

Εργασία

Βιοπληροφορική

Ακαδημαϊκό έτος 2018 - 2019



Π16036 – Ιωαννίδης Παναγιώτης
Π16112 – Παραβάντης Αθανάσιος

Περιεχόμενα

1	Άσκηση 6.14.....	3
1.1	Εκφώνηση.....	3
1.2	Σχεδιασμός	3
1.3	Υλοποίηση.....	3
1.4	Αποτέλεσμα	4
1.5	Αναφορές.....	4
2	Άσκηση 6.15.....	5
2.1	Εκφώνηση.....	5
2.2	Σχεδιασμός	5
2.3	Υλοποίηση.....	5
2.4	Αποτέλεσμα	5
2.5	Αναφορές.....	6
3	Άσκηση 6.22.....	7
3.1	Εκφώνηση.....	7
3.2	Σχεδιασμός	7
3.3	Υλοποίηση.....	7
3.4	Αποτέλεσμα	8
3.5	Αναφορές.....	9
4	Άσκηση 6.27.....	10
4.1	Εκφώνηση.....	10
4.2	Σχεδιασμός	10
4.3	Υλοποίηση.....	10
4.4	Αποτέλεσμα	10
4.5	Αναφορές.....	11
5	Άσκηση 6.37.....	12
5.1	Εκφώνηση.....	12
5.2	Σχεδιασμός	12
5.3	Υλοποίηση.....	12
5.4	Αποτέλεσμα	12
5.5	Αναφορές.....	14
6	Άσκηση 11.4.....	15
6.1	Εκφώνηση.....	15

6.2	Σχεδιασμός	15
6.3	Υλοποίηση.....	15
6.4	Αποτέλεσμα	17
6.5	Αναφορές.....	17
7	Άσκηση 11.6.....	18
7.1	Εκφώνηση.....	18
7.2	Σχεδιασμός	18
7.3	Υλοποίηση.....	19
7.4	Αποτέλεσμα	20
7.5	Αναφορές.....	21
8	Βοηθητικά αρχεία	22

1 Άσκηση 6.14

1.1 Εκφώνηση

Δύο παίκτες παίζουν το εξής παιχνίδι με δύο αλληλουχίες που έχουν μήκος n και m νουκλεοτίδια αντίστοιχα. Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να αφαιρέσει δύο νουκλεοτίδια από τη μία αλληλουχία (είτε την πρώτη είτε τη δεύτερη) και ένα νουκλεοτίδιο από την άλλη. Ο παίκτης που δεν μπορεί να κάνει κίνηση κερδίζει. Ποιος θα κερδίσει; Περιγράψτε τη νικηφόρα στρατηγική για όλες τις τιμές των n και m .

1.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7.

1.3 Υλοποίηση

Η άσκηση 6.14 έχει βασιστεί στην παραδοχή ότι δυο παίκτες παίζουν μεταξύ τους. Έτσι, λοιπόν, έχει γίνει προσομοίωση των παιχτών, τους οποίους χειρίζεται ο υπολογιστής. Ο πρώτος παίκτης παίζει με στρατηγική που σκοπό έχει να τον οδηγήσει σε κατάσταση νίκης. Από την άλλη οι κινήσεις του δεύτερο παίχτη είναι εντελώς τυχαίες.

Έστω, λοιπόν, m η ακολουθία 1 και n η ακολουθία 2. Ο πρώτος παίκτης θα βρίσκεται σε μειονεκτική θέση όταν:

$$m = 3k \text{ and } n \geq 3k \text{ για } k \in \mathbb{N}$$

$$n = 3k \text{ and } m \geq 3k \text{ για } k \in \mathbb{N}$$

$$m = n = 3k + 1 \text{ για } k \in \mathbb{N}$$

Για παράδειγμα:

$$m = 0 \text{ and } n \geq 0, m = 3 \text{ και } n \geq 3, m = 6 \text{ και } n \geq 6, \dots$$

$$n = 0 \text{ και } m \geq 0, n = 3 \text{ και } m \geq 3, n = 6 \text{ και } m \geq 6, \dots$$

$$m = n = 1, m = n = 4, m = n = 7, \dots$$

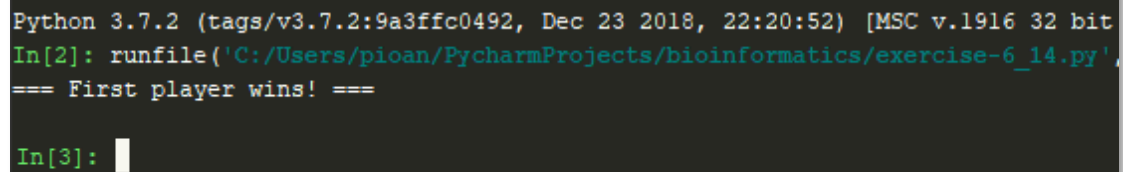
Επομένως εμείς με τον με πρέπει να τον φέρουμε σε πλεονεκτικές, για εκείνον, θέσεις.

Η ακολουθίες από επιλέξαμε για την εκτέλεση της άσκησης είναι αυτές της α-λακταλβουμίνη και της λυσοζύμης c.

1.4 Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6_14.py**.

Το αποτέλεσμα που εμφανίζεται είναι ο παίκτης που νικάει το παιχνίδι, όπως φαίνεται και στην εικόνα 1.1.



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit  
In[2]: runfile('C:/Users/pioan/PycharmProjects/bioinformatics/exercise-6_14.py',  
=== First player wins! ===  
  
In[3]:
```

Εικόνα 1.1

1.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD

2 Άσκηση 6.15

2.1 Εκφώνηση

Δύο παίκτες παίζουν το παρακάτω παιχνίδι με μια νουκλεοτιδική αλληλουχία που έχει μήκος n . Σε κάθε γύρο του παιχνιδιού, ένας παίκτης μπορεί να αφαιρέσει είτε ένα είτε δύο νουκλεοτίδια από την αλληλουχία. Ο παίκτης που αφαιρεί το τελευταίο γράμμα κερδίζει. Ποιος θα κερδίσει; Περιγράψτε τη νικηφόρα στρατηγική για όλες τις τιμές του n .

2.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7.

2.3 Υλοποίηση

Και αυτό το παιχνίδι αποτελείται από δυο παίκτες, οι οποίοι προσομοιώνονται από τον υπολογιστή. Η ακολουθία που επιλέξαμε για το παιχνίδι είναι η λυσοζύμης c .

Ο πρώτος παίκτης επιλέγει την κίνηση του με βάση το υπόλοιπο της διαίρεσης του πλήθους των νουκλεοτιδίων, που υπάρχουν κάθε στιγμή στη ακολουθία, και του 3. Αν το υπόλοιπο είναι ίσο με μηδέν τότε επιλέγει να αφαιρέσει ένα νουκλεοτίδιο, αν το υπόλοιπο είναι ίσο με ένα τότε επιλέγει να αφαιρέσει δυο νουκλεοτίδια αλλιώς η επιλογή γίνεται στην τύχη.

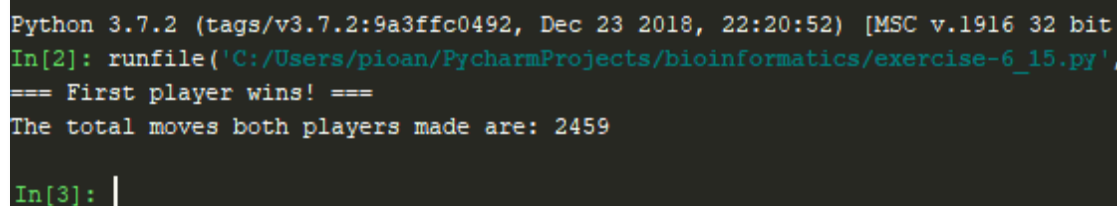
Οι επιλογές του δεύτερου παίκτη είναι καθαρά τυχαίες.

Με αυτόν τον τρόπο, θέτουμε τον πρώτο παίκτη σε πλεονεκτική θέση η οποία θα τον οδηγήσει στη νίκη.

2.4 Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο `bioinformatics` εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6_15.py**.

Το αποτέλεσμα που εμφανίζεται είναι ο παίκτης που νικάει το παιχνίδι καθώς και οι συνολικές κινήσεις και των δυο παικτών, εικόνα 2.1.



```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit  
In[2]: runfile('C:/Users/pioan/PycharmProjects/bioinformatics/exercise-6_15.py',  
=== First player wins! ===  
The total moves both players made are: 2459  
In[3]: |
```

Εικόνα 2.1

2.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- <https://en.wikipedia.org/wiki/Nim>

3 Άσκηση 6.22

3.1 Εκφώνηση

Ορίζουμε ότι η στοίχιση επικάλυψης μεταξύ δυο αλληλουχιών $\mathbf{v} = v_1 \dots v_n$ και $\mathbf{w} = w_1 \dots w_m$ είναι η στοίχιση ανάμεσα σε ένα πρόθεμα της \mathbf{v} και ένα επίθεμα της \mathbf{w} . Για παράδειγμα, αν $\mathbf{v} = \text{TATATA}$ και $\mathbf{w} = \text{AAATTT}$, τότε μια (όχι απαραίτητως βέλτιστη) στοίχιση επικάλυψης μεταξύ των \mathbf{v} και \mathbf{w} είναι η

ATA

AAA

Η βέλτιστη στοίχιση επικάλυψης είναι η στοίχιση που μεγιστοποιεί τη βαθμολογία της καθολικής στοίχισης μεταξύ των v_i, \dots, v_n και w_1, \dots, w_j , όπου το μέγιστο υπολογίζεται για όλα τα προθέματα v_i, \dots, v_n της \mathbf{v} και όλα τα επιθέματα w_1, \dots, w_j της \mathbf{w} .

Διατυπώστε έναν αλγόριθμο που υπολογίζει τη βέλτιστη στοίχιση επικάλυψης και εκτελείται σε χρόνο $O(nm)$.

3.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και τη βιβλιοθήκη Biopython (<https://biopython.org/>).

3.3 Υλοποίηση

Για να υπολογίσουμε τη βέλτιστη στοίχιση επικάλυψης δυο ακολουθιών σύμφωνα με τα ζητούμενα της άσκησης, το πρώτο βήμα που πρέπει να κάνουμε είναι να βρούμε όλα τα πιθανά προθέματα της \mathbf{v} και όλα τα πιθανά επιθέματα της \mathbf{w} . Αυτό μπορούμε να το επιτύχουμε εύκολα, δημιουργώντας δυο λίστες οι οποίες περιέχουν τους κατάλληλους συνδυασμούς υποακολουθιών προς σύγκριση. Οι υποακολουθίες αυτές, έχουν μήκος από $n = 1$ μέχρι και $n = N - 1$ όπου το N είναι το μήκος της αρχικής ακολουθίας \mathbf{v} ή \mathbf{w} .

Ως δεδομένα για την άσκηση, χρησιμοποιήσαμε τη λυσοζύμη c και την πρωτεΐνη α-λακταλβουμίνη από το NCBI. Παίρνουμε τους πρώτους 50 χαρακτήρες από τις δυο ακολουθίες για λόγους οικονομίας μνήμης και χρόνου εκτέλεσης του προγράμματος.

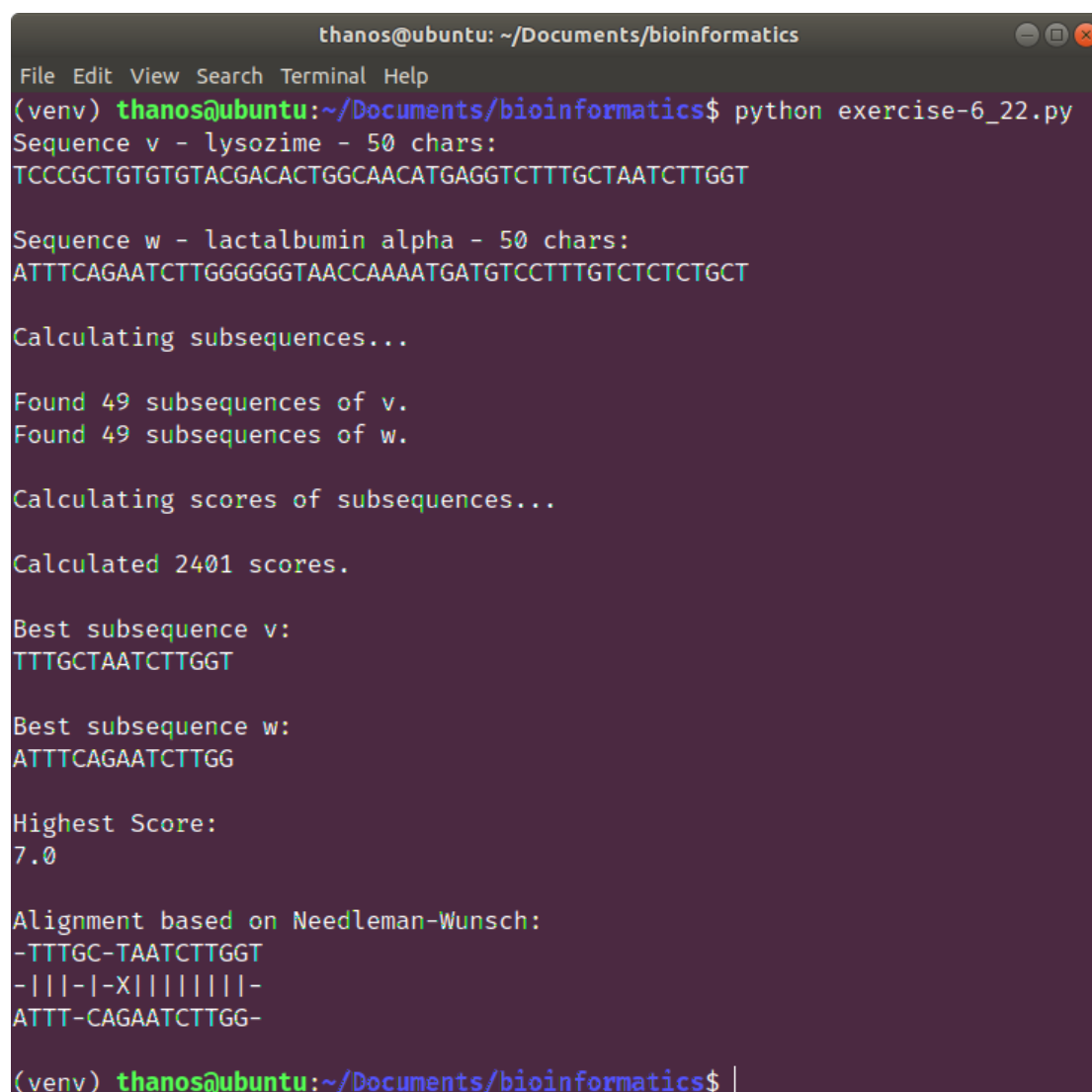
Έπειτα, αφού έχουμε δεδομένες όλες τις πιθανές υποακολουθίες της \mathbf{v} και \mathbf{w} , μπορούμε να αρχίσουμε τη διαδικασία βασιζόμενοι στη καθολική στοίχιση των Needleman-Wunsch. Για να έχουμε όμως τα επιθυμητά αποτελέσματα, οφείλουμε να κάνουμε μια τροποποίηση στη βαθμολόγηση του αλγορίθμου. Όταν υπάρχει αντιστοιχία μεταξύ δυο νουκλεοτιδίων, επιβραβεύουμε τον αλγόριθμο με +1 και σε κάθε άλλη περίπτωση τιμωρούμε τον αλγόριθμο με -1. Επομένως, θα έχουμε έναν

σταθερό τρόπο βαθμολόγησης που θα μας βοηθήσει να βρούμε την βέλτιστη στοίχιση επικάλυψης.

Τέλος, συγκρίνουμε μία προς μία όλες τις υποακολουθίες που μας αφορούν και συμπεραίνουμε με βάση τα κριτήρια βαθμολόγησης, ποιες υποακολουθίες έχουν την καλύτερη επίδοση. Έτσι εφόσον βρούμε μια v' και μια w' επιλέγουμε και εμφανίζουμε μια τυχαία καθολική στοίχιση στην οθόνη.

3.4 Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6_22.py** και ενδεικτικά εμφανίζονται τα παρακάτω αποτελέσματα, εικόνα 3.1.



```
thanos@ubuntu: ~/Documents/bioinformatics
File Edit View Search Terminal Help
(venv) thanos@ubuntu:~/Documents/bioinformatics$ python exercise-6_22.py
Sequence v - lysozyme - 50 chars:
TCCCGCTGTGTGTACGACACTGGCAACATGAGGTCTTTGCTAATCTTGGT

Sequence w - lactalbumin alpha - 50 chars:
ATTT-CAGAATCTTGGGGGGTAACCAAAATGATGTCCTTTGTCTCTCTGCT

Calculating subsequences...

Found 49 subsequences of v.
Found 49 subsequences of w.

Calculating scores of subsequences...

Calculated 2401 scores.

Best subsequence v:
TTTGCTAATCTTGGT

Best subsequence w:
ATTT-CAGAATCTTGG-

Highest Score:
7.0

Alignment based on Needleman-Wunsch:
-TTTGC-TAATCTTGGT
-|||-|-X|||||||-
ATTT-CAGAATCTTGG-
```

Εικόνα 3.1

3.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- https://www.ncbi.nlm.nih.gov/nucore/NC_006088.4?report=fasta&from=35425914&to=35429601
- https://www.ncbi.nlm.nih.gov/nucore/AC_000162.1?report=fasta&from=31347861&to=31349882

4 Άσκηση 6.27

4.1 Εκφώνηση

Για μια παράμετρο k , υπολογίστε την καθολική στοίχιση δύο συμβολοσειρών, με τον περιορισμό ότι η στοίχιση περιέχει το πολύ k κενά (μπλοκ με συνεχόμενες προσθαφαιρέσεις).

4.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και τη βιβλιοθήκη Biopython (<https://biopython.org/>).

4.3 Υλοποίηση

Ξεκινάμε την επίλυση της άσκησης ανατρέχοντας στην α-λακταλβουμίνη ως ακολουθία v και στη λυσοζύμη c ως την ακολουθία w . Επιλέγουμε τους πρώτους 50 χαρακτήρες της ακολουθίας v και έναν τυχαίο αριθμό από το 45 έως το 50 με πρώτους χαρακτήρες της w . Ακολουθούμε αυτή τη διαδικασία ώστε η w να είναι τις περισσότερες φορές μικρότερη σε μήκος από v , για να προσομοιώσουμε τα αποτελέσματα της άσκησης.

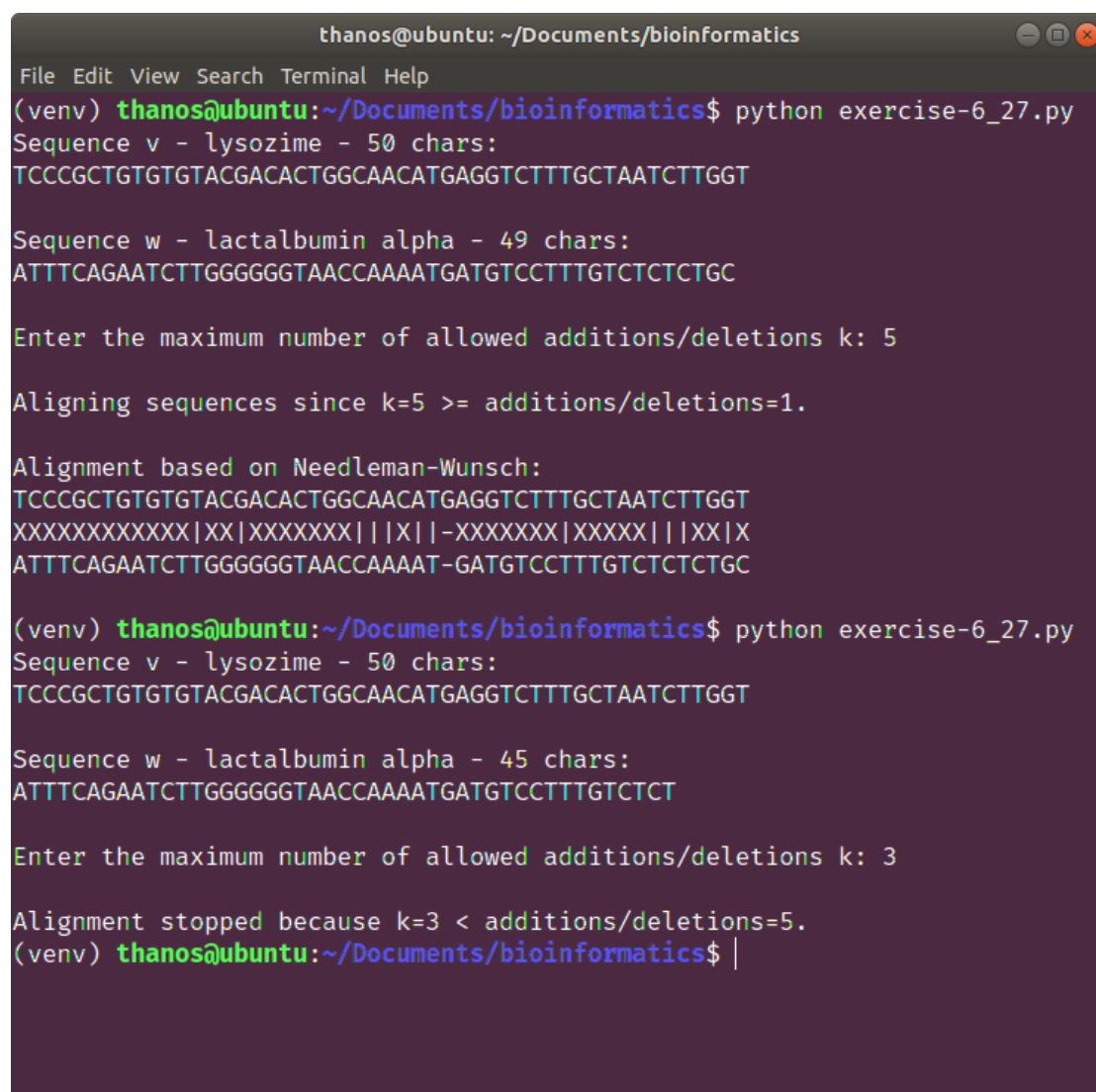
Αμέσως μετά, το πρόγραμμα ζητάει από τον χρήστη τον μέγιστο επιτρεπτό αριθμό προσθαφαιρέσεων k , που χρησιμοποιείται μετέπειτα ως κριτήριο για τη στοίχιση των δυο ακολουθιών. Ο αριθμός των k κενών προκύπτει από τη τροποποίηση του τρόπου βαθμολόγησης της καθολικής στοίχισης v και w , έχοντας $+0$ για ταίριασμα ή ασυμφωνία και -1 για κάθε άλλη περίπτωση προσθαφαίρεσης χαρακτήρων. Με αυτό τον τρόπο, ο αλγόριθμος των Needleman-Wunsch θα επιστρέφει πάντα μια βαθμολογία που αντιστοιχεί με τον αριθμό των κενών μεταξύ των v και w .

Η βαθμολογία που επιστρέφεται γίνεται θετικός ακέραιος αριθμός ώστε να συγκριθεί με την επίσης θετική ακέραια σταθερά k που καθόρισε ο χρήστης. Εάν ο αριθμός των κενών είναι μεγαλύτερος από το επιτρεπτό όριο k , το πρόγραμμα τερματίζει. Διαφορετικά, εάν είναι μικρότερος ή ίσος με την σταθερά k , τότε προχωράμε στην εμφάνιση τυχαίας καθολικής στοίχισης των v και w .

4.4 Αποτέλεσμα

Στην παρακάτω εικόνα έχουμε δυο στιγμιότυπα εκτέλεσης. Στη πρώτη περίπτωση, το $k = 5$ και κενά = 1 επομένως εκτελείται το πρόγραμμα και εμφανίζεται η καθολική στοίχιση. Στη δεύτερη περίπτωση, έχουμε $k = 3$ και κενά = 5 τερματίζει το πρόγραμμα και εμφανίζεται κατάλληλο μήνυμα.

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6_27.py** και ενδεικτικά εμφανίζονται τα παρακάτω αποτελέσματα, εικόνα 4.1.



```
thanos@ubuntu: ~/Documents/bioinformatics
File Edit View Search Terminal Help
(venv) thanos@ubuntu:~/Documents/bioinformatics$ python exercise-6_27.py
Sequence v - lysozyme - 50 chars:
TCCCGCTGTGTGTACGACACTGGCAACATGAGGTCTTTGCTAATCTTGGT

Sequence w - lactalbumin alpha - 49 chars:
ATTTTCAGAATCTTGGGGGGTAACCAAAATGATGTCCTTTGTCTCTCTGC

Enter the maximum number of allowed additions/deletions k: 5

Aligning sequences since k=5 >= additions/deletions=1.

Alignment based on Needleman-Wunsch:
TCCCGCTGTGTGTACGACACTGGCAACATGAGGTCTTTGCTAATCTTGGT
XXXXXXXXXXXX|XX|XXXXXXXX||X||-XXXXXXX|XXXXX|||XX|X
ATTTTCAGAATCTTGGGGGGTAACCAAAAT-GATGTCCTTTGTCTCTCTGC

(venv) thanos@ubuntu:~/Documents/bioinformatics$ python exercise-6_27.py
Sequence v - lysozyme - 50 chars:
TCCCGCTGTGTGTACGACACTGGCAACATGAGGTCTTTGCTAATCTTGGT

Sequence w - lactalbumin alpha - 45 chars:
ATTTTCAGAATCTTGGGGGGTAACCAAAATGATGTCCTTTGTCTCT

Enter the maximum number of allowed additions/deletions k: 3

Alignment stopped because k=3 < additions/deletions=5.
(venv) thanos@ubuntu:~/Documents/bioinformatics$ |
```

Εικόνα 4.1

4.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- https://www.ncbi.nlm.nih.gov/nuccore/NC_006088.4?report=fasta&from=35425914&to=35429601
- https://www.ncbi.nlm.nih.gov/nuccore/AC_000162.1?report=fasta&from=31347861&to=31349882

5 Άσκηση 6.37

5.1 Εκφώνηση

Στο πρόβλημα της Χιμαιρικής Στοίχισης, δίνονται μια συμβολοσειρά v και ένα σύνολο συμβολοσειρών $\{w_1, \dots, w_N\}$, και πρέπει να βρεθεί το $\max_{1 \leq i, j \leq N} s(v, w_i \circ w_j)$, όπου $w_i \circ w_j$ είναι η συνένωση των w_i και w_j και το $s(.,.)$ συμβολίζει τη βαθμολογία της βέλτιστης καθολικής στοίχισης.

5.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και τη βιβλιοθήκη Biopython (<https://biopython.org/>).

5.3 Υλοποίηση

Στην επίλυση του προβλήματος της Χιμαιρικής Στοίχισης χρησιμοποιούμε τον ιό του αφθώδη πυρετού και συγκεκριμένα τη τριδιάστατη δομή του καψιδίου. Έχουμε υπ' όψη την αμινοξική ακολουθία του ιού ώστε από αυτή να εξαγάγουμε τη νουκλεοτιδική ακολουθία που θα μας βοηθήσει στην επεξεργασία. Για την εξαγωγή του dna από την αμινοξική ακολουθία χρησιμοποιήσαμε τη συνάρτηση του Matlab aa2nt επειδή το toolbox Biopython δεν παρέχει κάποιον αντίστοιχο εύκολο τρόπο.

Αφού εκτυπώσουμε την αμινοξική και νουκλεοτιδική ακολουθία στην οθόνη, αποθηκεύουμε το dna και το χωρίζουμε ως εξής: παίρνουμε τα πρώτα 20 νουκλεοτίδια για να ορίσουμε την ακολουθία v , έπειτα χρησιμοποιούμε τα υπόλοιπα και τα χωρίζουμε σε ομάδες των 10 για να σχηματιστεί μια λίστα υποακολουθιών. Με αυτό το σκεπτικό έχουμε μια βάση πάνω στην οποία μπορούμε να κάνουμε την επεξεργασία που μας ζητείται.

Για τη λίστα w βρίσκουμε όλους τους συνδυασμούς υποακολουθιών $w_i + w_j$ και εφαρμόζουμε καθολική στοίχιση με βαθμολόγηση +1 για ταίριασμα και -1 σε κάθε άλλη περίπτωση. Βρίσκουμε την ακολουθία με το καλύτερο σκορ και κάνουμε ξανά καθολική στοίχιση με την αρχική ακολουθία v . Τέλος, εμφανίζουμε μια τυχαία στοίχιση στην οθόνη ως αποτέλεσμα.

5.4 Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-6_37.py** και ενδεικτικά εμφανίζονται τα παρακάτω αποτελέσματα, εικόνες 5.1 και 5.2.

```

thanos@ubuntu: ~/Documents/bioinformatics
File Edit View Search Terminal Help
(venv) thanos@ubuntu:~/Documents/bioinformatics$ python exercise-6_37.py
1BBT amino acid sequence:
TTSAGESADPVTITTVENYGGETQIQRRQHTDVSFIMDRFVKVTPQNQINILDLMQVPSHTLVGGGLLRASTYFF
SDLEIAVKGHGDLTWVPNGAPEKALDNTTNPATYHKAPLTRALPYTAPHRVLATVYNGGECRYSRNAVPNLRG
DLQVLQAKVARTLPTSFNYGAIKATRVTELLYRMKRAETCYCPRLLLAIHPTEARHKQKIVAPVKQTL

1BBT dna sequence:
ACGACGCTCGCGGGAGAAATCCGCAGACCCGGTTACGACGACCGTGGAAAAATTACGGGGGGGAGACTCAGATAC
AGAGAAGACACACACTGACGCTTTCCTTTATTATGGACCGTTTCGTTAAAGTCACGCGGCAAAATCAAATCAA
CATATTGGACCTCATGCAAGTCCCGTCACACACTCTAGTCGGAGGACTGCTGCGGGCTTCTACCTACTATTTT
TCCGACCTTGAAAATGCAAGTCAAACAGGAGGAGACTTTCGAGTGGGTCCTCCAAATGGTGCACCGGAGAAAGCAC
TTGATAATACTACGAATCCAATGCATATCACAAAGGCACCTTAAGTGGTTCGCTCCCGTATACCGCTCC
GCATAGAGTGGCTGGCTACCGTCTACAATGGGAATGCTTATAGCGGAAGACGCTGTGCCAAATCTCCGAGGT
GATTTACAAAGTCTTGTCTCAGAAGTCCGCGCGAAGCGTTCCTACGCTCATTTAATTATGTGCGATAAAGGCTA
CTCGAGTGACTGAATTGCTCTACCGGATGAAGCGCGCAGAACATATTTGCCAAGACCTCTGCTAGCCATCCA
TCCACAGAGGCGAGACACAAACAGAAGATCGTGGCGCCAGTAAAGCAAAACCTTT

Sequence v - 1BBT - 20 chars:
ACGACGCTCGCGGGAGAAATC

Sequence w - 1BBT - 62 splits of 10 chars:
['GACAGCCGG', 'TTACGAGCAC', 'CGTGGAAAAAT', 'TACGGGGGGG', 'AGACTCAGAT', 'A
CAGAGAAGA', 'CAACACACTG', 'ACGTTTCCTT', 'TATTATGGAC', 'CGCTTCGTTA', 'AAGT
CAGGCC', 'GCAAAATCAA', 'ATCAACATAT', 'TGGACCTCAT', 'GCAAGTCCGG', 'TCACACA
CTC', 'TAGTCGGAGG', 'ACTGCTGCGG', 'GCTTCTACCT', 'ACTATTTCTC', 'CGACCTTGAA
', 'ATTGCACTGA', 'AACACGAGGG', 'AGACTTGAGC', 'TGGGTCCCAA', 'ATGGTGCACC',
'GGAGAAAGCA', 'CTTGATAATA', 'CTACGAATCC', 'AAGTGCATAT', 'ACAAGAGCAT', 'CC
TTAACTAG', 'GTTAGCGCTC', 'CCGTATACCG', 'CTCCGCATAG', 'AGTGTGGCT', 'ACCGT
CTACA', 'ATGGGGAATG', 'TCGTATAGC', 'CGGAACGCTG', 'TGCCAAATCT', 'CCGAGGTG
AT', 'TTACAAGTTC', 'TTGCTCAGAA', 'GGTCGCCCCA', 'ACGCTTCTCTA', 'CGTCATTTAA',
'TTATGGTGGC', 'ATAAAGGCTA', 'CTCGAGTGAC', 'TGAATTGCTC', 'TACCGGATGA']

```

Εικόνα 5.1

```

thanos@ubuntu: ~/Documents/bioinformatics
File Edit View Search Terminal Help
CACGCC', 'GCAAAATCAA', 'ATCAACATAT', 'TGGACCTCAT', 'GCAAGTCCCG', 'TCACACA
CTC', 'TAGTCGGAGG', 'ACTGCTGCGG', 'GCTTCTACCT', 'ACTATTCTC', 'CGACCTTGAA
', 'ATTGCAGTCA', 'AACACGAGGG', 'AGACTTGACG', 'TGGGTCCCCA', 'ATGGTGCACC',
'GGAGAAAAGCA', 'CTTGATAATA', 'CTACGAATCC', 'AACTGCATAT', 'CACAAGGCAC', 'CC
TTAACTAG', 'GTTAGCGCTC', 'CCGTATACCG', 'CTCCGCATAG', 'AGTGCTGGCT', 'ACCGT
CTACA', 'ATGGGGAATG', 'TCGTTATAGC', 'CGGAACGCTG', 'TGCCAAATCT', 'CCGAGGTG
AT', 'TTACAAGTTC', 'TTGCTCAGAA', 'GGTCGCCCGA', 'ACGCTTCCTA', 'CGTCATTTAA',
'TTATGGTGCG', 'ATAAAGGCTA', 'CTCGAGTGAC', 'TGAATTGCTC', 'TACCGGATGA', '
AGCGCGCAGA', 'AACATATTGT', 'CCAAGACCTC', 'TGCTAGCCAT', 'CCATCCCACA', 'GAG
GCGAGAC', 'ACAAACAGAA', 'GATCGTGCGG', 'CCAGTAAAGC', 'AAACCCTT']

Calculating combinations of w...

Found 3782 combinations of w.

Calculating scores of combinations...

Calculated 3782 scores.

Best combination of w:
AGACTTGACGGGAGAAAGCA

Highest Score:
9.0

Alignment based on Needleman-Wunsch:
ACGACGTCTG-CGGGAG-AATC-
|-|||-|-||-|||||-|X|-
A-GAC-T-TGACGGGAGAAAGCA

(venv) thanos@ubuntu:~/Documents/bioinformatics$ |

```

Εικόνα 5.2

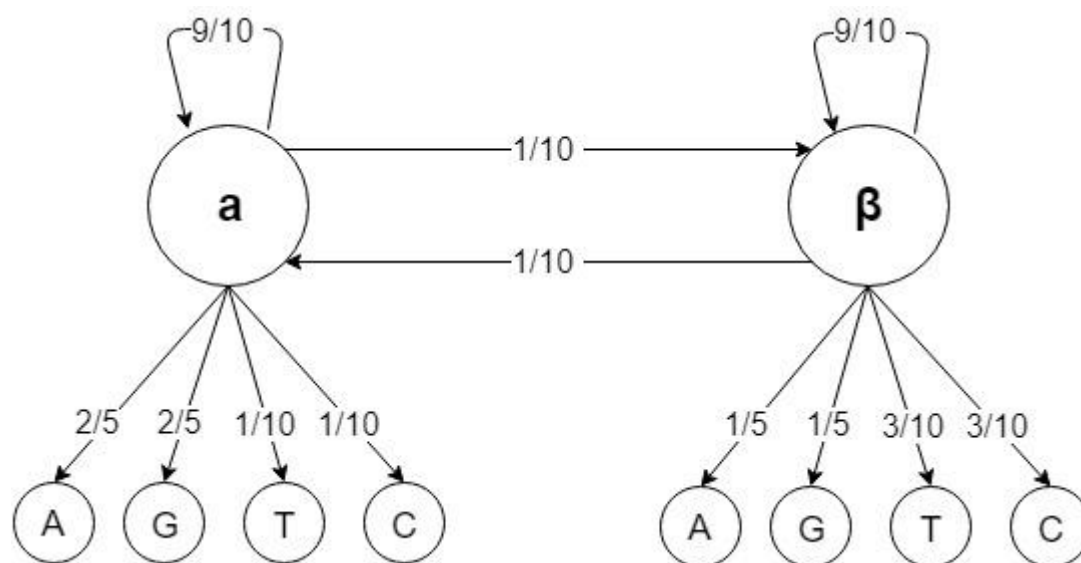
5.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- <https://www.rcsb.org/structure/1BBT>

6 Άσκηση 11.4

6.1 Εκφώνηση

Στο σχήμα 1 φαίνεται ένα HMM με δύο καταστάσεις α και β . Όταν το HMM βρίσκεται στην κατάσταση α , έχει μεγαλύτερη πιθανότητα να εκπέμψει πουρίνες (A και G). Όταν βρίσκεται στην κατάσταση β έχει μεγαλύτερη πιθανότητα να εκπέμψει πυριμιδίνες (C και T). Αποκωδικοποιήστε την πιο πιθανή ακολουθία των καταστάσεων (α/β) για την αλληλουχία **GGCT**. Χρησιμοποιήστε λογαριθμικές βαθμολογίες αντί για κανονικές βαθμολογίες πιθανοτήτων.



Σχήμα 1

6.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και η βιβλιοθήκη `numpy`.

6.3 Υλοποίηση

Για να αποκωδικοποιήσουμε την ακολουθία των καταστάσεων (α/β) για την αλληλουχία **GGCT** θα χρησιμοποιήσουμε τον αλγόριθμο του Viterbi και θα στηριχτούμε στις αρχές του δυναμικού προγραμματισμού.

Αρχικά αποκωδικοποιούμε τα δεδομένα που μας παρέχει το πρόβλημα και σχηματίζουμε τους απαραίτητους πίνακες, εικόνα 4.1.

από/προς	α	β
START	0.50	0.50
α	0.90	0.10
β	0.10	0.90

Πίνακας 1 – Πίνακας μεταβάσεων

εκπομπή/κατάσταση	α	β
A	0.40	0.20
G	0.40	0.20
T	0.10	0.30
C	0.10	0.30

Πίνακας 2 – Πίνακας εκπομπών

Όπως φαίνεται και στο σχήμα 1 το HMM αποτελείται από δυο καταστάσεις, την κατάσταση α και την κατάσταση β. Υπάρχουν πάρα πολλά μονοπάτια που οδηγούν στην ζητούμενη κατάσταση αλλά εμείς θα υπολογίζουμε αυτό με την μεγαλύτερη πιθανότητα.

Για τους υπολογισμούς μας, είναι αποδοτικότερο να χρησιμοποιήσουμε τον λογάριθμο των πιθανοτήτων. Έτσι, θα πρέπει να υπολογίσουμε το άθροισμα των πιθανοτήτων αντί για το γινόμενο τους.

Θεωρούμε ότι η πιθανότητα για την επιλογή αρχικής κατάστασης είναι $1/2$ για την μετάβαση στην κατάσταση α και $1/2$ για την μετάβαση στην κατάσταση β.

6.4 Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-11_4.py**.

Με βάση τα δεδομένα μας, η βέλτιστη αλληλουχία καταστάσεων για την αλληλουχία **GGCT** είναι η αααα.

Στην εικόνα 6.1 παρουσιάζεται το αποτέλεσμα της εκτέλεσης του προγράμματος. Εκτός από την βέλτιστη αλληλουχία καταστάσεων παρουσιάζονται και οι μέγιστες πιθανότητες για κάθε κόμβο της διαδρομής καταστάσεων που επιλέχθηκε.

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
In[2]: runfile('C:/Users/pioan/PycharmProjects/bioinformatics/exercise-11_4.py', wdir='C:/Users/p
The best path for the sequence GGCT is:
['a', 'a', 'a', 'a']

The best score for each node GGCT is:
[-2.321928094887362, -3.7958592832197744, -7.269790471552186, -10.743721659884597]

In[3]:
```

Εικόνα 6.1

6.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Αναγνώριση προτύπων
Σ. Θεοδωρίδης, Κ. Κουτρούμπας
- Σημειώσεις Αναγνώριση Προτύπων, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- https://en.wikipedia.org/wiki/Viterbi_algorithm

7 Άσκηση 11.6

7.1 Εκφώνηση

Θεωρήστε ένα διαφορετικό παιχνίδι στο οποίο κρουπιέρης δεν ρίχνει νόμισμα αλλά ζάρια με τρεις πλευρές που έχουν ετικέτες 1, 2, και 3. (Μην προσπαθήσετε να σκεφτείτε την εμφάνιση ενός τέτοιου ζαριού.) Ο κρουπιέρης έχει δύο στημένα ζάρια D1 και D2. Για κάθε ζάρι D, η πιθανότητα να προκύψει ο αριθμός είναι ίση με $1/2$, και η πιθανότητα για τα άλλα δύο αποτελέσματα είναι ίση με $1/4$. Σε κάθε γύρο, ο κρουπιέρης πρέπει να αποφασίσει αν (1) θα κρατήσει το ίδιο ζάρι, (2) θα αλλάξει ζάρι, ή (3) θα σταματήσει το παιχνίδι. Επιλέγει το (1) με πιθανότητα $1/2$ και τα (2) και (3) με πιθανότητα $1/4$. Στην αρχή, ο κρουπιέρης επιλέγει ένα από τα δύο ζάρια με την ίδια πιθανότητα.

- Διατυπώστε ένα HMM για την παραπάνω κατάσταση. Προσδιορίστε το αλφάβητο, τις καταστάσεις, τις πιθανότητες μεταβολής κατάστασης, και τις πιθανότητες εκπομπής. Συμπεριλάβετε μια αρχική κατάσταση start, και υποθέστε ότι το HMM ξεκινάει στην κατάσταση start με πιθανότητα 1. Συμπεριλάβετε επίσης και μια τελική κατάσταση end.
- Ας υποθέσουμε ότι παρατηρείτε την εξής ακολουθία από ρίψεις ζαριών : **112122**. Βρείτε μια ακολουθία καταστάσεων που εξηγεί καλύτερα την ακολουθία των ρίψεων. Ποια είναι η πιθανότητα της συγκεκριμένης ακολουθίας; Βρείτε την απάντηση συμπληρώνοντας τον πίνακα Viterbi. Συμπεριλάβετε βέλη οπισθοδρόμησης στα κελιά έτσι ώστε να είστε σε θέση να ανιχνεύσετε αντίστροφα την ακολουθία των καταστάσεων. Μερικά από τα παρακάτω δεδομένα ίσως φανούν χρήσιμα :

$$\begin{aligned} \log_2(0) &= -\infty \\ \log_2(1/4) &= -2 \\ \log_2(1/2) &= -1 \\ \log_2(1) &= 0 \end{aligned}$$

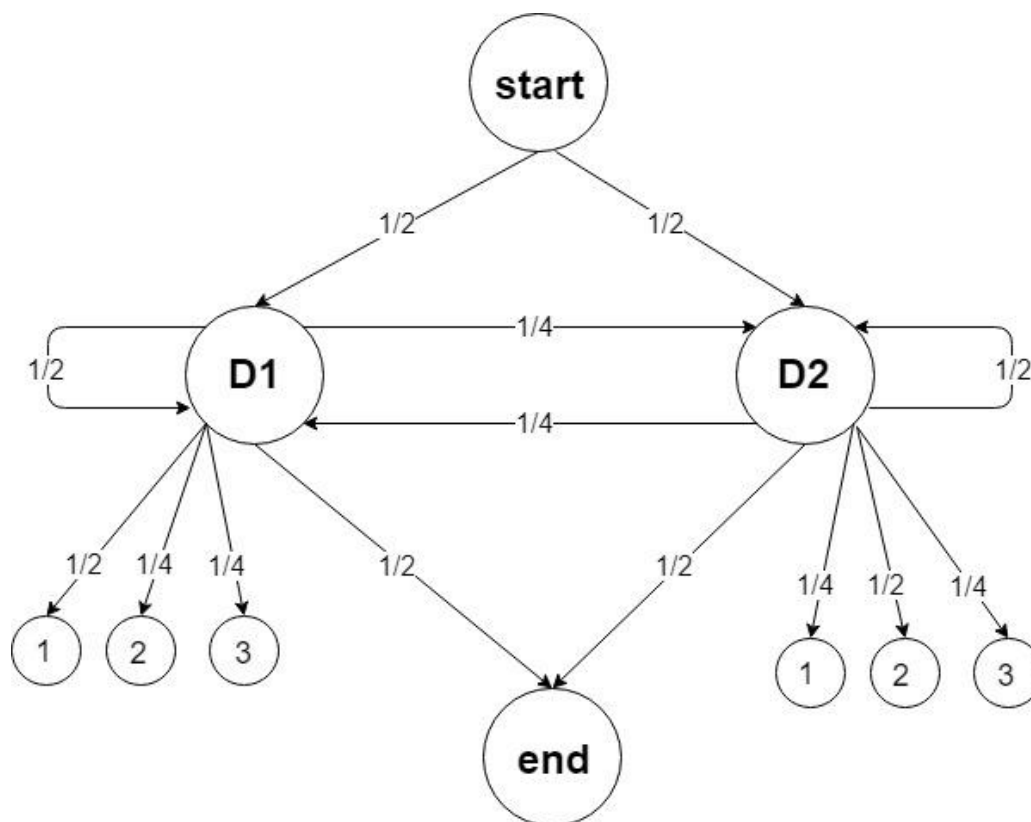
- Υπάρχουν στην πραγματικότητα δύο βέλτιστες ακολουθίες καταστάσεων για τη συγκεκριμένη ακολουθία ρίψεων των ζαριών. Ποια είναι η άλλη ακολουθία καταστάσεων;

7.2 Σχεδιασμός

Για την επίλυση της άσκησης χρησιμοποιούμε τη Python 3.7 και η βιβλιοθήκη numpy.

7.3 Υλοποίηση

Στο σχήμα 2 παρουσιάζεται το HMM που περιγράφεται στη εκφώνηση της άσκησης και στους πίνακες 3 και 4 τα αντίστοιχα δεδομένα.



Σχήμα 2

από/προς	START	D1	D2	END
START	0.00	0.50	0.50	0.00
D1	0.00	0.50	0.25	0.25
D2	0.00	0.25	0.50	0.25
END	0.00	0.00	0.00	0.00

Πίνακας 3 – Πίνακας μεταβάσεων

κατάσταση/εκπομπή	1	2	3
START	0.00	0.00	0.00
D1	0.5	0.25	0.25
D2	0.25	0.50	0.25
END	0.00	0.00	0.00

Πίνακας 4 – Πίνακας εκπομπών

Και αυτή η άσκηση θα υλοποιηθεί με τη μέθοδο του δυναμικού προγραμματισμού και του αλγορίθμου Viterbi.

Για τους υπολογισμούς μας, είναι αποδοτικότερο να χρησιμοποιήσουμε τον λογάριθμο των πιθανοτήτων. Έτσι, θα πρέπει να υπολογίσουμε το άθροισμα των πιθανοτήτων αντί για το γινόμενο τους.

Δεδομένου ότι δεν αρκεί να βρούμε μόνο την βέλτιστη ακολουθία θα πρέπει να υπολογίσουμε τις πιθανότητες για όλες τις αλλαγές καταστάσεων και μεταβάσεων. Σε κάθε κατάσταση κρατάμε την μέγιστη πιθανότητα.

Αφού τις υπολογίσουμε όλες, με την μέθοδο του backtracking, θα βρούμε τις δυο βέλτιστες ακολουθίες καταστάσεων για την ακολουθία ρίψεων ζαριών : **112122**.

7.4 Αποτέλεσμα

Αφού εισέλθουμε στο φάκελο bioinformatics εκτελούμε το αρχείο της άσκησης με την εντολή **python exercise-11_6.py**.

Με βάση τα δεδομένα μας, οι δυο βέλτιστες ακολουθίες που προκύπτουν, συμπεριλαμβανομένων των καταστάσεων START και END είναι οι:

1. **START-D1-D1-D1-D1-D2-D2-END**
2. **START-D1-D1-D2-D2-D2-D2-END**

Ο Viterbi πίνακας που περιλαμβάνει τις μέγιστες πιθανότητες προέκυψαν είναι:

D1	-2	-4	-7	-9	-12	-15
D2	-3	-6	-7	-10	-12	-14

Στην εικόνα 7.1 παρουσιάζεται το αποτέλεσμα της εκτέλεσης του προγράμματος. Εκτός από τις δυο βέλτιστες ακολουθίες καταστάσεων παρουσιάζονται και οι μέγιστες πιθανότητες για κάθε κόμβο καθώς επίσης και το βέλτιστο Viterbi score το οποίο συμπεριλαμβανομένων του αρχικού και τελικού κόμβου είναι -16.

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit  
In[2]: runfile('C:/Users/pioan/PycharmProjects/bioinformatics/exercise-11_6.py',  
The viterbi matrix is:  
[[ -2  -4  -7  -9 -12 -15]  
 [ -3  -6  -7 -10 -12 -14]]  
  
The best sequences are:  
START- D1-D1-D1-D1-D2-D2 -END  
START- D1-D1-D2-D2-D2-D2 -END  
  
The best score is: -16.0  
  
In[3]:
```

Εικόνα 7.1

7.5 Αναφορές

- Εισαγωγή στους αλγόριθμους Βιοπληροφορικής
Neil. C Jones, Pavel A. Pevzner
- Αναγνώριση προτύπων
Σ. Θεοδωρίδης, Κ. Κουτρούμπας
- Σημειώσεις Αναγνώριση Προτύπων, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- Σημειώσεις Βιοπληροφορικής, Πανεπιστήμιο Πειραιά
Πικράκης Άγγελος, PhD
- https://en.wikipedia.org/wiki/Viterbi_algorithm

8 Βοηθητικά αρχεία

Η κλάση `Data` που βρίσκεται στο αρχείο **`data.py`** περιέχει την στατική μέθοδο `load_data(filename)`. Αυτή η μέθοδος δέχεται ως όρισμα το όνομα του αρχείου από το οποίο θέλουμε να φορτώσουμε κάποια ακολουθία.

Επίσης στο φάκελο **`data`** βρίσκονται τα αρχεία που περιέχουν τις ακολουθίες που χρησιμοποιήθηκαν για τις ασκήσεις.