|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Εργασία Βιοπληροφορικής  εαρινό εξάμηνο ακαδημαϊκού έτους 2023-2024 |
| |  |  |  | | --- | --- | --- | | ΜΑΤΙΝΑ ΠΑΠΑΔΑΚΟΥ - Π21127 |  | ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΠΟΥΛΟΣ – Π20206 | |

ΠΕΡΙΕΧΟΜΕΝΑ

[**ΘΕΜΑ I 2**](#thema1)

[(a)2](#thema1a)

[(b)2](#thema1b)

[(c)3](#thema1c)

[**ΘΕΜΑ ii 4**](#thema2)

[**ΘΕΜΑ iii 5**](#thema3)

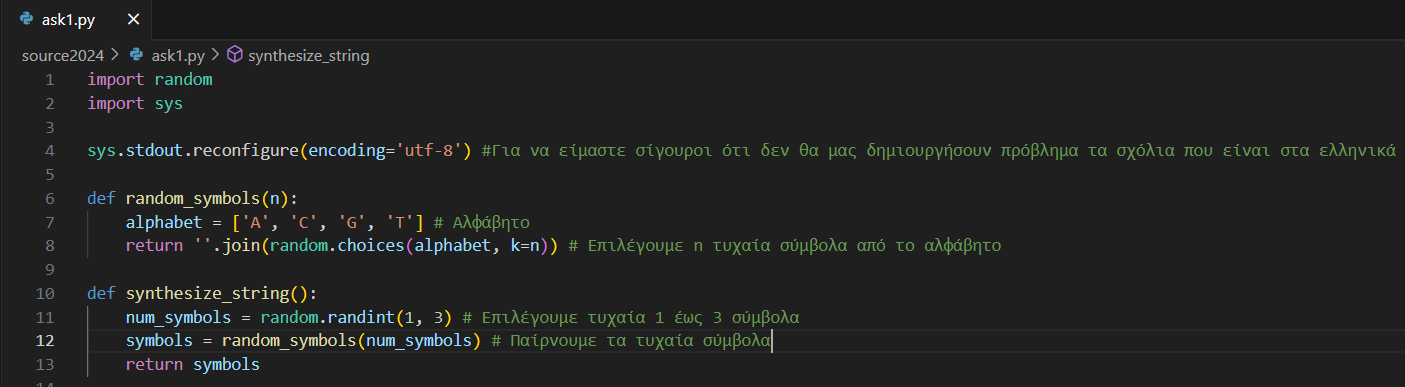
[**Απεικόνιση ενός HMM Profile με πολλαπλή στοίχιση 5**](#thema3)

**ΘΕΜΑ i**

**(a)**

Αφού καλέσουμε τις βιβλιοθήκες random & sys , θα χρειαστούμε 2 μεθόδους για το πρώτο υπο ερώτημα.

Η synthesize\_string() επιλέγει έναν αριθμό από το ένα έως το τρία, καλεί την random\_symbols(n) και αυτή επιστρέφει ενα string με n σύμβολα που διαλέγει τυχαία από την λίστα alphabet .



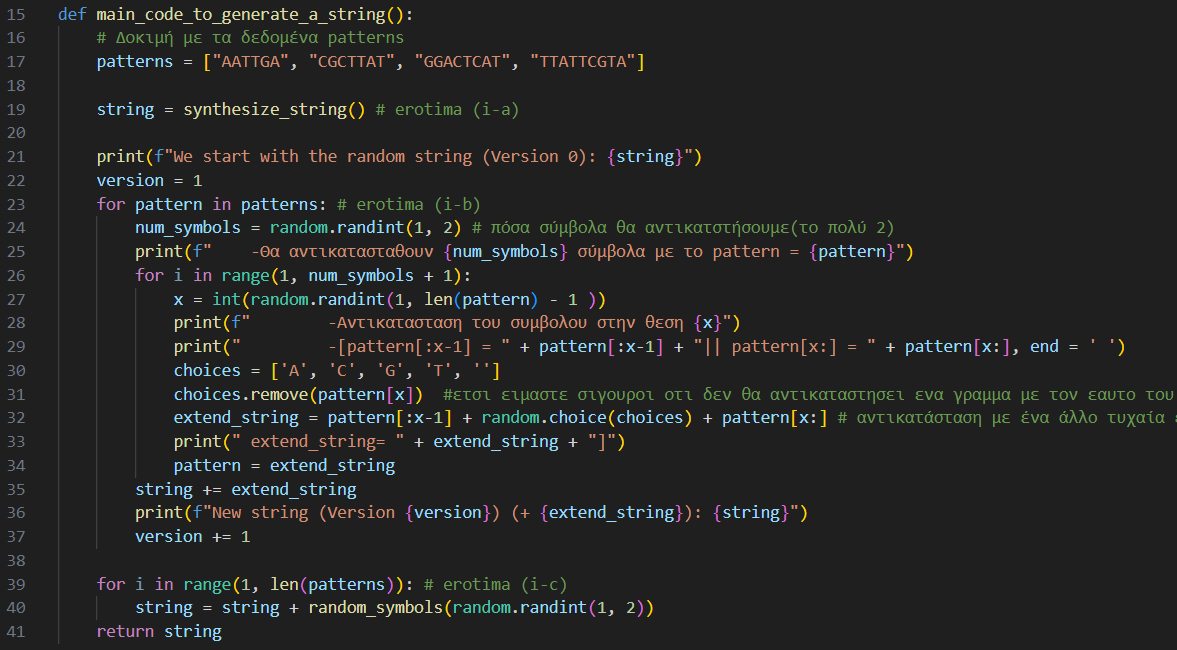
Τα τυχαία σύμβολα θα καταχωρηθούν στην συμβολοσειρά string(μέσα στην main\_code\_to\_generate\_a\_string()) :



**(b)**

Για κάθε ένα από τα 4 patterns θα εκτελεστεί το block κώδικα μέσα στην for. Οι εντολές print χρησιμοποιήθηκαν για την καλύτερη κατανόηση του αναγνώστη. Η extend\_string σπάει το pattern στα 3 μέρη ως εξής:  
Εστω το pattern = ΑΑΤΤGA, αν επιλέξουμε τυχαία να αντικαταστησουμε το γράμμα στην θεση 5, το G, τότε pattern[:x-1] = AATT και pattern[x:] = A και με την βοήθεια της random.choice([‘A’, ‘C’,’G’, ‘T’ , ‘ ‘]) επιλέγεται τυχαία ένα γράμμα ή το κενο . Αρα AATT + A + A = AATTAA.

Ο κώδικας που το επιτυγχάνει αυτο είναι:

****

Ξεκινάμε αρχικοποιώντας την συμβολοσειρά string με την διαδικασία του [ερωτήματος (1)](#thema1)

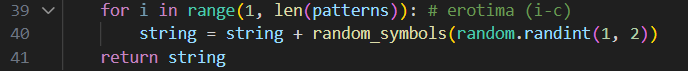
Στην συνέχεια για κάθε pattern από την λίστα patterns, επιλέγονται τυχαία πόσα σύμβολα (1ή2) απο το pattern θα αντικατασταθούν. Για κάθε επιλεγμένο σύμβολο, ορίζεται μια τυχαία θέση στο pattern, όπου θα γίνει η αντικατάσταση.

Και έπειτα το σύμβολο στην συγκεκριμένη θέση αντικαθίσταται με ένα άλλο τυχαία επιλεγμένο σύμβολο απο το αλφάβητο (A, C, G, T), είτε με κενό, ώστε να προκύψει μια τροποποιημένη έκδοση του pattern. Τελικά η νέα αυτή εκδοχή προστίθεται στην αρχική συμβολοσειρά string

Το δεύτερο for loop που φαίνεται στην παραπάνω στιγμιότυπο(γραμμή 39) εκτελεί το [ερώτημα (c)](#thema1c) που θα αναλυθεί παρακάτω.

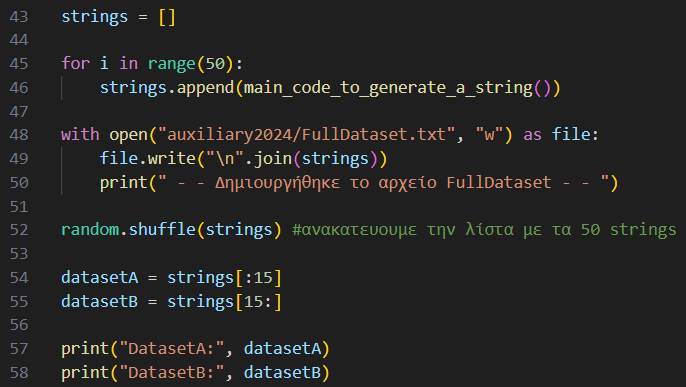
**(c)**

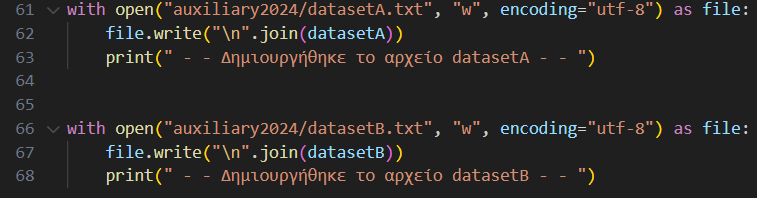
Τελειώνοντας το **[ΘΕΜΑ I](#thema1)**, θα χρησιμοποιήσουμε πάλι την random\_symbols(n), για να μας επιστρέψει ενα string με ένα εώς δύο τυχαία σύμβολα.



Μένει να αναφέρουμε το πώς δημιουργήσαμε 50 συμβολοσειρές και τις καταχωρησαμε τυχαία σε 2 σύνολα datasetA[15] και datasetB[35]:

Καλούμε με επαναληπτική διαδικασία(for loop) την main\_code\_to\_generate\_a\_string(), οπότε έχουμε την λίστα string με τις 50 συμβολοσειρές. Επειτα ανοίγουμε(ή δημιουργούμε) το αρχείο κειμένου FullDataset και καταγράφουμε τα 50 strings με την βοήθεια της μεθόδου join(). Στη συνέχεια ανακατεύουμε την λίστα και αποθηκεύουμε στα datasetA & datasetB 15 και 35 συμβολοσειρές αντίστοιχα απο την λίστα strings.

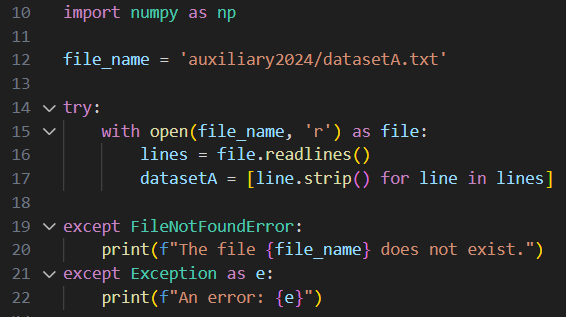


Και εδώ αποθηκεύουμε στα αρχεία κειμένου datasetA, datasetB τις αντίστοιχες συμβολοσειρες 

**ΘΕΜΑ ii**

* Open DatasetA:

Αρχικά θα πρέπει να διαβάσουμε τα δεδομένα από το αρχείο κειμένου DatasetA με την βοήθεια της εντολής with open() as file, που θα εξασφαλίσει ότι το αρχείο θα κλείσει σωστά μετά την ανάγνωση. Συγκεκριμένα η μέθοδος readlines() θα διαβάσει όλες τις γραμμές του αρχείου καταχωρώντας το περιεχόμενο στην λίστα lines. Ενώ η μέθοδος strip() θα αφαιρέσει τυχόν περιττά κενά. Αν το αρχείο δεν βρεθεί ή αν προκύψει κάποιο άλλο σφάλμα όταν προσπαθήσουμε να διαβάσουμε το αρχείο, θα εκτυπωθεί ένα αντίστοιχο μύνημα σφάλματος.



* global\_alignment(A, B, alpha = 2):

Η μέθοδος αυτή στοχεύει να ευθυγραμμίσει δύο συμβολοσειρές σε όλο το μήκος τους, ακόμα και αν αυτό σημαίνει την εισαγωγή κενών για την βέλτιστη στοίχιση.

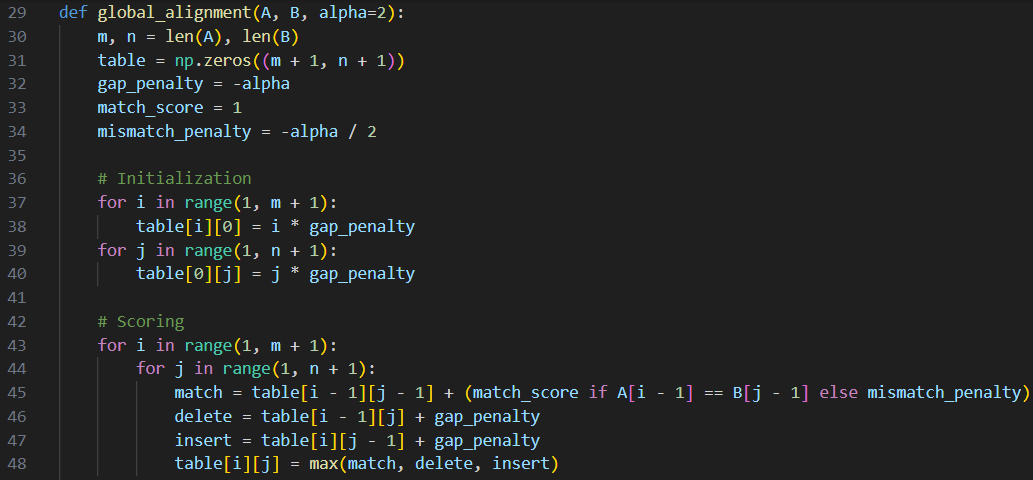
Αναλυτικά:

* Δημιουργούμε έναν πίνακα table για να αποθηκευσουμε τις βέλτιστες τιμές κάθε στοίχισης για κάθε υποπρόβλημα. Τον αρχικοποιούμε με μηδενικά.
* Alpha = 2 γιατι ΑΜ : 21127, 20206.
* Για να υπολογίσουμε το σκορ έχουμε:

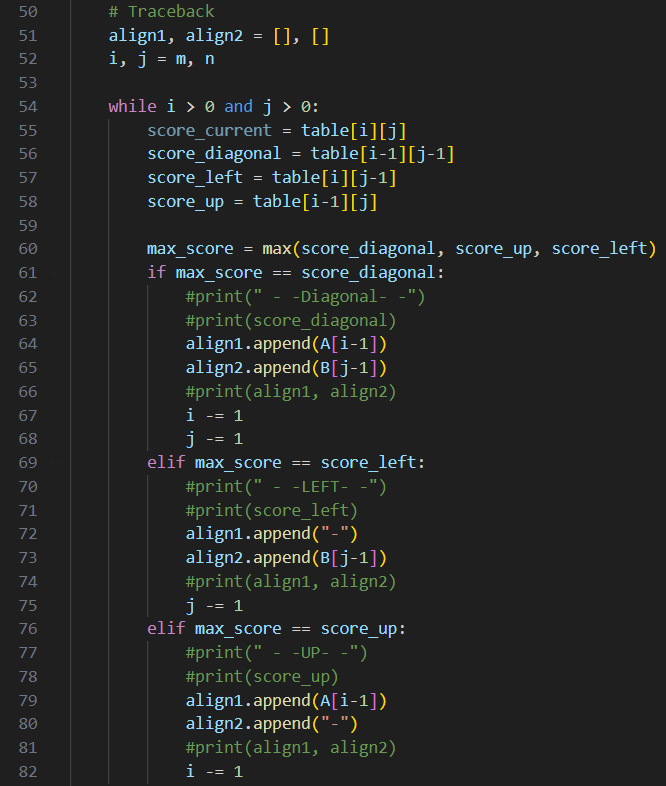
Για κάθε ζέυγος χαρακτήρων (i, j ) των δύο συμβολοσειρών, υπολογίζονται τρεις πιθανές βαθμολογίες

* match: Προστίθεται στη βαθμολογία της διαγώνιας κίνησης. Προσθέτει 1 εάν οι χαρακτήρες είναι ίδιοι, διαφορετικά αφαιρεί alpha/2.
* delete: Προστίθεται στη βαθμολογία για κίνηση προς τα πάνω
* insert: Προστίθεται στη βαθμολογία για κίνηση προς τα αριστερά

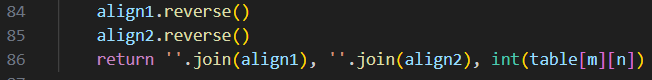
Και η μέγιστη από αυτές τις τρεις τιμές αποθηκεύεται στο κελί του πίνακα table[i][j]



* Ξεκινώντας από το τελευταίο κελί του πίνακα, ακολουθείται η διαδρομή με τη μέγιστη τιμή προς τα πίσω για να βρούμε την βέλτιστη στοίχιση.
* Αν η διαδρομή ακολουθεί τη διαγώνια κίνηση, τότε οι χαρακτήρες ευθυγραμμίζονται. Ενώ εάν ακολουθείται η κίνηση προς τα πάνω ή αριστερά, εισάγεται κανό σε μια από τις δυο συμβολοσειρές.

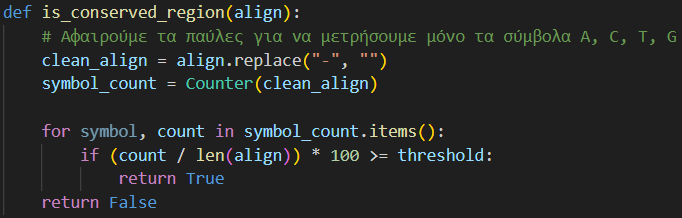
****

* Και τέλος επιστρέφονται οι δύο ευθυγραμμισμένες συμβολοσειρές μαζί με την συνολική βαθμολογία της στοίχισης.

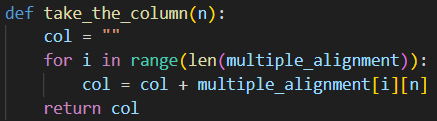
****

**ΘΕΜΑ iii**

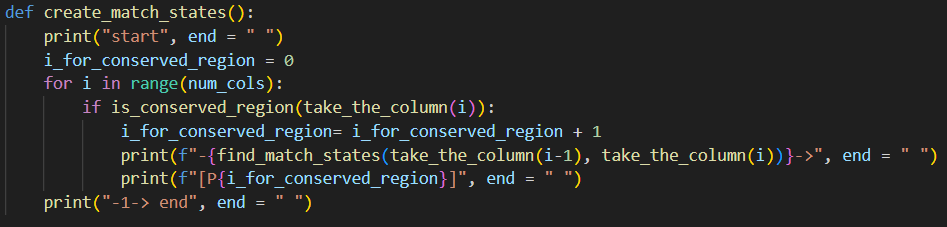
* Η πρώτη συνάρτηση που συναντάμε είναι η **is\_conserved\_region(align)**. Μετράει το πλήθος κάθε συμβόλου που υπάρχει σε κάθε στήλη της πολλαπλής στοίχισης και αν υπάρχει κάποιο σύμβολο που εμφανίζεται σε ένα συγκεκριμένο ποσοστό(π.χ. Threshold = 70%) επιστρέφει **True**, αλλιώς επιστρέφει **False**.

****

* Η συνάρτηση **take\_the\_column(n)** φτιάχνει μια συμβολοσειρά, την **col** οπου την επιστρέφει στο τέλος. Αυτή η συμβολοσειρά είναι μια στήλη του πίνακα **multiple\_alignment**, ανάλογα με το n >=0 && n <= cols. Η κύρια χρήση της, είναι να ετοιμάσει την συμβολοσειρά για να την αξιοποιήσει η παραπάνω συναρτηση και να ελέγξει αν ειναι **conserved\_region**.

****

* Η συνάρτηση **create\_match\_states()**  βρίσκει ποιες καταστάσεις είναι match.

****

Ενα παράδειγμα εκτέλεσης της **create\_match\_states()**  είναι το παρακάτω:



**Απεικόνιση ενός HMM Profile με πολλαπλή στοίχιση:**

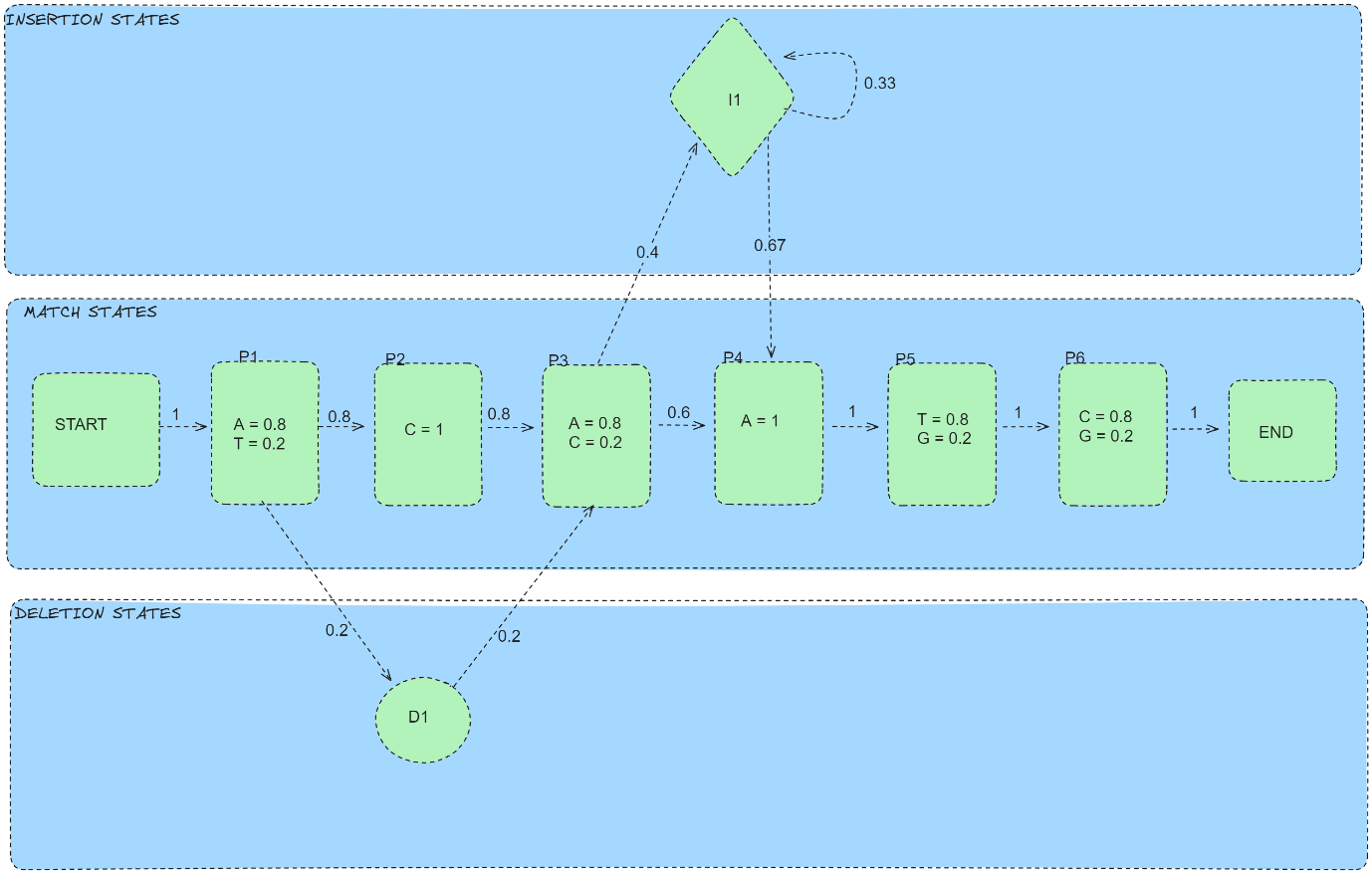
    ACA---ATG

    TCAACTATC

    ACAC--AGC

    ACA---ATC

    A-C---ATC

****