write(fastq_data)
print(count_low_quality_reads("fastq_data.fastq", 28))

```

- Τώρα θα μεταφερθούμε στο επόμενο πρόβλημα. Μέσω της σελίδας <http://www.bioinformatics.org/sms2/translate.html>, του εργαλείου μετάφρασης του SMS 2. Βλέπουμε ότι για την ακολουθία dna που δώσαμε, παράγεται η ζητούμενη πρωτεΐνη με το γεννετικό κώδικα 1. Αυτός ο τρόπος είναι αργός καθώς χρειάζεται να κάνουμε πολλά ερωτήματα μέχρι να πετύχουμε το σωστό κώδικα.

← → C vN Μη ασφολής | http://www.bioinformatics.org/sms2/translate.html

Sequence Manipulation Suite: Translate

Translate accepts a DNA sequence and converts it into a protein in the reading frame you specify. Translate

Paste a raw sequence or one or more FASTA sequences into the text area below. Input limit is 200,000,000

```
ATGGCCATGGCGCCCAGAACTGAGATCAATAGTACCCGTATTAACGGGTGA
```

Submit Clear Reset

- Translate in reading frame 1 on the direct strand.
- Use the standard (1) genetic code.

Θα δοκιμάσουμε να λύσουμε το ίδιο πρόβλημα στην Biopython.

```
main.py > ...
1  from Bio.Seq import translate
2
3  # Given DNA string
4  dna_string = "ATGGCCATGGCGCCCAGAACTGAGATCAATAGTACCCGTATTAACGGGTGA"
5
6  # Loop over the genetic code tables
7  for gen_table in range(1, 26): # There are 25 known genetic code tables
8      protein_string = translate(dna_string, table = gen_table, to_stop=True) # table -> function of translate
9      if protein_string == "MAMAPRTEINSTRING":
10         print("The index of the genetic code variant used for translation is "+ str(gen_table))
11         break
12
```

The index of the genetic code variant used for translation is 1

- Τώρα πάμε στο επόμενο πρόβλημα, την εύρεση ποιότητας μέσω του fastg quality filter. Η απάντηση είναι η αναμενόμενη, 2.

Filter by quality (Galaxy Version 1.0.2+galaxy2)

Tool Parameters

Input FASTQ file *

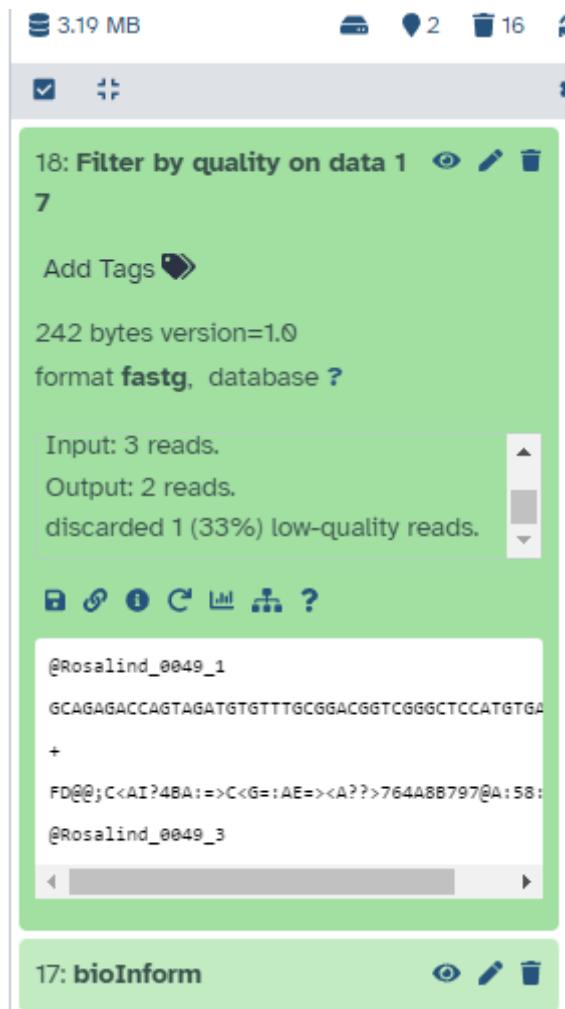
accepted formats *

Quality cut-off value *

Percent of bases in sequence that must have quality equal to / higher than cut-off value *

90

Run Tool



- Στο επόμενο πρόβλημα κοιτάμε αν οι 2 ακολουθίες μας είναι ίδιες με τις αντίστροφες συμπληρωματικές τους. Όπως βλέπουμε μόνο η 1 είναι.
Χρησιμοποιήσαμε το [Reverse Complement](#) εργαλείο της sms2. Θα χρησιμοποιήσουμε και Biopython.

Sequence Manipulation Suite:

Reverse Complement

Reverse Complement converts a DNA sequence into its reverse, complement, or reverse-complement counterpart. The character case of each input sequence is maintained. You may want to work with the reverse-complement of a sequence.

Paste the raw sequence or one

```
>Rosalind_64
ATAT
>Rosalind_48
GCATA
```

Submit Clear Reset

• reverse-complement ▾

*This page requires JavaScript.
*You can mirror this page or use

Sun 14 Jun 00:37:00 2020
Valid XHTML 1.0; Valid CSS

Sequence Manipulation Suite - Avast Secure Browser

Reverse Complement results

```
>Rosalind_64 reverse complement
ATAT

>Rosalind_48 reverse complement
TATGC
```

```
main.py > ...
1  from Bio.Seq import Seq
2
3  dna_strings = {
4      "Rosalind_64": "ATAT",
5      "Rosalind_48": "GCATA"
6  }
7
8  # Initialize a counter for the matches
9  matches = 0
10
11 # Iterate over the DNA strings
12 for name, dna_string in dna_strings.items():
13     # Create a Seq object for the DNA string
14     my_seq = Seq(dna_string)
15
16     # Get the reverse complement of the DNA string
17     reverse_complement = my_seq.reverse_complement()
18
19     # If the DNA string matches its reverse complement
20     if str(my_seq) == str(reverse_complement):
21         matches += 1
22
23
24 print("Number of DNA strings that match their reverse complements:", matches)
```

Number of DNA strings that match their reverse complements: 1

- Στο επόμενο πρόβλημα θα χρησιμοποιήσουμε το Lalignt εργαλείο του Ebi. Στο εργαλείο αυτό είναι δύσκολο να εντοπίσουμε αλληλουχίες που διαφέρουν μόνο 3 ζευγάρια. Στις παραμέτρους βάζουμε gap opening=0 ώστε σε περίπτωση που υπάρχει κάποια διαγραφή να μην την μετρήσει πιο δύσκολα καθώς έχουμε δικαίωμα 3 αλλαγών. Στο gap extend βάζουμε -2, ώστε να μην επεκταθεί πολύ το κενό. Στην τιμή E βάζουμε χαμηλή τιμή διότι θέλουμε μια πιο ακριβή αντιστοίχηση(δικαιούμαστε μόνο 3 λάθη).

Επιλογή αρχείου Δεν επιλέχθηκε κανένα αρχείο.

[Use the example](#) [Clear sequence](#) More example input

Parameters	MATRIX ⓘ	GAP OPEN ⓘ	GAP EXTEND ⓘ	E(0) THRESHOLD ⓘ	OUTPUT FORMAT ⓘ
	BLOSUM50	0	-2	1.0	MARKX 0

GRAPHICS ⓘ
yes

[Less options ^](#)

Submit Title

<https://www.ebi.ac.uk/jdispatcher/psa/lalign/summary?jobId=lalign-I20240508-115512-0330-98093280-p1m>

```

threshold: E(1) < 1 score: 351
Algorithm: Smith-Waterman (SSE2, Michael Farrar 2006) (7.2 Nov 2010)
Parameters: BL50 matrix (13:-5), open/ext: 0/-2
Scan time: 0.000

The best non-identical alignments are: ls-w bits E(1) %_id %_sim alen
Rosalind_37 ( 96) [f] 469 25.3 0.00023 0.796 93
Rosalind_37 ( 96) [r] 0 -22.6 1 -1.000 -1.000 0

>>>Rosalind_12, 98 nt vs lalign-I20240508-115512-0330-98093280-p1m.bsequence library

>>>Rosalind_37 (96 nt)
Waterman-Eggert score: 469; 25.3 bits; E(1) < 0.00023
78.5% identity (79.6% similar) in 93 nt overlap (1-75:4-96)

          10    20    30    40
Rosalind GACTCTTGTGCGCTTAAATAGATACATATT-----T---ACT---C-TTG---A
          ::::::::::::::::::::: : : : : :: : :
Rosalind GACTCTTGTGCGCTTAAATAGATACATATTCAACAAGTGTGCACCTAGCCTTGCAGA
          10    20    30    40    50    60
          50    60    70
Rosalind CTCTTTGTTGGCCTTAAATAGATACATATTG
          ::: ::::::::::::: :::::
Rosalind CTCCCTTGTTGCCTTAAATAGATACATATTG
          70    80    90

>>>///

98 residues in 1 query sequences
96 residues in 1 library sequences
Scomplib [36.3.8h May, 2020]
start: Wed May 8 10:55:14 2024 done: Wed May 8 10:55:14 2024
Total Scan time: 0.000 Total Display time: 0.000

```

Στο επόμενο πρόβλημα μας δίνεται ένα αρχείο FASTQ και ένα όριο 26. Πρέπει να βρούμε τον αριθμό των θέσεων των αλληλουχιών όπου η ποιότητα βάσεων πέφτει κάτω από 26.

```

labin.py
1 from Bio import SeqIO
2 from Bio.SeqRecord import SeqRecord
3 from Bio.Seq import Seq
4
5 def count_low_quality_positions(records, quality_threshold):
6     # Initialize a list to store quality scores
7     quality_scores = []
8
9     # Parse the records
10    for record in records:
11        # Append the quality scores of the current record to the list
12        quality_scores.append(record.letter_annotations["phred_quality"])
13
14    # Compute the mean quality score for each position
15    mean_quality_scores = []
16    for i in range(len(quality_scores[0])): # assuming all records have the same length
17        sum_scores = sum(record[i] for record in quality_scores)
18        mean_quality_scores.append(sum_scores / len(quality_scores))
19
20    # Count the number of positions where the mean quality score falls below the threshold
21    num_low_quality_positions = sum(i < quality_threshold for i in mean_quality_scores)
22
23    return num_low_quality_positions
24
25 # Your data
26 data = [
27     SeqRecord(Seq("GCCCGAGGGAAACCCCTCCGACCGAGGATCGT"), id="Rosalind_0029", description="",
28             letter_annotations={"phred_quality": [ord(c)-33 for c in ">?F?@6<HF?<85486B;85:8488/2/"]}),
29     SeqRecord(Seq("TGTGATGGCTCTGAATGGTTCAGGCAGT"), id="Rosalind_0029", description="",
30             letter_annotations={"phred_quality": [ord(c)-33 for c in "@J@H@>B9:B;<D==:<;,<::?463-,,,"]}),
31     SeqRecord(Seq("CACTCTACTCCCTAGCCGAACCTCTTTT"), id="Rosalind_0029", description="",
32             letter_annotations={"phred_quality": [ord(c)-33 for c in "=88;99637@5,4664-65)/?4-2+$]$"]}),
33     SeqRecord(Seq("GATTATGATATCAGTTGGCTCCGAGAGCGT"), id="Rosalind_0029", description="",
34             letter_annotations={"phred_quality": [ord(c)-33 for c in "<@BGE@8C9=B9:B<>>7?B>7:02+33."]}),
35 ]
36
37 # Test the function with your data
38 quality_threshold = 26
39 print(count_low_quality_positions(data, quality_threshold))

```

Τώρα εισάγουμε στο εργαλείο clustal τις αλληλουχίες dna και βλέπουμε οτι αυτή με τις πιο πολλές διαφορές (τα πιο πολλά κενά) είναι η πρώτη.

 <https://www.ebi.ac.uk/jdispatcher/msa/clustalo/summary?jobId=clustalo-I20240508-135316-0394-26111112-p1m>

Results for Job ID: clustalo-I20240508-135316-0394-2611111

Tool Output	Alignments	Guide Tree	Phylogenetic Tree
Tool output Download <pre> CLUSTAL O(1.2.4) multiple sequence alignment Rosalind_7 -----CACGTCTGTTCGCCTAAAACCTTGATTGCCGGCTACGCTAGTTAGTTA 49 Rosalind_28 GGGGTCAATGGCTGTTGCCTAAACCCCTGGCGGCCTAGCCGTAATGTTT---- 50 Rosalind_51 --TCCTATGTTGTTGCCTCAAACCTCTGGCGGCCTAGCCGTAAGGTAAG--- 49 Rosalind_18 ---GACATGTTGTTGCCTAAACTCTGTGGCGGCCTAGCCGTAAGGTAAG--- 48 Rosalind_23 --ACTCATGTTGTTGCCTAAACTCTGGCGGCCTAGCCGTAACTTAAG--- 49 * * ***** * * * * * * </pre>			
Alignment with colours			

- Έπειτα στο επόμενο πρόβλημα προσπαθούμε να βρούμε την μεγαλύτερη πρωτεΐνη που μπορεί να μεταφραστεί από ένα ORF από αυτά που υπάρχουν στην αλληλουχία μας. To SMS2 δεν δίνει ίδια αποτελέσματα με το rosalind.

Δοκιμάσαμε να λύσουμε το πρόβλημα με Biopython. Όμως πάλι βρήκαμε διαφορετική λύση.

```

1  from Bio.Seq import Seq
2
3
4  def find_longest_protein(dna_string):
5
6      dna_seq = Seq(dna_string)
7
8
9      longest_protein = ""
10
11
12      protein = dna_seq.translate(to_stop=True)
13
14
15      if len(protein) > len(longest_protein):
16          longest_protein = protein
17
18
19      return longest_protein
20
21
22  dna_string = "AGCCATGTAGCTA...TCAG"
23
24
25  print(find_longest_protein(dna_string))

```

Η λύσεις των Biopython και sms2 είναι ίδιες.

Έπειτα χρησιμοποιούμε το εργαλείο trimmer quality του galaxy για να εντοπίσουμε τις βάσεις που έχουν κακή ποιότητα και να ελαφρύνουμε την αποθήκευσή τους χρησιμοποιώντας λιγότερο χώρο. Το πρόγραμμα μας δυσκόλεψε λίγο σε αυτό το αρχείο. Όλα διορθώθηκαν όταν αντί να επιλέγουμε μορφή αρχείου πατήσαμε autodetection.

ΚΩΔΙΚΑΣ:

```

from Bio.Seq import Seq

my_seq
Seq("AGCTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC")

print("Adenine :" + str(my_seq.count("A")))
print("Cytosine :" + str(my_seq.count("C")))
print("Guanine :" + str(my_seq.count("G")))
print("Thymine :" + str(my_seq.count("T")))
-----


from Bio import Entrez
Entrez.email = "up1067508@upnet.gr"
handle = Entrez.esearch(db="nucleotide", term="Anthoxanthum" + "[Organism] AND (2003/7/25 : 2005/12/27 [Publication Date])")
record = Entrez.read(handle)
print("\n[GenBank gene database]:", record["Count"])
-----


from Bio import SeqIO
from Bio import Entrez
def fetch_and_convert(ids):
    fasta_records = []

    for id in ids:
        handle = Entrez.efetch(db="nucleotide", id=id, rettype="gb", retmode="text")
        record = SeqIO.read(handle, "genbank")
        handle.close()

        # format to FASTA
        fasta_record = record.format("fasta")
        fasta_records.append(fasta_record)

```

```

# return shorter
return min(fasta_records, key=len)

Entrez.email = "up1067508@upnet.gr"
ids = ["FJ817486", "JX069768", "JX469983"]
print(fetch_and_convert(ids))
-----
from Bio import Entrez
from Bio import SeqIO

Entrez.email = "up1067508@upnet.gr"
handle = Entrez.efetch(db="nucleotide", id=["FJ817486", "JX069768", "JX469983"], rettype="fasta")
records = list(SeqIO.parse(handle, "fasta")) # Get the list of SeqIO objects in FASTA format

length = [0, 0, 0]
length[0] = len(records[0].seq) # First record ID
length[1] = len(records[1].seq)
length[2] = len(records[2].seq)

last = min(length)
if last == length[0]:
    print(records[0])
elif last == length[1]:
    print(records[1])
elif last == length[2]:
    print(records[2])
-----
from Bio import Entrez
from Bio import SeqIO

from Bio import SeqIO

records = SeqIO.parse("gg.fastq", "fastq")
count = SeqIO.write(records, "ggcopy.fasta", "fasta")
print("Converted %i records" % count)
-----
from Bio import SeqIO

def count_low_quality_reads(fastq_file, quality_threshold):
    count = 0
    for record in SeqIO.parse(fastq_file, "fastq"):
        avg_quality = sum(record.letter_annotations["phred_quality"]) / len(record.letter_annotations["phred_quality"]) #letter.annotations is quality scores for every base, stored in the seqrecord
    #len -> returns the num items
        if avg_quality < quality_threshold:
            count += 1
    return count

# Test the function with your data
fastq_data = """@Rosalind_0041
GGCCGGTCTATTAGTTCTACCCGACGTGACGTACGGTCC
+
6.3536354;.151<211/0?:6/-2051)-*"40/.,+%
@Rosalind_0041

```

```

TCGTATGCGTAGCACTTGGTACAGGAAGTGAACATCCAGGAT
+
AH@FGGGJ<GB<<9:GD=D@GG9=?A@DC=;:?:>839/4856
@Rosalind_0041
ATTCGGTAATTGGCGTGAATCTGTTCTGACTGATAGAGACAA
+
@DJEJEJA?JHJ@8?F?IA3=;8@C95=;=?;D/;74792.""""

with open("fastq_data.fastq", "w") as f:
    f.write(fastq_data)

print(count_low_quality_reads("fastq_data.fastq", 28))
-----
from Bio.Seq import translate

# Given DNA string
dna_string = "ATGCCATGGCGCCCAGAACTGAGATCAATAGTACCGTATTAACGGGTGA"

# Loop over the genetic code tables
for gen_table in range(1, 26): # There are 25 known genetic code tables
    protein_string = translate(dna_string, table = gen_table, to_stop=True) # table
    -> function of translate
    if protein_string == "MAMAPRTEINSTRING":
        print("The index of the genetic code variant used for translation is "+str(gen_table))
        break
-----
from Bio.Seq import Seq

dna_strings = {
    "Rosalind_64": "ATAT",
    "Rosalind_48": "GCATA"
}

# Initialize a counter for the matches
matches = 0

# Iterate over the DNA strings
for name, dna_string in dna_strings.items():
    # Create a Seq object for the DNA string
    my_seq = Seq(dna_string)

    # Get the reverse complement of the DNA string
    reverse_complement = my_seq.reverse_complement()

    # If the DNA string matches its reverse complement
    if str(my_seq) == str(reverse_complement):
        matches += 1

print("Number of DNA strings that match their reverse complements:", matches)
-----
from Bio import SeqIO
from Bio.SeqRecord import SeqRecord
from Bio.Seq import Seq

```

```

def count_low_quality_positions(records, quality_threshold):
    # Initialize a list to store quality scores
    quality_scores = []

    # Parse the records
    for record in records:
        # Append the quality scores of the current record to the list
        quality_scores.append(record.letter_annotations["phred_quality"])

    # Compute the mean quality score for each position
    mean_quality_scores = []
    for i in range(len(quality_scores[0])):  # assuming all records have the same
length
        sum_scores = sum(record[i] for record in quality_scores)
        mean_quality_scores.append(sum_scores / len(quality_scores))

    # Count the number of positions where the mean quality score falls below the
threshold
    num_low_quality_positions = sum(i < quality_threshold for i in
mean_quality_scores)

    return num_low_quality_positions

# Your data
data = [
    SeqRecord(Seq("GCCCCAGGAAACCCCTCCGACCGAGGATCGT"), id="Rosalind_0029",
description="",
letter_annotations={"phred_quality": [ord(c)-33 for c in
">?F?@6<C<HF?<85486B;85:8488/2/"]}),
    SeqRecord(Seq("TGTGATGGCTCTGAATGGTCAGGCAGT"), id="Rosalind_0029",
description="",
letter_annotations={"phred_quality": [ord(c)-33 for c in
"@J@H@>B9:B;<D=:@;,:<::?463-,,"]}),
    SeqRecord(Seq("CACTCTACTCCCTAGCCGAACTCCTTTT"), id="Rosalind_0029",
description="",
letter_annotations={"phred_quality": [ord(c)-33 for c in
"=88;99637@5,4664-65)/?4-2+$]"}),
    SeqRecord(Seq("GATTATGATATCAGTTGGCTCCGAGAGCGT"), id="Rosalind_0029",
description="",
letter_annotations={"phred_quality": [ord(c)-33 for c in
"<@BGE@8C9=B9:B<>>7?B>7:02+33."]}),
]

# Test the function with your data
quality_threshold = 26
print(count_low_quality_positions(data, quality_threshold))

```

(ii)

Το COBALΤ χρησιμοποιεί προοδευτική πολλαπλή ευθυγράμμιση για τον συνδυασμό διαζευκτικών περιορισμών από διάφορες πηγές σε μια πολλαπλή ευθυγράμμιση. Χρησιμοποιώντας μια μέθοδο βαθμολόγησης, λαμβάνει υπόψη του πληροφορίες από διάφορες πηγές και τις ενσωματώνει σε μια ενιαία ευθυγράμμιση. Ξεκινά με την ευθυγράμμιση των πιο σχετικών ακολουθιών και στη συνέχεια προσθέτει προοδευτικά τις λιγότερο σχετικές. Οι διαζευκτικοί περιορισμοί που χρησιμοποιούνται μπορεί να προέρχονται από διάφορες πηγές, όπως η φυλογενετική ανάλυση, η δομική βιολογία ή η

βιοχημική πληροφορία. Το αποτέλεσμα είναι μια πολλαπλή ευθυγράμμιση που αντικατοπτρίζει την πληροφορία από όλες αυτές τις πηγές, παρέχοντας ένα πλαίσιο για την κατανόηση της εξέλιξης και της λειτουργίας των πρωτεΐνων.

Χρησιμοποιώντας αρχικά το RPS-BLAST, ψάχνει στο CDD για παρόμοιους τομείς πρωτεΐνων με τις εισόδους. Όταν ένας τομέας ταιριάζει σε πολλές εισόδους τότε αποκτούμε μεγάλη πληροφορία. Μέσω του CDD δημιουργούμε και προσωρινά προφίλ στις εισόδους με αποδοτικό και φθηνό τρόπο. Τέλος σε επόμενα στάδια χρησιμοποιούμε PROSITE μοτίβα για την ανάλυσή μας (καθώς είναι μικρότερα).

Το COBALT είναι ένα ισχυρό εργαλείο για την ανάλυση των ακολουθιών πρωτεΐνων και την πρόβλεψη της δομής και της λειτουργίας τους.

Γραμμένο σε c,c++,perl.

Για την βαθμολόγηση πολλαπλών πρωτεΐνων χρησιμοποιεί έναν συνδυασμό από την καταγραφή των λογαριθμικών αναλογιών (log-odds) και τη βασισμένη στην εντροπία βαθμολόγηση. Αυτή η μέθοδος βαθμολογεί μια ακολουθία βάσει της πιθανότητας εμφάνισης ενός συγκεκριμένου αμινοξέου σε σχέση με την πιθανότητα εμφάνισης αυτού.

Βήματα αλγορίθμου :

- Εύρεση ευθυγραμμίσεων για τη δημιουργία περιορισμών..- RPS-BLAST,- CDD,- PROSITE,- PHI-BLAST
- Εύρεση πρόχειρων προφίλ και συνεπών περιορισμών.
- Δημιουργία Δένδρου-οδηγού.-συνεπείς περιορισμοί,- BLOSUM62
- Δημιουργία πολλαπλής ευθυγράμμισης χρησιμοποιώντας το τρέχον σύνολο περιορισμών και το δέντρο.- Needleman-Wunsch,- (Edgar, 20004a),- (Edgar and Sjölander, 2004; Wang and Dunbrack, 2004)
- Δημιουργία διαχωρισμών και επανευθυγράμμιση.
- Εκτέλεση (προαιρετικής) εικλεπτυσμένης διαδικασίας με τον καθορισμό ενός νέου συνόλου περιορισμών και επανάληψη από το Βήμα 4 όσο το πλήθος των περιορισμών συνεχίζει να αυξάνεται.

Το COBALT είναι ένα εργαλείο με τις εξής επιλογές:

Gap Penalties Πρόστιμο κενών (Διαφορετικό για το πρώτο κενό που θα βρεθεί)

Opening

Extension

End-Gap Penalties Πρόστιμο κλείσιμου κενών (Διαφορετικό για το πρώτο κενό που θα βρεθεί)

Opening

Extension

Constraint Parameters

RPS blast Εδώ έχουμε μια επιλογή να απενεργοποιήσουμε το RPS blast. Αυτό συνελεί στο να μην χρησιμοποιηθεί αναζήτηση για συντηρημένα πεδία και άρα ο αλγόριθμος τελειώνει πιο γρήγορα. Όταν έχουμε πρωτεΐνες ίδια οικογένειας αποεπιλέγουμε το rps blast. Όταν έχουμε πρωτεΐνες διαφορετικών οικογενειών το επιλέγουμε γιατί θα μας δωθούν αποτελέσματα κακής ποιότητας.

Use RPS BLAST to guide alignment

Constraint E-value Καθορίζει την μέση αναμενόμενη τιμή σφαλμάτων στα αμινοξέα. Αν έχουμε πολύ σημαντικές ακολουθίες, βάζουμε χαμηλό δείκτη. “Μόνο οι αντιστοιχίες με Evalue χαμηλότερο από το κατώφλι θα χρησιμοποιηθούν”

Conserved columns Αυτή η επιλογή επανεξετάζει τις στήλες που είχαν επιτυχία μετά το πέρας του αλγορίθμου για τυχόν βελτιώσεις. Αν το αποεπιλέξουμε θα γίλιτώσουμε χρόνο αλλά θα χάσουμε ποιότητα.

Find Conserved Columns and Recompute Alignment

Query Clustering Parameters

Query Clustering Parameters Η επιλογή “Use query clusters” στο Cobalt είναι μια προαιρετική ρύθμιση που μπορεί να βοηθήσει στη μείωση του χρόνου υπολογισμού¹. Όταν αυτή η επιλογή είναι ενεργοποιημένη, το Cobalt χρησιμοποιεί συστάδες από παρόμοιες ακολουθίες για να μειώσει τον αριθμό των απαραίτητων συγκρίσεων.

Η ιδέα πίσω από τη χρήση συστάδων είναι ότι οι περιορισμοί δεν συμβάλλουν στην ευθυγράμμιση πολύ παρόμοιων ακολουθιών. Αυτό σημαίνει ότι, αν έχετε πολλές ακολουθίες που είναι πολύ παρόμοιες μεταξύ τους, το Cobalt μπορεί να τις θεωρήσει ως μια συστάδα και να τις επεξεργαστεί ως μια ενιαία οντότητα, αντί να τις επεξεργαστεί ξεχωριστά. Αυτό μπορεί να επιταχύνει σημαντικά τη διαδικασία ευθυγράμμισης, ειδικά για μεγάλα σύνολα δεδομένων. Ωστόσο, πρέπει να σημειωθεί ότι η χρήση συστάδων μπορεί να έχει επιπτώσεις στην ακρίβεια της τελικής ευθυγράμμισης, εξαρτάται δε από τα δεδομένα και τους στόχους της ανάλυσης.

Use query clusters

Word Size Η επιλογή “Word Size” στο Cobalt αναφέρεται στο μέγεθος των λέξεων που χρησιμοποιούνται για τη σύγκριση των ακολουθιών. Συγκεκριμένα, ορίζει τον αριθμό των συνεχόμενων χαρακτήρων (ή “λέξεων”) που πρέπει να ταιριάζουν μεταξύ δύο ακολουθιών για να θεωρηθούν ως αντιστοιχία.

Για παράδειγμα, αν ορίσετε το “Word Size” σε 3, τότε το Cobalt θα ψάχνει για ακολουθίες 3 συνεχόμενων χαρακτήρων που ταιριάζουν μεταξύ των δύο ακολουθιών. Αν βρεθούν τέτοιες αντιστοιχίες, τότε οι ακολουθίες θεωρούνται ότι έχουν κάποια ομοιότητα.

Η επιλογή αυτή είναι σημαντική για την απόδοση και την ακρίβεια της ανάλυσης. Μια μεγαλύτερη τιμή για το “Word Size” μπορεί να οδηγήσει σε πιο ακριβείς αντιστοιχίες, αλλά μπορεί επίσης να αυξήσει τον χρόνο εκτέλεσης της ανάλυσης. Αντίθετα, μια μικρότερη τιμή μπορεί να οδηγήσει σε πιο γρήγορη ανάλυση, αλλά μπορεί να μην εντοπίζει όλες τις πιθανές αντιστοιχίες. Η κατάλληλη τιμή εξαρτάται από τα δεδομένα και τους στόχους της ανάλυσης.

Max Cluster distance Η επιλογή “Max Cluster distance” στο Cobalt αναφέρεται στη μέγιστη επιτρεπόμενη απόσταση μεταξύ δύο ακολουθιών σε μια συστάδα. Αυτό το όριο αποτρέπει το Cobalt από το να δημιουργεί συστάδες από άσχετες ακολουθίες. Η απόσταση μεταξύ δύο ακολουθιών υπολογίζεται ως το ποσοστό των λέξεων που εμφανίζονται και στις δύο ακολουθίες σε σχέση με τον αριθμό όλων των λέξεων στη μεγαλύτερη ακολουθία.

Οι τιμές που παραθέτεται (0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95) είναι πιθανά ρυθμίσεις για αυτήν την επιλογή. Μια χαμηλότερη τιμή θα οδηγήσει σε πιο στενές συστάδες, ενώ μια υψηλότερη τιμή θα επιτρέψει μεγαλύτερη απόσταση μεταξύ των ακολουθιών σε μια συστάδα, πιθανώς δημιουργώντας πιο ευρείες συστάδες. Η κατάλληλη τιμή εξαρτάται από τα δεδομένα και τους στόχους της ανάλυσης.

0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95

Alphabet Αυτή η παράμετρος αναφέρεται στο αλφάβητο που χρησιμοποιείται για τη δημιουργία αναπαραστάσεων αριθμού k-mer των ακολουθιών. Οι διαθέσιμες επιλογές είναι SE-B15 και SE-V10, που είναι αλφάβητα 15 και 10 γραμμάτων. Το “Regular” αναφέρεται στο κανονικό αλφάβητο 20 αμινοξέων

SE-B15
SE-V10
Regular

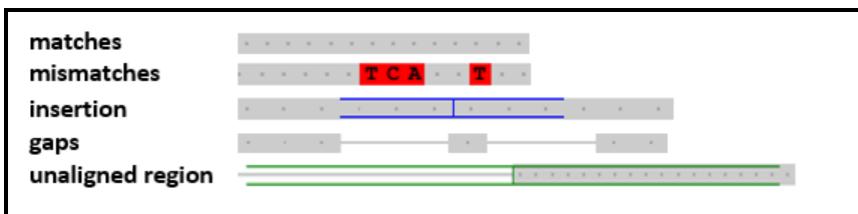
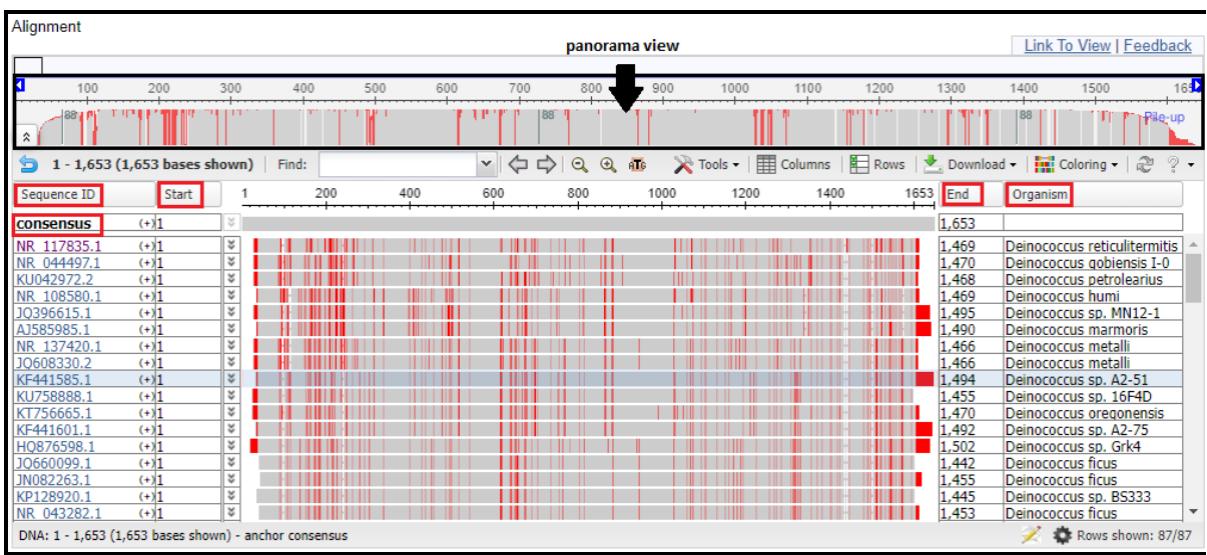
Πηγή: COBALT: constraint-based alignment tool for multiple protein sequences Jason S. Papadopoulos and Richa Agarwala https://watermark.silverchair.com/bioinformatics_23_9_1073.pdf?token=AQECAHi208BE49Ooan9k khW_Ercy7Dm3ZL_9Cf3qfKAc485ysgAAA3kwggN1BgkqhkiG9w0BBwagggNmMIIDYgIBADCC

A1sGCSqGSIb3DQEHTAeBglhgkBZQMEAS4wEQQMI65iZrBU8t8wrr_6AgEQgIIDLK GqXCH
6PYRFCFnIm5B1vQb6MzJqNBCfPAEQH2KtYmBIWiyu00YGIUU3L6-
fURopfaaCyTp_R_xTPSnNLQ4r260BKmxwVEhjUSPGa_Ze6ccnfPpfW2G0HNH5t1_bBiqHFcK_v
AdGYFcHoSGfoVPUwaawFZcs4YCoiZ_7A4LZpb062KvTanZ6L_PYGMBXcBddZjiN_uTaKlzz7uz
e9-N-lZsof72-
q8grvhjWzsIhC7fpM21JNKRT8X58SoPORpRwxk5VH2rycWS3AvvHRaqseOHxZBkIhEd9Gjz_h2eq
RxM7BDVnMFrYoB4oYgpku8lBUYLsAbmY_UWdHqWaTqHjd9aZF8m74zYGBmh50lRb2XQ6r
WpEn8pNVEKDvBqglxmdrKTtnlr2gRIQpPsLqP0jLd0klumnFKIQ8Sh21ofTMX9LIQ3ihK5ru8g1gl
vHhLDmH4Gmhd70DzN_RDhzl02fObp9OqrD70fDkVs-
nwq5xmxrWJTKDf2z2Jk7yAj7irEjthulGmvcdXrRtRUS-tgj_-
xxyIIrU1W4TbjLVY2gkGmWo51nllTG DyPxrMEbYAREbiA6AKz1rtnNdXffcLOJYcvYTQz6FQoY
NRZcvpQK_x9rdNb2J4_ba7k1B1Wer06g5os6KPv7znLghn1dqu4poiahimUQxL6LxCLmlJfGCJZBY
_NxtTJZNClxXh7x9XfQPlxZCCZkmFuM7wOh3R493cPoyPnaob2HBvHI_aKq1wsilh8gRbCd4H3b
qj_Fr_G93aU2eK0nUDQw0kfZIGmfr7b1AAZbzsBmPqlG0pdK7a7ckC4Y_Z6zpthwnCcN89UflvM
9eZPUJ0cMkiF_Ue210vAIecPK_a2WwbQzqJMnCW7EdvAEN9wFFPYB3iehMuDK82a6Vhlu0aO6
SqtK5YgMysdmqan9WpVTK-QqvxDa16f1JMT0bul03u7gEd5v6uT7RE9x-
Nuy5t2ofuc904GwYb2ZLpPLMzoyim6tGFLPJ6gzuZPvg5S7BA27p8ZBW5cgi4Lb-
Xtxe7ML9nzBBFZ4PAW0pfJByJ3u0QslPk6Rx0ZKMOQt-uDv5xWr8y

NCBI Multiple Sequence Alignment Viewer

Ένα χρήσιμο εργαλείο για την αναπαράσταση ευθυγραμμίσεων με γραφικά, οι οποίες έχουν γίνει με εργαλεία όπως το Cobalt.

Χρησιμοποιεί χρώματα, (τα οποια μπορούν να οριστούν από το χρήστη) για να δείχνει τις διαφορές στις ευθυγραμμίσεις. προεπιλογή είναι τα κενά με γκρι, οι αστοχίες με κόκκινο και οι εισαγωγές με μπλε. Αν τα αρχεία μας είναι πολύ μεγάλα μπορούμε να κάνουμε ζουμ σε διάφορες περιοχές και να δούμε πιο αναλυτικά αποτελέσματα. Υπάρχουν πολλές επιλογές χρωματισμού. Μια από αυτές είναι να χρωματίσουμε τα αμινοξέα μέσω της υδροπάθειάς τους, πράγμα που θα ήταν χρήσιμο στη μελέτη που έγινε για την δημοσίευση με τις υδροπάθειες των μορίων πο χρησιμοποιήθηκαν για να εξετάσουν την δομή ενός μορίου στη 2ρη εργασία βιοπληροφορικής.



Αν στοχεύουμε σε συγκεκριμένη ακολουθία μπορούμε να την θέσουμε ως "άγκυρα" και να δούμε όλες τις άλλες σε σχέση προς αυτήν.

Βλέπουμε ότι οι στήλες προσδιορισμού δεξιά και αριστερά έχουν τα ID και το όνομα του οργανισμού. Αυτές αλλάζουν για την διευκόλυνσή μας με ότι πεδία θέλουμε, πχ με την χώρα που προήλθε κάτι (πολύ χρήσιμο για ιούς). Αφού γίνει αυτό υπάρχουν και επιλογές ταξινόμησης για ότι μας ενδιαφέρει, πχ best alignments at the top. Συνήθεις επιλογές είναι **Sequence ID**, **Organism**, **Gene**, **Date**, **Country**, **Host**, **Source**, (%) **Identity,(%) Coverage**, **Mismatches** relative to an anchor

Επιπλέον μπορούμε να ζητάμε αναλυτική περιγραφή σε διάφορα σημεία που θέλουμε παραπάνω πληροφορίες. Γίνεται να συνδεθούμε και κατευθείαν με γνωστές βάσεις οργανισμών όπως GenBank και να βρούμε το αρχείο μας με ένα κλικ. Αν θέλουμε να εντοπίσουμε μια συγκεκριμένη ακολουθία αμινοξέων απλά την πληκτρολογούμε.

- Clustal Omega: Είναι ένα νέο πρόγραμμα πολλαπλής ευθυγράμμισης ακολουθιών που χρησιμοποιεί οδηγούς δέντρων και τεχνικές HMM profile-profile για τη δημιουργία ευθυγραμμίσεων μεταξύ τριών ή περισσότερων ακολουθιών. Είναι κατάλληλο για μεσαίες έως μεγάλες ευθυγραμμίσεις. Επιλογές: OUTPUT FORMAT, DEALIGN INPUT, MBED-LIKE CLUSTERING, MBED-LIKE CLUSTERING, COMBINED ITERATIONS ,MAX GUIDE TREE ,MAX HMM ITERATIONS ,ORDER ,DISTANCE MATRIX ,OUTPUT GUIDE TREE

- EMBOSS Cons: Το EMBOSS Cons δημιουργεί μια ακολουθία συναίνεσης από μια πολλαπλή ευθυγράμμιση πρωτεϊνών ή νουκλεοτιδίων. Επιλογές: PLURALITY ,SETCASE ,IDENTITY ,NAME ,MATRIX
 - Kalign: Είναι ένα πολύ γρήγορο εργαλείο MSA που επικεντρώνεται σε τοπικές περιοχές. Είναι κατάλληλο για μεγάλες ευθυγραμμίσεις. Επιλογές: OUTPUT FORMAT ,MACSIM,GAP OPEN PENALTY ,GAP EXTENSION PENALTY ,TERMINAL GAP PENALTIES
 - MAFFT: Είναι ένα εργαλείο MSA που χρησιμοποιεί γρήγορους μετασχηματισμούς Fourier. Είναι κατάλληλο για μεσαίες έως μεγάλες ευθυγραμμίσεις. Επιλογές: OUTPUT FORMAT ,MATRIX (PROTEIN ONLY) ,GAP OPEN PENALTY ,GAP EXTENSION PENALTY ,ORDER ,TREE REBUILDING NUMBER ,GUIDE TREE OUTPUT ,MAX ITERATE ,PERFORM FFTS
 - MUSCLE: Είναι ένα ακριβές εργαλείο MSA, ιδιαίτερα καλό με πρωτεΐνες. Είναι κατάλληλο για μεσαίες ευθυγραμμίσεις. Επιλογές: OUTPUT FORMAT ,OUTPUT TREE
 - MView: Μετατρέπει ένα αποτέλεσμα αναζήτησης ομοιότητας ακολουθίας σε μια πολλαπλή ευθυγράμμιση ακολουθιών ή μεταμορφώνει μια πολλαπλή ευθυγράμμιση ακολουθιών χρησιμοποιώντας το πρόγραμμα MView. Επιλογές: INPUT FORMAT ,OUTPUT FORMAT ,HTML MARKUP ,CSS ,PCID ,ALIGNMENT ,RULER ,ALIGNMENT WIDTH ,COLORING ,COLOR MAP ,CONSENSUS ,CONCOLOURING ,GROUPMAP ,CONCOLORMAP ,CONGROUP MAP ,CONGAPS
 - T-Coffee: Είναι ένα εργαλείο MSA βασισμένο στη συνέπεια που προσπαθεί να αντιμετωπίσει τις παγίδες των προοδευτικών μεθόδων ευθυγράμμισης. Είναι κατάλληλο για μικρές ευθυγραμμίσεις. Επιλογές: OUTPUT FORMAT ,MATRIX , ORDER
 - WebPRANK: Το EBI έχει ένα νέο πρόγραμμα πολλαπλής ευθυγράμμισης ακολουθιών που είναι ευαισθητοποιημένο στη φυλογένεση και χρησιμοποιεί πληροφορίες εξέλιξης για να βοηθήσει στην τοποθέτηση εισαγωγών και διαγραφών -- Αυτό το εργαλείο ξεχωρίζει από τα παραπάνω καθώς έχει ενσωματωμένα μοντέλα δομής των ακολουθιών, που μπορούν να βοηθήσουν στην κατανόηση της λειτουργίας των πρωτεϊνών και στην ακριβέστερη ευθυγράμμιση τους.
1. **OUTPUT FORMAT:** Καθορίζει τη μορφή των αποτελεσμάτων που επιστρέφονται από το πρόγραμμα.
 2. **DEALIGN INPUT:** Αφαιρεί τις αντιστοιχίσεις από τις εισαγόμενες ακολουθίες.
 3. **MBED-LIKE CLUSTERING:** Χρησιμοποιεί μια τεχνική ομαδοποίησης για να βελτιώσει την απόδοση της διαδικασίας ευθυγράμμισης.
 4. **COMBINED ITERATIONS:** Συνδυάζει πολλαπλές επαναλήψεις για να βελτιώσει την ακρίβεια της ευθυγράμμισης.
 5. **MAX GUIDE TREE:** Καθορίζει τον μέγιστο αριθμό των δέντρων οδηγών που θα χρησιμοποιηθούν.

6. **MAX HMM ITERATIONS:** Καθορίζει τον μέγιστο αριθμό των επαναλήψεων HMM.
7. **ORDER:** Καθορίζει τη σειρά των ακολουθιών στην ευθυγράμμιση.
8. **DISTANCE MATRIX:** Υπολογίζει μια μήτρα αποστάσεων μεταξύ των ακολουθιών.
9. **OUTPUT GUIDE TREE:** Επιστρέφει το δέντρο οδηγό που χρησιμοποιήθηκε για την ευθυγράμμιση.
10. **PLURALITY:** Καθορίζει τον αριθμό των ακολουθιών που πρέπει να συμφωνούν για να σχηματίσουν μια συναίνεση.
11. **SETCASE:** Καθορίζει την περίπτωση των ακολουθιών εξόδου.
12. **IDENTITY:** Υπολογίζει την ταυτότητα μεταξύ των ακολουθιών.
13. **NAME:** Καθορίζει το όνομα της ευθυγράμμισης.
14. **MATRIX:** Καθορίζει τη μήτρα αντικατάστασης που θα χρησιμοποιηθεί.
15. **MACSIM:** Επιτρέπει την ανάλυση των αποτελεσμάτων με το εργαλείο MACSIM.
16. **GAP OPEN PENALTY:** Καθορίζει την ποινή για το άνοιγμα ενός κενού.
17. **GAP EXTENSION PENALTY:** Καθορίζει την ποινή για την επέκταση ενός κενού.
18. **TERMINAL GAP PENALTIES:** Καθορίζει τις ποινές για τα κενά στα άκρα των ακολουθιών.
19. **MATRIX (PROTEIN ONLY):** Καθορίζει τη μήτρα αντικατάστασης πρωτεΐνων που θα χρησιμοποιηθεί.
20. **TREE REBUILDING NUMBER:** Καθορίζει τον αριθμό των επαναδομήσεων δέντρου που θα πραγματοποιηθούν.
21. **GUIDE TREE OUTPUT:** Επιστρέφει το δέντρο οδηγό που χρησιμοποιήθηκε για την ευθυγράμμιση.
22. **MAX ITERATE:** Καθορίζει τον μέγιστο αριθμό επαναλήψεων που θα πραγματοποιηθούν.
23. **PERFORM FFTS:** Καθορίζει αν θα πραγματοποιηθούν γρήγοροι μετασχηματισμοί Fourier.
24. **OUTPUT TREE:** Επιστρέφει το δέντρο που παράγεται από την ευθυγράμμιση.
25. **INPUT FORMAT:** Καθορίζει τη μορφή των εισαγόμενων ακολουθιών.
26. **HTML MARKUP:** Καθορίζει αν θα χρησιμοποιηθεί HTML για τη διαμόρφωση των αποτελεσμάτων.
27. **CSS:** Καθορίζει το CSS που θα χρησιμοποιηθεί για τη διαμόρφωση των αποτελεσμάτων.
28. **PCID:** Υπολογίζει την ταυτότητα των ακολουθιών.
29. **ALIGNMENT:** Καθορίζει τη μορφή της ευθυγράμμισης.
30. **RULER:** Καθορίζει αν θα εμφανιστεί ένας χάρακας για την ευθυγράμμιση.
31. **ALIGNMENT WIDTH:** Καθορίζει το πλάτος της ευθυγράμμισης.
32. **COLOR MAP:** Καθορίζει τον χάρτη χρωμάτων που θα χρησιμοποιηθεί για την ευθυγράμμιση.
33. **CONSENSUS:** Υπολογίζει μια ακολουθία συναίνεσης από την ευθυγράμμιση.
34. **CONCOLOURING:** Καθορίζει τον τρόπο χρωματισμού της ακολουθίας συναίνεσης.
35. **GROUPMAP:** Καθορίζει τον χάρτη ομάδας που θα χρησιμοποιηθεί για την ευθυγράμμιση.
36. **CONCOLORMAP:** Καθορίζει τον χάρτη χρωμάτων που θα χρησιμοποιηθεί για την ακολουθία συναίνεσης.
37. **CONGROUPMAP:** Καθορίζει τον χάρτη ομάδας που θα χρησιμοποιηθεί για την ακολουθία συναίνεσης.
38. **CONGAPS:** Καθορίζει αν θα εμφανιστούν κενά στην ακολουθία συναίνεσης.
39. **COLORING:** Καθορίζει τα χρώματα.

Ερώτημα 2

Παρακάτω πραγματοποιούμε αυτή την αναζήτηση NP_000137 (διαλέγοντας BLASTP protein-to-protein) και διαλέγουμε να ταξινομήσουμε τις πρωτεΐνες από τις πιο όμοιες στις λιγότερο. Έπειτα επιλέγουμε τις 3 πρώτες.

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%	1e-125	99.43%	227	6WX6_A
ferritin-like domain-containing protein [Pseudomonas aeruginosa]	Pseudomonas aeruginosa	361	361	100%	8e-125	100.00%	237	WP_217683847.1
hypothetical protein [Homo sapiens]	Homo sapiens	361	361	100%	1e-124	100.00%	241	CAE11873.1

```
>6WX6_A:33-207 Chain A, Ferritin light chain [Homo sapiens]
MSSQIRQNYSTDVEAAVNSLVNLYLQASYYTSLGFYFDRDDVALEGVSHPHRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMG
>WP_217683847.1:63-237 ferritin-like domain-containing protein, partial [Pseudomonas aeruginosa]
MSSQIRQNYSTDVEAAVNSLVNLYLQASYYTSLGFYFDRDDVALEGVSHPHRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMG
>CAE11873.1:67-241 hypothetical protein, partial [Homo sapiens]
MSSQIRQNYSTDVEAAVNSLVNLYLQASYYTSLGFYFDRDDVALEGVSHPHRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMG
```

To t-coffee εδώ και κάποιες μέρες δεν είναι διαθέσιμο στην επίσημη σελίδα του. Παρόλα αυτά από το documentation του στην σελίδα www.ebi.ac.uk, είναι αρκετά καλό εργαλείο για μικρές ευθυγραμμίσεις, ώρα θα το χρησιμοποιήσουμε από την σελίδα του EBI.

C https://www.ebi.ac.uk/jdispatcher/msa/tcoffee

Protein DNA RNA

Paste your sequence here - or use the example sequence

```
>6WX6_A:33-207 Chain A, Ferritin light chain [Homo sapiens]
MSSQIRQNYSTDVEAVNSLVNLYLQASYTYLSLGFYFDRDDVALEGVSHFFRELAEKREGYERLLKMQNQRGGRALFQ
DIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDHALGSARTDPHLCDFLETHFLDEEVKLICKMGDHLTNLHRLGGPEA
GLGEYLFERLTLRH
```

Επιλογή αρχείου Δεν επλέχθηκε κανένα αρχείο.

Use the example Clear sequence More example inputs

Parameters

OUTPUT FORMAT ⓘ

ClustalW

More options ▾

Submit

Title

T-Coffee's job

Submit

Αποτελέσματα

Tool Output

Alignments

Guide Tree

Phylogenetic Tree

Tool output

CLUSTAL W (1.83) multiple sequence alignment

Download

```
6WX6_A_33-207      MSSQIRQNYSTDVEAVNSLVNLYLQASYTYLSLGFYFDRDDVALEGVSH
CAE11873.1_67-241  MSSQIRQNYSTDVEAVNSLVNLYLQASYTYLSLGFYFDRDDVALEGVSH
WP_217683847.1_63-237 MSSQIRQNYSTDVEAVNSLVNLYLQASYTYLSLGFYFDRDDVALEGVSH
*****
FFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKTPDAMKA
CAE11873.1_67-241  FFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKTPDAMKA
WP_217683847.1_63-237 FFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKTPDAMKA
*****
MALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMGDH
6WX6_A_33-207      NLHRLGGPEAGLGEYLFERLTLRH
CAE11873.1_67-241  NLHRLGGPEAGLGEYLFERLTLKH
WP_217683847.1_63-237 NLHRLGGPEAGLGEYLFERLTLKH
*****

```

Alignments

Guide Tree

Phylogenetic Tree

Results Viewers

Result Files

Submission Details

COLOR SCHEME

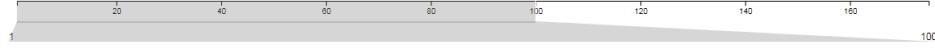
clustal2

LEGEND

A R N D C Q E G H I L K M F P S T W Y V B X Z

3 sequences

6WX6_A_33-207
CAE11873.1_67-241
WP_217683847.1_63-237



Α Μη ασφαλής | http://tcoffee.crg.cat/tcs

Service Unavailable

The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.

O server τελικά δούλεψε:

← → ⌂ tcoffee.crg.eu/apps/tcoffee/do:regular

Gmail YouTube Maps

T COFFEE

Home History Tutorial References Contacts Projects Download

T-Coffee

Aligns DNA, RNA or Proteins using the default T-Coffee

Sequences input
Paste or upload your set of sequences in FASTA format

Sequences to align
[Click here to use the sample file](#)

```
>6WX6_A Chain A, Ferritin light chain [Homo sapiens]
TGWSHPQFEKLKGSSRGGGSGGSGGSGMSQIRQNYSTDVEAVNSLVNLQASYTYLSGFYFD
RDKVALEGV
SHFRRELAEKREGYERLLKMNQRGGRALFQDIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGS
ARTDPHLCD
FLETHFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLTLRHDGGSGGSGGSGGASGGS
>WP_217683847.1 ferritin-like domain-containing protein, partial
[Pseudomonas aeruginosa]
```

- OR - [Click here to upload a file](#)

Show more options

Your email address

Submit Reset

T-Coffee Server is hosted by the Centre for Genomic Regulation (CRG) of Barcelona - Powered by **nextflow**

Back to top

← → ⌂ tcoffee.crg.eu/apps/tcoffee/result?rid=9a522dd7

Gmail YouTube Maps

T COFFEE

Home History Tutorial References Contacts Projects Download

Processing your job request

Your request ID is 9a522dd7

Wait please ...

Do not reload this page. If you have provided your email you can close it, you will be notified by e-mail when your job is completed.

If you have not provided an e-mail, you can close this page and come back later to check the completion of your job. Your server history is kept in a cookie on your browser and you can access it at any moment using the [History](#) link on the main menu to check request status or retrieve a previous results.

CLOSE X

T-Coffee Server is hosted by the Centre for Genomic Regulation (CRG) of Barcelona - Powered by **nextflow**

Back to top

← → ⌂ tcoffee.crg.eu/apps/tcoffee/result?rid=9a522dd7

Gmail YouTube Maps

T COFFEE

- [Home](#)
- [History](#)
- [Tutorial](#)
- [References](#)
- [Contacts](#)
- [Projects](#)
- [Download](#)

T-Coffee alignment result

MSA
The multiple sequence alignment result as produced by T-coffee.

T-COFFEE, Version_11.00 (Version_11.00)
Cedric Notredame
SCORE=978

	BAD	AVG	GOOD
6WX6_A	:	96	
WP_217683847.1	:	95	
CAE11873.1	:	95	
cons	:	97	
6WX6_A	-----TGWSHP0FEKLKGGS SRGGGGGG -----SGGGGG		
WP_217683847.1	PAGLSSLASTVFGNRNSGDSL PASDRPPISSPLATS -GTIFSAISCFWDLPAFLWLAPSCOP		
CAE11873.1	QFGGPAGLSSLASTVFGNRNSGDSL PASDRPPISSPLATS -GTIFSAISCFWDLPAFLWLAPSCOP		
cons			
6WX6_A	SMSS0IRONYSTDVEAAVNSLVNLY0ASYTYSLSLFYFDRDDVALEGVS HSHF RELAEEKREGYER		
WP_217683847.1	TMSS0IRONYSTDVEAAVNSLVNLY0ASYTYSLSLFYFDRDDVALEGVS HSHF RELAEEKREGYER		
CAE11873.1	TMSS0IRONYSTDVEAAVNSLVNLY0ASYTYSLSLFYFDRDDVALEGVS HSHF RELAEEKREGYER		
cons	*****		
6WX6_A	LLKMONQRGRALFODIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLE		
WP_217683847.1	LLKMONQRGRALFODIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLE		
CAE11873.1	LLKMQNQRGRALFODIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLE		
cons	*****		
6WX6_A	THFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLT LKH -		
WP_217683847.1	THFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLT LKH -		
CAE11873.1	THFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLT LKH -		
cons	*****		

← → ⌂ tcoffee.crg.eu/apps/tcoffee/result?rid=9a522dd7

Gmail YouTube Maps

T-Coffee alignment result

MSA
The multiple sequence alignment result as produced by T-coffee.

T-COFFEE, Version_11.00 (Version_11.00)
Cedric Notredame
SCORE=978

	BAD	AVG	GOOD
6WX6_A	:	96	
WP_217683847.1	:	95	
CAE11873.1	:	95	
cons	:	97	
6WX6_A	-----TGWSHP0FEKLKGGS SRGGGGGG -----SGGGGG		
WP_217683847.1	PAGLSSLASTVFGNRNSGDSL PASDRPPISSPLATS -GTIFSAISCFWDLPAFLWLAPSCOP		
CAE11873.1	QFGGPAGLSSLASTVFGNRNSGDSL PASDRPPISSPLATS -GTIFSAISCFWDLPAFLWLAPSCOP		
cons			
6WX6_A	SMSS0IRONYSTDVEAAVNSLVNLY0ASYTYSLSLFYFDRDDVALEGVS HSHF RELAEEKREGYER		
WP_217683847.1	TMSS0IRONYSTDVEAAVNSLVNLY0ASYTYSLSLFYFDRDDVALEGVS HSHF RELAEEKREGYER		
CAE11873.1	TMSS0IRONYSTDVEAAVNSLVNLY0ASYTYSLSLFYFDRDDVALEGVS HSHF RELAEEKREGYER		
cons	*****		
6WX6_A	LLKMONQRGRALFODIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLE		
WP_217683847.1	LLKMONQRGRALFODIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLE		
CAE11873.1	LLKMQNQRGRALFODIKKPAEDEWGKTPDAMKAAMALEKKLNQALLDLHALGSARTDPHLCDFLE		
cons	*****		
6WX6_A	THFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLT LKH -		
WP_217683847.1	THFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLT LKH -		
CAE11873.1	THFLDEEVKLICKMGDHLTNLHRLGGPEAGLGEYLFERLT LKH -		
cons	*****		

← → ⌂ tcoffee.crg.eu/apps/tcoffee/result?rid=9a522dd7

Gmail YouTube Maps

Request ID: 9a522dd7
Created at: 28 May 2024, 11:05 (CEST)
Elapsed time: 16 sec
Expiration at: 07 Jun

Update

Command Line

This is the command line used to execute your alignment. You can use it as reference to run this alignment on your desktop.

```
t_coffee -in=data_223b1f6f.in -mode=regular -output=score_html clustalw_aln fasta_aln score_ascii phylip -maxseq=150 - maxlen=10000 -case=upper -seqnos=off -outorder=input -run_name=result -multi_core=4 -quiet=stdout
```

Replay

Change some input parameters and resubmit this alignment [clicking here](#).

Are you a T-Coffee guru? You may want to use the full featured T-Coffee [command line options](#).

Feedback

Give us feedback about T-coffee web server

Are you satisfied with this result? Do you like this alignment server? If so recommend it using Google+1 or Facebook.

For suggestions, questions or any problem send an email to tcoffee@googlegroups.com

T-Coffee Server is hosted by the [Centre for Genomic Regulation](#) (CRG) of Barcelona - Powered by [nextflow](#)

[Back to top](#)

Send results

Forward this result to other online tools:

Core/TCS

Evaluates your Alignment indicating the local reliability

ProtoGene

Turning amino acid alignments into bona fide CDS nucleotide alignments

MSA hub

MyHits: a new interactive resource for protein annotation and domain identification

JalView

Open this alignment in the [Jalview](#) viewer

ESPrift

ESPrift server renders sequence similarities and secondary structure information from aligned sequences

Evaluates your Alignment and outputs a Colored version indicating the local reliability.

Alignment input
Paste or upload your Multiple Sequence Alignment in CLUSTAL, FASTA, MSF, NEXUS or PHYLIP format.

MSA to evaluate
[Click here to use the sample file](#)

CLUSTAL W (1.83) multiple sequence alignment

```
6WX6_A -----TGWSHPQFEKLKGSSRGGGGGSGG-----SGGSGGSMSSQIRQNYSTDV
WP_217683847.1 ---PAGLSLASTVFGNRSGDSLPAASDRPPISSPLATS-GTIFSAISCFWDLPAAPFLWLAPSCOPTMSSQIRQNYSTDV
CAE11873.1 QFGGPAGLSLASTVFGNRSGDSLPAASDRPPISSPLATS-GTIFSAISCFWDLPAAPFLWLAPSCOPTMSSQIRQNYSTDV
6WX6_A EAAVNLVNLQASYTYLQLGFYFDRDDVALEGVSFFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKT
- OR - Click here to upload a file
```

Note: TCS requires a multiple sequence alignment. If your sequences are *not* aligned, click the the following link to align them with [T-Coffee](#). In the result page click the "Core/TCS" button to return to this page.

Show more options

Output options
Use these options to control the output formats of your output alignment

Filtered clustalw_align fasta_align phylip

Weighted score_ascii tcs_weighted tcs_replicate

Graphic score_html score_pdf

TCS evaluation result

Citation

If you found these results useful please cite the following [publications](#).

MSA
The multiple sequence alignment colored according to TCS scheme

T-COFFEE, Version_11.00 (Version_11.00)
Cedric Notredame
SCORE=978

* BAD AVG GOOD

	6WX6_A	WP_217683847.1	CAE11873.1	cons
6WX6_A	: 96			
WP_217683847.1	: 95			
CAE11873.1	: 95			
cons	: 97			

6WX6_A -----TGWSHPQFEKLKGSSRGGGGGSGG-----SGGSGGSMSSQIRQNYSTDV
WP_217683847.1 ---PAGLSLASTVFGNRSGDSLPAASDRPPISSPLATS-GTIFSAISCFWDLPAAPFLWLAPSCOPTMSSQIRQNYSTDV
CAE11873.1 QFGGPAGLSLASTVFGNRSGDSLPAASDRPPISSPLATS-GTIFSAISCFWDLPAAPFLWLAPSCOPTMSSQIRQNYSTDV
cons

6WX6_A NSLVNLQASYTYLQLGFYFDRDDVALEGVSFFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKT
WP_217683847.1 NSLVNLQASYTYLQLGFYFDRDDVALEGVSFFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKT
CAE11873.1 NSLVNLQASYTYLQLGFYFDRDDVALEGVSFFRELAEKREGYERLLKMQNQRGGRALFQDIKKPAEDEWGKT
cons

6WX6_A ALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMGDHTNLHRLGGPEAGLGEYLFERLT
WP_217683847.1 ALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMGDHTNLHRLGGPEAGLGEYLFERLT
CAE11873.1 ALEKKLNQALLDLHALGSARTDPHLCDFLETHFLDEEVKLICKMGDHTNLHRLGGPEAGLGEYLFERLT
cons

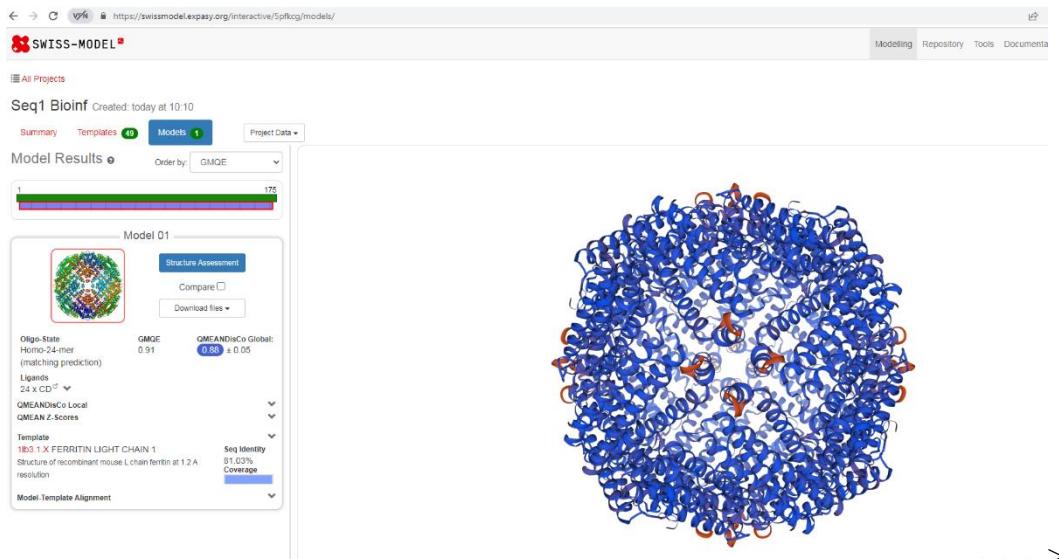
6WX6_A GSAGGASGGS
WP_217683847.1 -----
CAE11873.1 -----
cons

Τα σκούρα ροζ κομμάτια είναι πολύ αξιόπιστα, ενώ τα μπλε και πράσινα κομμάτια είναι αναξιόπιστα.
Τα κίτρινα είναι μέτριας αξιοπιστίας.

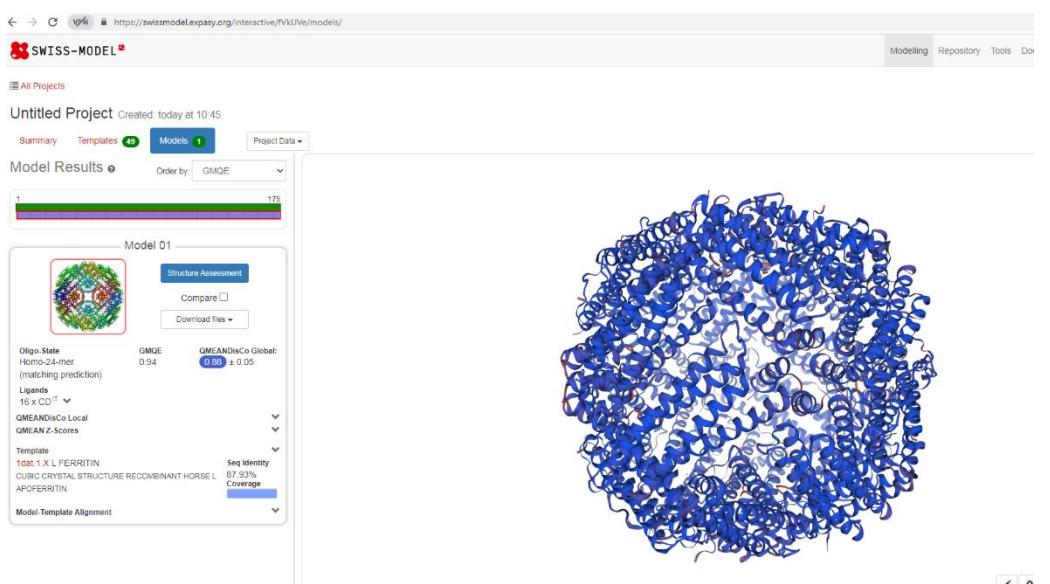
Ερώτημα (III)

<https://swissmodel.expasy.org>

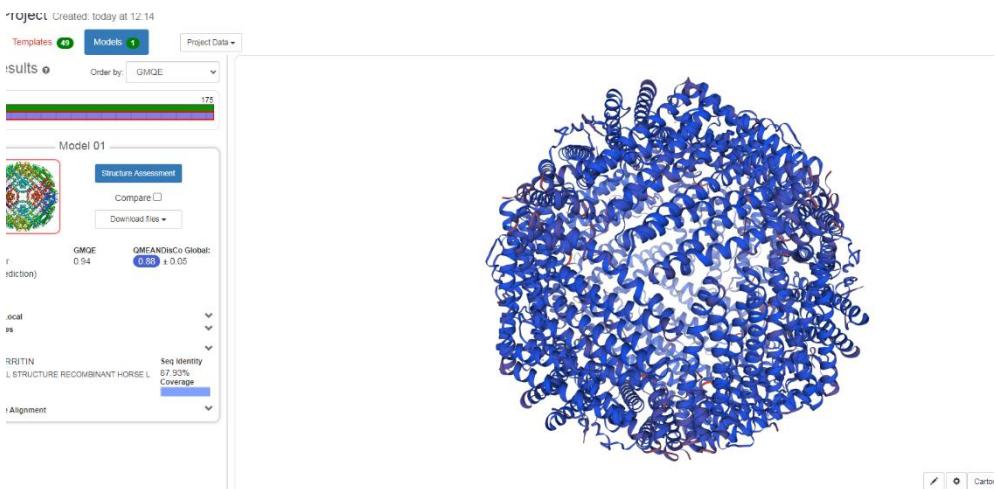
>6WX6_A:33-207 Chain A, Ferritin light chain [Homo sapiens]



847.1:63-237 ferritin-like domain-containing protein, partial [Pseudomonas aeruginosa]



>CAE11873.1:67-241 hypothetical protein, partial [Homo sapiens]



Παρατηρήσαμε ότι η σελίδα χρειάζεται αρκετή ώρα για να ανταποκριθεί στους σχεδιασμούς (σχεδόν 2 ώρες για το κάθε ένα).

<http://ekhidna.biocenter.helsinki.fi/dali/>

PROTEIN STRUCTURE COMPARISON SERVER

About PDB search PDB25 AF-DB search Pairwise All against all Tutorials References Statistics Download

All against all structure comparison

STEP 1 - Enter your input protein structures

Use the +/- buttons to create input fields. Structures may be specified by concatenating the PDB identifier (4 characters) and a chain identifier (1 character). Files should be entered before PDB identifiers. The maximum number of input structures is 64. If your input set consists only of structures in the PDB, you can upload files.

+ -

PDB identifier + chain identifier	OR upload file	Επιλογή αρχείων	model_01.pdb	>6WX6_A:33-207 Chain A	-
PDB identifier + chain identifier	OR upload file	Επιλογή αρχείων	model_01 (1).pdb	>WP_217683847.1:63-23	-
PDB identifier + chain identifier	OR upload file	Επιλογή αρχείων	model_01 (2).pdb	[in, partial [Homo sapiens]]	-

STEP 2 - Optional data

You may leave an e-mail address for notification when the job has finished. The job title is used as subject heading in the e-mail.

Job title
E-mail

Παρακάτω βλέπουμε την ακολουθία που δώσαμε χωρισμένη σε επιμέρους κομματάκια διαφορετικών πρωτεΐνων.

6WX6_A:33-207

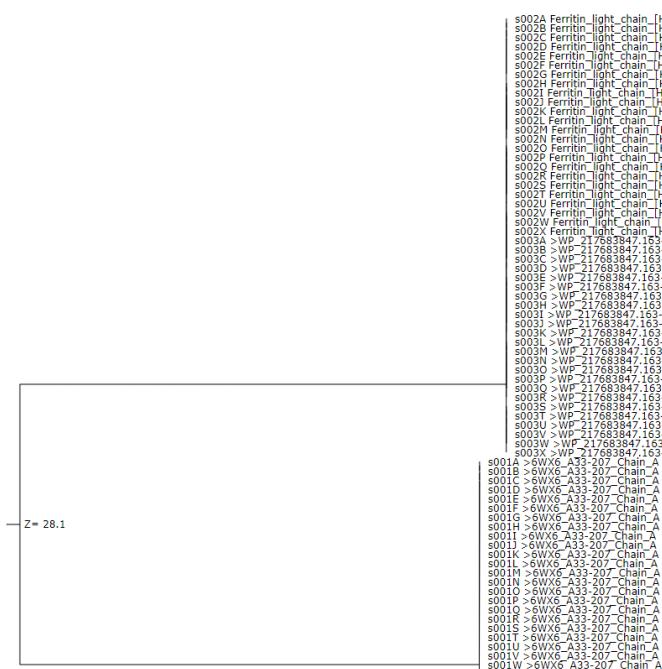
WP_217683847.1:63-237

CAE11873.1:67-241

Σύνολο 522 κομματάκια πρωτεΐνών.



Structural similarity dendrogram. Labels are linked to structural summaries. The dendrogram is derived by average linkage clustering of the structural similarity matrix (Dali Z-scores).



Z=28.1

Η ομοιότητα μετράται με βάση τα Z-scores του Dali. Οι ‘σημαντικές ομοιότητες’ έχουν Z-score πάνω από 2 και συνήθως αντιστοιχούν σε παρόμοια folds. Οι ‘δυνατές αντιστοιχίες’ έχουν ταυτότητα ακολουθίας πάνω από 20% ή ένα Z-score πάνω από 1 στο όριο που εξαρτάται από το μέγεθος της πρωτεΐνης ερωτήματος. Το όριο Z-score ορίστηκε εμπειρικά σε n/10 - 4, όπου n είναι το πλήθος των αμινοξέων στη δομή της πρωτεΐνης ερωτήματος. Επιπλέον, απαιτούμε ότι η πλήρης δομή καλύπτεται από δομικές αντιστοιχίες· ένα τμήμα της δομής της πρωτεΐνης ερωτήματος μεγαλύτερο από 80 αμινοξέα χωρίς καμία δομική αντιστοίχιση αποκλείει πάντα μια δυνατή αντιστοιχίση.

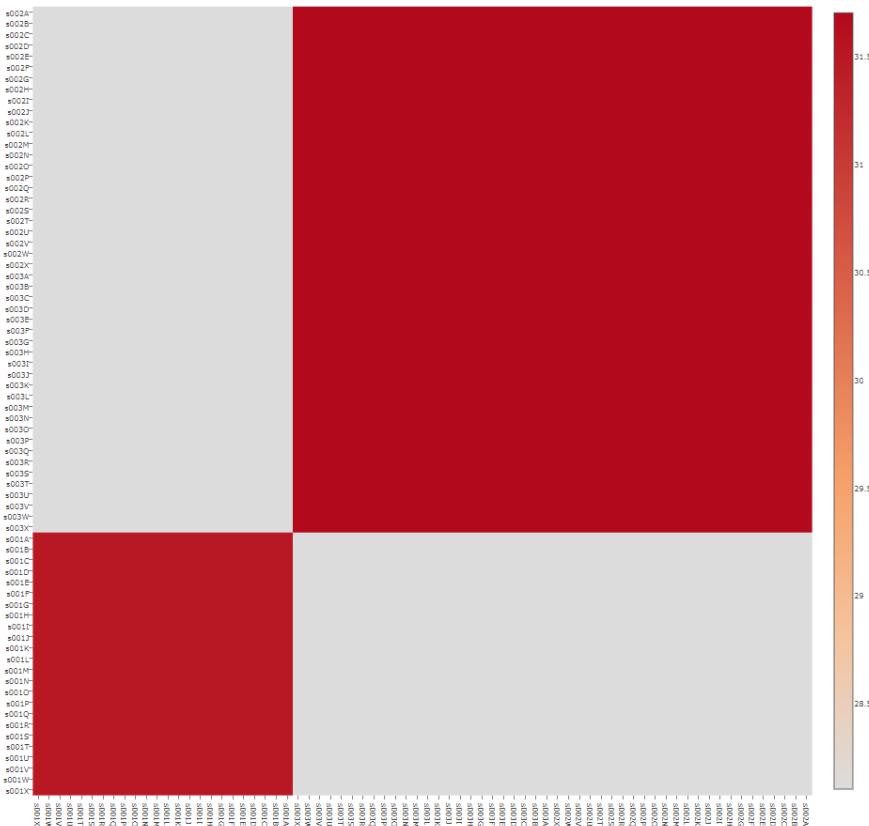
Το Z =28.1 δείχνει μεγάλη ομοιότητα όπως και περιμέναμε.

Πηγή : Searching protein structure databases with DaliLite v.3 L. Holm1,2,* , S. Kääriäinen2, P. Rosenström2 and A. Schenkel2

Όπως βλέπουμε το ιστόγραμμα μπορούμε να καταλάβουμε κάποια επιπλέον πράγματα. Η ferritin-like domain-containing protein, partial [Pseudomonas aeruginosa] και η hypothetical protein, partial [Homo sapiens] είναι λίγο πιο κοντά σχηματικά μεταξύ τους παρά με την 6WX6_A:33-207 Chain A, Ferritin light chain [Homo sapiens]. Αυτή την παρατήρηση τυχαίναι να μπορούμε να την αντιληφθούμε και από τις εικόνες των 3D δομών παραπάνω. Αυτή είναι μια πληροφορία που η σειρά αμινοξέων των πρωτεϊνών δεν μας τη δίνει (χωρίς να ξέρουμε τα δομικά στοιχεία).

Πηγή: <https://www.displayr.com/what-is-dendrogram/>

Παρακάτω βλέπουμε ένα διαφορετικό διάγραμμα ομοιότητας για τις επιμέρους πρωτεΐνες ξεχωριστά. Το έντονο κόκκινο υποδεικνύει μεγάλη ομοιότητα.



Τέλος αν πατήσουμε πάνω σε κάποιο επιμέρους κομματάκι πρωτεΐνης βλέπουμε αναλυτικά κάποια πράγματα για αυτήν, όπως και το επιμέρους σχήμα της.

Recommended browsers: Chrome, Firefox. Internet Explorer does not support all WebGL features used by PV.

Hints: Use full screen window to see options and viewer side by side. Rotate with left mouse button. Zoom in/out with middle mouse button. Click left mouse button to label atoms. Double-click left mouse button to center image.

Your query: **s002W**

C-alpha trace Cartoon Spin

Query side chains > **6.3 bits**

Display the side chains of the query where sequence conservation is above [] bits (values 0 - 6.3)
> 6.3" limit hides and "> 0" limit shows all the query side chains



Query color: Monochrome Rainbow
 Sequence Conservation Structure Conservation

Show/hide structures:

All Query

Show/hide ligands of:

All Query

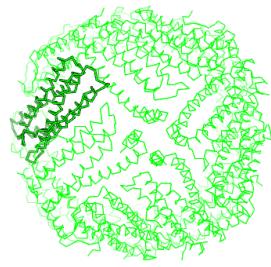
Show/hide side chains of:

All Query

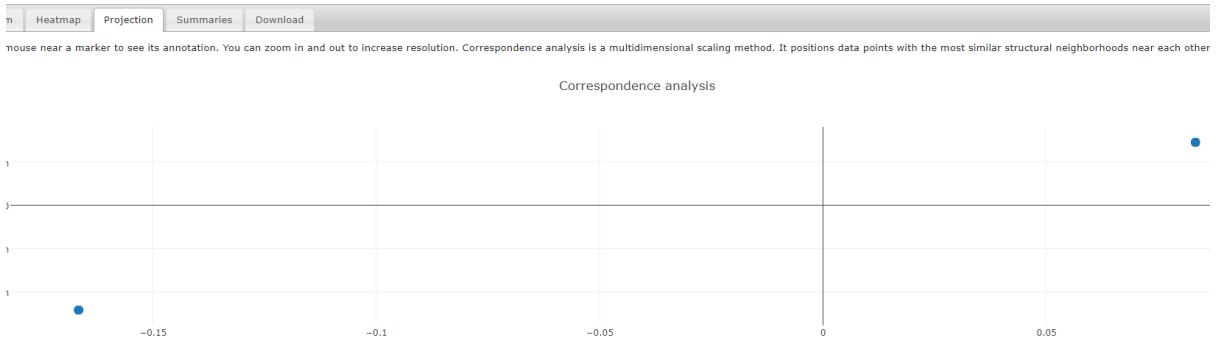
Load whole structures

Show/hide other chains of:

All Query



Τέλος η προβολή στο συγκεκριμένο προβλημα δείχνει μόνο 2 επιμέρους τμήματα πρωτεΐνών. Μάλλον επειδή τα υπόλοιπα είναι πολύ απομακρυσμένα και δεν υπάρχει κάποια επιλογή στοίχησης.



Ερώτημα 3

Οι κ συμβολοσειρές μπορούν να αποθηκευτούν σε ένα γενικευμένο δέντρο επιθεμάτων σαν μια ενιαία συμβολοσειρά που θα είναι η συνένωση όλων των συμβολοσειρών, όπου ανάμεσα σε κάθε μία από τις συμβολοσειρές μπορεί να παρεμβάλλεται κάποιος ειδικός χαρακτήρας που δεν ανήκει στο αλφαριθμητό τους (για παράδειγμα ο χαρακτήρας "\$"). Δηλαδή για δεδομένο σύνολο με τις συμβολοσειρές x_1, x_2, \dots, x_k , το δέντρο θα κατασκευαστεί από τη συμβολοσειρά $x_1\$x_2\$...x_k$$. Για να μπορούν να προστίθενται ή να αφαιρούνται από αυτό συμβολοσειρές πρέπει να μπορεί να διατηρεί και να προσαρμόζει «δυναμικά» τη δομή του, δηλαδή να είναι δυναμικό δέντρο επιθεμάτων (dynamic suffix tree). Πρέπει επίσης να μπορεί να γίνεται αναζήτηση για ταιριάσματα συμβολοσειρών στο δέντρο που δεν έχει σταθερή δομή (dynamic string matching).

Για τις παραπάνω λειτουργίες απαιτείται όμως περισσότερος χρόνος (για αναδιοργάνωση της δομής μετά από κάθε εισαγωγή ή διαγραφή και για αναζήτηση στο δέντρο) και χώρος (γιατί η κατάσταση του δέντρου πριν και μετά από κάθε αλλαγή πρέπει να διατηρείται στη μνήμη του υπολογιστή). Η αναζήτηση για ταιριάσματα μπορεί να γίνει πολύ χρονοβόρα, ειδικά όταν πρόκειται για βιολογικές ακολουθίες, που είναι μεγάλες σε μέγεθος. Επιπλέον, η διαγραφή μιας συμβολοσειράς x_j

γίνεται ουσιαστικά με διαγραφή των φύλλων στο δέντρο που αντιστοιχούν στις υποσυμβολοσειρές της χ. όμως η κατάσταση των εσωτερικών κόμβων του δέντρου δεν ανανεώνεται αμέσως στη μνήμη και έτσι ο συνολικός χώρος για την αποθήκευση της δομής αυξάνεται.

Μια λύση για βελτίωση της χρονικής και χωρικής πολυπλοκότητας είναι να αποθηκεύονται οι κ συμβολοσειρές ως γραμμές ή στήλες ενός μητρώου που θα αποθηκεύεται στη μνήμη. Οι εισαγωγές/διαγραφές συμβολοσειρών μπορούν να γίνονται με εισαγωγή/διαγραφή γραμμών ή στηλών. Κάθε υποσυμβολοσειρά, που κατά την αναζήτηση ταιριάσματος θα είναι ουσιαστικά ένα υπομητρώο του μητρώου αυτού, θα αποθηκεύεται σε ένα ξεχωριστό dynamic suffix tree.

Μια άλλη λύση είναι τα dynamic Fully Compressed Suffix Trees (FCSTs). Αποτελούν βελτίωση των στατικών Compresses Suffix Trees για δυναμικά μεταβαλλόμενα δεδομένα. Ένα FCST αποτελείται από έναν πίνακα Compressed Suffix Array (που χρησιμοποιείται ως ευρετήριο για τις συμβολοσειρές που είναι αποθηκευμένες σε αυτό), από ένα δ-Sampled tree (που εκμεταλλεύεται τις ιδιότητες των suffix trees ώστε να μπορούν να αποθηκεύονται μόνο κάποιοι κρίσιμοι κόμβοι και να μη χρειάζεται αποθήκευση ολόκληρης της δενδρικής δομής) και από αντιστοιχίσεις ανάμεσα στις δύο αυτές δομές. Το FCST μεταβάλλεται με χρήση ενός κανόνα που τοποθετεί νέα στοιχεία από δεξιά προς τα αριστερά δημιουργώντας ένα suffix tree ως «ενδιάμεση» κατασκευή. Έτσι μπορούν να αναπαρασταθούν όλο και μεγαλύτερες συμβολοσειρές. Κάθε αλλαγή στο δέντρο γίνεται μόνο στα φύλλα και αλλαγές στους εσωτερικούς κόμβους γίνονται μόνο αν χρειάζεται. Τα δέντρα αυτά πετυχαίνουν βέλτιστη χωρική πολυπλοκότητα.

Πηγές:

1. Y. Choi, T.W. Lam, Dynamic suffix tree and two-dimensional texts management, Information Processing Letters, Volume 61, Issue 4, 1997, Pages 213-220, ISSN 0020-0190, [https://doi.org/10.1016/S0020-0190\(97\)00018-5](https://doi.org/10.1016/S0020-0190(97)00018-5) .
2. Russo, L.M.S., Navarro, G., Oliveira, A.L. (2008). Dynamic Fully-Compressed Suffix Trees. In: Ferragina, P., Landau, G.M. (eds) Combinatorial Pattern Matching. CPM 2008. Lecture Notes in Computer Science, vol 5029. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69068-9_19

Ερώτημα 4

Ως «κενό» ανάμεσα σε δύο εμφανίσεις μιας συμβολοσειράς μπορεί να οριστεί ο αριθμός των συμβόλων ανάμεσά τους. Ένα ζευγάρι εμφανίσεων είναι μέγιστο (maximal) όταν καμία από τις δύο δεν μπορεί να επεκταθεί προς τα δεξιά ή προς τα αριστερά χωρίς να πάψει να υπάρχει ταίριασμα μεταξύ τους. Ένα ζευγάρι είναι αριστερά μέγιστο (left-maximal) αν τα σύμβολα στα αριστερά των δύο εμφανίσεων είναι διαφορετικά και αντίστοιχα δεξιά μέγιστο (right-maximal) αν τα σύμβολα στα δεξιά είναι διαφορετικά. Έχει σημασία να προσδιοριστεί τι θα θεωρείται «κενό» ανάμεσα σε δύο εμφανίσεις για να φαίνεται πότε υπάρχει επικάλυψη και πότε δύο εμφανίσεις βρίσκονται σε μεγάλη απόσταση μεταξύ τους μέσα στη συμβολοσειρά. Σε κάποιες περιπτώσεις μπορεί μια συμβολοσειρά να επαναλαμβάνεται δύο διαδοχικές φορές, δηλαδή το κενό έχει μήκος 0 (tandem repeat).

Μια προσέγγιση [1] που βάζει περιορισμούς στα κενά ανάμεσα σε δύο εμφανίσεις της ίδιας συμβολοσειράς θεωρεί ότι ένα ζευγάρι εμφανίσεων συμβολοσειράς α είναι μέγιστο μόνο αν το κενό βρίσκεται στο διάστημα ανάμεσα σε δύο συναρτήσεις $g1(|a|)$ και $g2(|a|)$. Οι υποσυμβολοσειρές αποθηκεύονται σε ένα δυαδικό δέντρο επιθεμάτων. Τα μέγιστα ζευγάρια όλης της συμβολοσειράς

μπορούν να βρεθούν αφού πρώτα βρεθούν όλα τα δεξιά-μέγιστα ζευγάρια στο δέντρο με κενά στο διάστημα ανάμεσα στις δύο συναρτήσεις και στη συνέχεια αφαιρεθούν από αυτά όλα τα δεξιά μέγιστα που δεν είναι και αριστερά μέγιστα. Για να μπορεί να γίνει αναζήτηση, χρησιμοποιούνται δέντρα αναζήτησης, για παράδειγμα AVL, όπου η πληροφορία αποθηκεύεται στους κόμβους Καλούνται leaf-list trees και κατασκευάζονται βήμα-βήμα με διαπέραση του αρχικού δέντρου και με βάση τις λίστες φύλλων (Leaf Lists) κάθε κόμβου. Για κάθε κόμβο ένα leaf list αποθηκεύει όλες τις υποσυμβολοσειρές που καταλήγουν σε συγκεκριμένο επίθεμα(δηλαδή τους δείκτες προς τα ανίστοιχα φύλλα του δέντρου). Για να μπορεί να γίνει το φιλτράρισμα χωρίς να χρειάζεται να βρεθούν όλα τα δεξιά μέγιστα κατασκευάζεται ένα επιπλέον βιοηθητικό AVL δέντρο, το block-start tree, που διατηρεί συνδέσμους προς το leaf-list tree. Τα leaf-list trees αποθηκεύουν την πληροφορία και τα block-start trees αποθηκεύουν το υποσύνολο των στοιχείων που ξεκινούν σε κάθε «μπλοκ». Για τα φύλλα, τα δέντρα αυτά κατασκευάζονται απευθείας. Για τους εσωτερικούς κόμβους κατασκευάζονται με συγχωνεύσεις υποδέντρων. Όμως η ενημέρωση των AVL δέντρων κοστίζει σε χρόνο. Μια εναλλακτική δομή είναι η χρήση δέντρων σωρού (heap trees) και μιας στοίβας στην οποία θα αποθηκεύεται ουσιαστικά το μονοπάτι (backbone) από τη ρίζα προς κάποιο φύλλο που δημιουργείται ακολουθώντας σε κάθε βήμα το «δεξιότερο» παιδί. Οπότε στην περίπτωση αυτή το «κενό» θα περιορίζεται μόνο από ένα κάτω όριο $g1(|\alpha|)$ αντί για διάστημα. Αντίστοιχη είναι και η χρήση χρωματισμένων δέντρων (colored trees). Οι προσεγγίσεις αυτές έχουν ίδια χωρική πολυπλοκότητα χειρότερης περίπτωσης με την προηγούμενη, $O(n)$, αλλά βελτιώνουν τη χρονική πολυπλοκότητα από $O(nlogn + z)$ σε $O(n + z)$, όπου n το μήκος της αρχικής συμβολοσειράς και z ο αριθμός των ζευγών μέγιστων εμφανίσεων.

Μια δεύτερη προσέγγιση [3] γενικεύει την έννοια των ζευγών εμφανίσεων για εμφανίσεις της ίδιας υποσυμβολοσειράς περισσότερες από δύο φορές στην ίδια συμβολοσειρά: τις πολυεμφανίσεις (multirepeats). Θεωρεί ένα σύνολο από N συμβολοσειρές, όπου κάθεμία έχει μήκος n ή (στην περίπτωση που δεν έχουν όλες το ίδιο μήκος) ο μέσος όρος των μηκών τους είναι n . Στόχος είναι να βρεθούν όλες οι επαναλήψεις μιας συμβολοσειράς από αλφάριθμο μεγέθους σ που συναντώνται τουλάχιστον m φορές σε κάθε μια από q συμβολοσειρές του συνόλου, χωρίς περιορισμούς στο μήκος των κενών. Μια πολυεμφάνιση F χαρακτηρίζεται left ή right maximal με βάση τα ίδια κριτήρια με προηγούμενως, αν δηλαδή δεν μπορεί να επεκταθεί. Το multirepeat πρέπει να εμφανίζεται τον ίδιο αριθμό φορών σε κάθε συμβολοσειρά για να ληφθεί υπ' όψη από τον αλγόριθμο. Γίνεται χρήση generalized suffix trees με την παραδοχή ότι είναι δυαδικά. Για να βρεθούν οι πολυεμφανίσεις χωρίς περιορισμούς στα κενά, συγκρίνονται με bottom-up τρόπο αναδρομικά κόμβοι που έχουν κοινό «πατέρα» στο δέντρο και συνενώνονται οι leaf-lists τους. Η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(\sigma N^2 + z)$. Αν υπάρχουν περιορισμοί στα κενά, οι leaf-lists κατασκευάζονται ως finger search trees σε φάσεις και με τη βοήθεια ενός γράφου. Στην περίπτωση αυτή η πολυπλοκότητα γίνεται $O((c^2 + \sigma)N^2 nlog(nN) + z)$, όπου c ο περιορισμός για τα κενά. Οι αλγόριθμοι της παραπάνω προσέγγισης έχουν το μειονέκτημα ότι η χρονική πολυπλοκότητα παραμένει μικρή μόνο όταν το μήκος των κενών είναι μικρό.

Μια τρίτη προσέγγιση [2] εξετάζει την περιοδικότητα (periodicity) συμβολοσειρών. Μια συμβολοσειρά S είναι ημιπεριοδική (quasiperiodic) αν μπορεί να κατασκευαστεί από συνενώσεις και υπερθέσεις μιας μικρότερης συμβολοσειράς a , που λέγεται ότι «καλύπτει» την S . Άλλιώς, αν η S δεν μπορεί να καλυφθεί από κάποια μικρότερη υποσυμβολοσειρά, είναι superprimitive. Παρουσιάζεται ένας αλγόριθμος που βρίσκει τις μέγιστα ημιπεριοδικές υποσυμβολοσειρές σε μια συμβολοσειρά πρώτα με κατασκευή και μετά με bottom-up διαπέραση ενός suffix tree. Ο αλγόριθμος αυτός έχει χρονική πολυπλοκότητα $O(nlogn)$.

Ένα πρόβλημα που φαίνεται να είναι κοινό στις παραπάνω προσεγγίσεις είναι ότι η τελική χρονική πολυπλοκότητα των αλγορίθμων μπορεί να γίνει μεγάλη γιατί η κατασκευή των δέντρων

επιθεμάτων εξαρτάται από το μέγεθος και τη συχνότητα των κενών ανάμεσα σε δύο επαναλήψεις μιας συμβολοσειράς.

Μια εναλλακτική προσέγγιση [4] που θα μπορούσε να είναι αποδοτική μπορεί να είναι κάποια παραλλαγή της χρήσης k-μερών, όπου να χρησιμοποιείται ένα ευρετήριο με καταχωρημένες υποσυβολοσειρές της αρχικής συμβολοσειράς και κάθε φορά που προκύπτει νέα εμφάνιση μιας συμβολοσειράς να αυξάνεται ένας δείκτης που θα μετράει τον αριθμό των εμφανίσεων. Σε μια τέτοια παραλλαγή όμως πρέπει κάπως να ληφθούν υπ' όψη και οι περιορισμοί για τα κενά. Επιπλέον, τα k-μερή είναι ευαίσθητα στις μεταλλάξεις, που στις βιολογικές συμβολοσειρές συμβαίνουν συχνά [5]. Γι' αυτό ίσως να είναι αποδοτικότερη η μελέτη της κατανομής των k-μερών στις περιοχές της συμβολοσειράς όπου υπάρχουν ταιριάσματα μέσω διανυσμάτων (vector seeds), ή η χρήση με κάποιον τρόπο ομάδων από k-μερή που να είναι διασυνδεδεμένα μεταξύ τους (όπως τα strobemers).

Πηγές:

1. Gerth Stølting Brodal, Rune B. Lyngsø, Christian N. S. Pedersen, Jens Stoye: Finding Maximal Pairs with Bounded Gap. CPM 1999: 134-149
2. Gerth Stølting Brodal, Christian N. S. Pedersen: Finding Maximal Quasiperiodicities in Strings. CPM 2000: 397-411
3. A. Bakalis, Costas S. Iliopoulos, Christos Makris, Spyros Sioutas, Evangelos Theodoridis, Athanasios K. Tsakalidis, Kostas Tsichlas: Locating Maximal Multirepeats in Multiple Strings Under Various Constraints. Comput. J. 50(2): 178-185 (2007).
4. Back to sequences: find the origin of k-mers Anthony Baire, Pierre Peterlongo bioRxiv 2023.10.26.564040; doi: <https://doi.org/10.1101/2023.10.26.564040>
5. Sahlin K. Effective sequence similarity detection with strobemers. Genome Res. 2021 Nov;31(11):2080-2094. doi: 10.1101/gr.275648.121. Epub 2021 Oct 19. PMID: 34667119; PMCID: PMC8559714.

Ερώτημα 5

Ο αλγόριθμος θα μπορούσε να έχει την παρακάτω περιγραφή:

Ξεκινά από τη ρίζα.

Αρχικοποιεί έναν μετρητή για τους χαρακτήρες σε 0 και έναν άλλο μετρητή για τον αριθμό των εμφανίσεων του προτύπου σε 0. Ξεκινά την διαπέραση του δέντρου χαρακτήρα προς χαρακτήρα.

Για κάθε χαρακτήρα που συναντά, ελέγχει αν είναι ο επόμενος χαρακτήρας του προτύπου (αρχικά έλεγχος αν είναι ο πρώτος χαρακτήρας του προτύπου).

Αν είναι, αυξάνει τον μετρητή χαρακτήρων κατά 1 και συνεχίζει με τον επόμενο χαρακτήρα του κλάδου του δέντρου στον οποίο βρίσκεται.

Αν δεν ανήκει, μηδενίζει τον μετρητή χαρακτήρων και ξεκινά την ίδια διαδικασία από τον χαρακτήρα στον οποίο βρίσκεται.

Αν ο μετρητής γίνει ίσος με το μήκος του προτύπου (4 για το δέντρο της εκφώνησης), το πρότυπο έχει βρεθεί. Αυξάνει τον μετρητή εμφανίσεων κατά 1, μηδενίζει τον μετρητή χαρακτήρων και εμφανίζει στην έξοδο την συμβολοσειρά που ξεκινά από τη ρίζα και καταλήγει στη θέση αυτή του δέντρου. Ξεκινά την ίδια διαδικασία από τον χαρακτήρα στον οποίο βρίσκεται.

Αν φτάσει σε φύλλο και έχει βρεθεί το πρότυπο, εμφανίζει στην έξοδο την συμβολοσειρά που ξεκινά από τη ρίζα και καταλήγει στο φύλλο αυτό. Ξεκινά την ίδια διαδικασία στον επόμενο κλάδο του δέντρου όπως θα προσπελαζόταν το δέντρο σε μια κατά βάθος αναζήτηση.

Αν φτάσει σε φύλλο χωρίς να έχει βρει το πρότυπο, μηδενίζει τον μετρητή χαρακτήρων και ξεκίνα την ίδια διαδικασία στον επόμενο κλάδο του δέντρου όπως θα προσπελαζόταν σε μια κατά βάθος αναζήτηση.

Ερώτημα 6

*	-	T	C	G	T	G	A	A	T	T
I	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
G	-1	(-1) → -2	-1	-2	-3	-4	-5	-6	-7	-7
U	-2	-2	(-2)	-2	-2	-3	-4	-5	-6	-7
G	-3	-3	-3	(-1) → -2	-1	-2	-3	-4	-5	-5
T	-4	-2	-3	-2	(-1) → -2	-1	-2	-3	-2	-1
T	-5	-3	-3	-3	(-1) → -2	-1	-2	-3	-2	-1
G	-6	-4	-4	-2	-2	(-1) → -2	-3	-2	-1	-2
T	-7	-5	-5	-3	-1	-1	(-1) → -2	-3	-2	-2
G	-8	-6	-6	-4	-2	0	-1	-2	-2	-2
G	-9	-7	-7	-5	-3	-1	-1	-2	-3	-3
$\text{TCG} - \text{TGAAT} - \text{T}$ (và vw) - σην διαδρομή GUG-T-T-GGG $\text{(và zw)} \quad \text{en διαδρομή} \quad \text{TCG} - \text{TGAAT} - \text{T}$ $\text{GUGTTG} - \text{TGG}$										

Αρχικές ακολουθίες:

v= GUGTTGTGG

w= TCGTGAATT

Τελικές ολικές στοιχήσεις με Needleman-Wunsch:

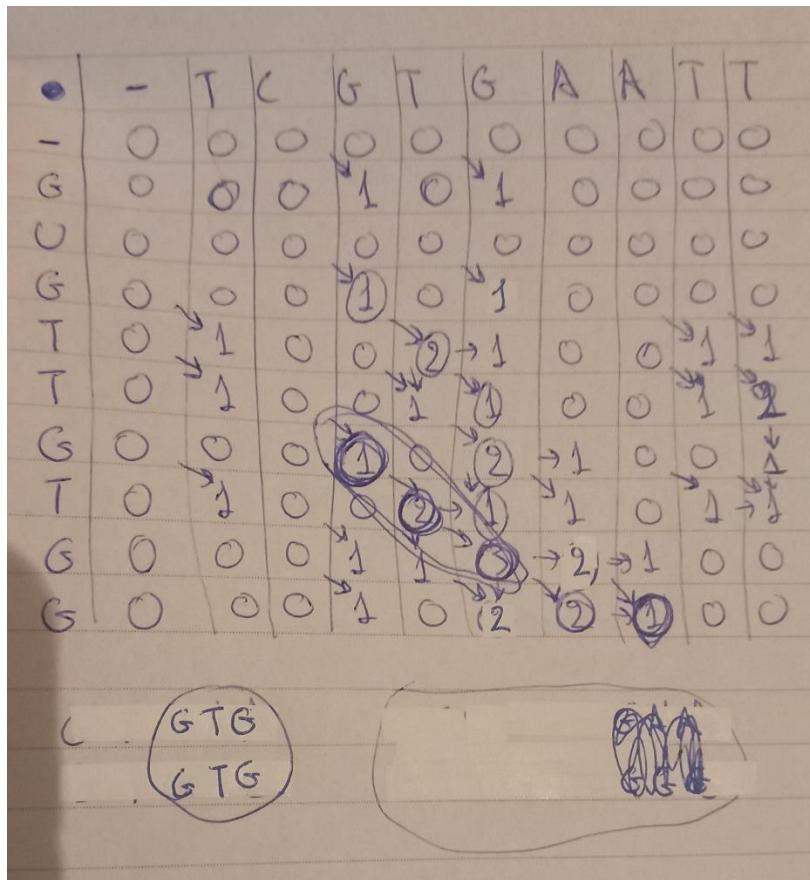
TCG-TGAAT-T

GUGTG--TGG

Τελικές τοπικές στοιχήσεις με Smith-Waterman:

GTG ή T-GTG

GTG T-GTG



Ερώτημα 7

```

import re
import collections
from Bio import SeqIO

#define the transcription factors, use of regular expressions
tf1 = 'BHTGTGGTYW'
p1 = r'([^A][^G](TGTGGT)[CT][AT])'

tf2 = 'WGACAGB'
p2 = r'([AT](GACAG)[^A])'

tf3 = 'BTGGGARD'
p3 = r'([^A](TGGGA)[AG][^C])'

#read the sequence from file "Sequence.txt"
try:
    with open("Sequence.txt", 'r') as dna:
        for record in SeqIO.parse(dna, "fasta"):
            dna_seq = record.seq
except FileNotFoundError:
    print("No fasta file found")
    exit()

```

```

#gen = readDna('Sequence.txt')
#print(gen, '\n')

#convert the dna sequence to string
gen = str(dna_seq)

#function to find matches
def findTranscriptionFactors(s, file):

    #search for tf1 matches
    found1 = re.finditer(p1, s)
    for i in found1:
        print(f"Match found for '{tf1}' at position: {i.start()} of genome\n")
        with open('Finds.txt', 'a') as matches:
            matches.write(f"Match found for '{tf1}' at position: {i.start()} of genome\n")
        matches.close()

    #search for tf2 matches
    found2 = re.finditer(p2, s)
    for i in found2:
        print(f"Match found for '{tf2}' at position: {i.start()} of genome\n")
        with open('Finds.txt', 'a') as matches:
            matches.write(f"Match found for '{tf2}' at position: {i.start()} of genome\n")
        matches.close()

    #search for tf3 matches
    found3 = re.finditer(p3, s)
    for i in found3:
        print(f"Match found for '{tf3}' at position: {i.start()} of genome\n")
        with open('Finds.txt', 'a') as matches:
            matches.write(f"Match found for '{tf3}' at position: {i.start()} of genome\n")
        matches.close()

findTranscriptionFactors(gen, 'Finds.txt')

```

```

#function to complement the given sequence
def complementGenome(s, file):
    complement = {'A': 'T', 'T': 'A', 'C': 'G', 'G': 'C'}
    emoneg = ''
    with open(file, 'a') as changes_file:
        for i in s:
            emoneg = complement[i]
            changes_file.write(emoneg)
    changes_file.close()
complementGenome(gen, 'Changes.txt')

```

```

#function to create the sequence after transcription to rna
def turnToRna(s, file):
    transcr = {'T': 'U', 'A': 'A', 'C': 'C', 'G': 'G'}

```

```

rna = ''
with open(file, 'a') as transcr_file:
    for i in s:
        rna = rna + transcr[i]
        transcr_file.write(rna)
transcr_file.close()
turnToRna(gen, 'Rna.txt')

#function to count how many times each base appears in given sequence
#and to calculate percentage of CG compared to other bases
def countBases(s, file):
    seq_length = len(s)
    print('Total number of bases:', seq_length, '\n')
    char_count = collections.Counter(s)
    print('Number of occurrences of each base in sequence:', char_count, '\n')
    gc = 0
    percent_gc = 0
    for c in s:
        for i in range(len(s)):
            if s[i] == 'C' or s[i] == 'G':
                gc += 1
    if gc > 0:
        percent_gc = (gc / seq_length) * 100
        print('Percentage of "CG" compared to other bases in given sequence
is:', percent_gc, '\n')
    with open(file, 'a') as count_file:
        count_file.write(str(seq_length))
        count_file.write('\n')
        count_file.write(str(char_count))
        count_file.write('\n')
        count_file.write(str(percent_gc))
    count_file.close()
countBases(gen, "Count.txt")

```

#Το αρχείο Sequence.txt περιέχει την ακολουθία όπως δίνεται στην εκφώνηση.
#Μέρος του κώδικα βασίζεται σε παραδείγματα που υπάρχουν στον σύνδεσμο:
<https://www.youtube.com/playlist?list=PL2mpR0RYFQsBiCWVJSvVAO3OJ2t7DzoHA>