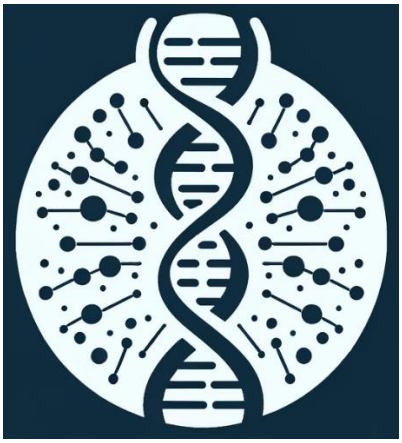


234124 – מבוא לתכנות מערכות
תרגיל בית מספר 5
סמסטר חורף 23/24 (אודיסיאה)

תאריך פרסום: 28/01/2024

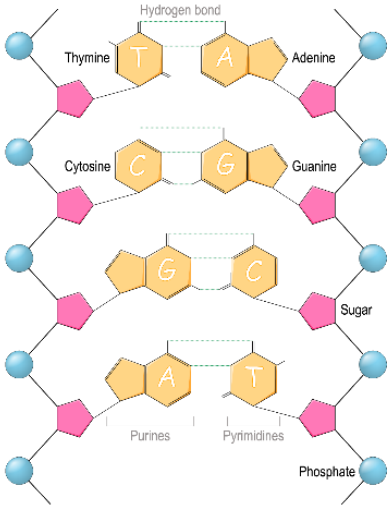
תאריך הגשה: 11/02/2024 בשעה 23:59

1. הערות כלליות

- **תרגיל זה מהווה 4% מהציון הסופי**
- התרגיל להגשה **בזוגות** בלבד
- מענה לשאלות בנוגע לתרגיל יינתן אך ורק בפורום התרגיל בפיאצה או בסדנאות. לפני פרסום שאלה בפיאצה אנא בדקו אם כבר נענתה.
- קראו את התרגיל עד סופו לפני שאתם מתחילים לממש. חובה להתעדכן בעמוד הפיאצה של התרגיל, הכתוב שם מחייב.
- העתקות קוד בין סטודנטים ובפרט גם העתקות מסמסטרים קודמים תטופלנה. עם זאת – מומלץ ומבורך להתייעץ עם חברים על ארכיטקטורת המימוש.
- קבצי התרגיל נמצאים ב-GitHub Repository הבא:
https://github.com/cs234124-odyssey/ex5.git
- המסמך נכתב בלשון זכר מטעמי נוחות בלבד ומיועד לשני המינים.
- מטרת תרגיל זה היא היכרות עם תכנות ב-Python.

2. מחלקת DNA

2.1. רקע



ה-DNA היא המולקולה המאחסנת את הקוד הגנטי בכל היצורים החיים (כמעט) והיא מורכבת מרצפים של "בסיסים" (נקראים גם – נוקלאוטידים). מכיוון שבמולקולת DNA סטנדרטית קיימים רק 4 בסיסים, ניתן לייצג רצפי DNA ארוכים כמחרוזות המורכבות מארבעת התווים: A, T, C, G.

בנוסף, מאפיין עיקרי של מולקולת ה-DNA הוא העובדה שהיא מורכבת מ-2 "גדילים" משלימים – ניתן לחשוב על מולקולת ה-DNA כמעין סולם עם שלבים כאשר כל שלב מורכב מ-2 חלקים (אחד מכל צד של הסולם) שמתחברים יחד. על כן, מול כל בסיס A יופיע תמיד בסיס T ומול כל בסיס C יופיע תמיד בסיס G.

2.2. מחלקת DNASequence

ממשו את המחלקה **'DNASequence'** שמכילה את המתודות הבאות:

1. **__init__(self, nucleotides)** – בנאי שמקבל רצף של בסיסי DNA כרשימה (list) של תווים ומאתחל עצם שמכיל את הרצף.
2. **get_sequence(self)** – מתודה שמחזירה את רצף ה-DNA השמור בעצם כרשימה של תווים.
3. **get_length(self)** – מתודה שמחזירה את אורך רצף ה-DNA השמור בעצם.
4. **get_complement(self)** – מתודה שמחזירה את רצף ה-DNA **המשלים** לרצף השמור בעצם (מחליפה A ב-T, C ב-G, T ב-A ו-G ב-C) כרשימה של תווים.
5. **get_nucleotide(self, index)** – מתודה שמקבלת index ומחזירה את הבסיס במיקום ה-index.
6. **find_alignment(self, seq)** – מתודה שמקבלת רצף של בסיסים כמחרוזת (string) ומחזירה את ה-index של תחילת המופע הראשון של רצף הבסיסים ב-DNA המלא (לדוגמה: המופע הראשון של "ATG" ב-"ATATATGCATG" הוא ב-index 4).
7. **replace_sequence(self, seq)** – מתודה שמקבלת רצף DNA חדש (כרשימה של תווים) על מנת להחליף את הרצף הקיים.

הערות:

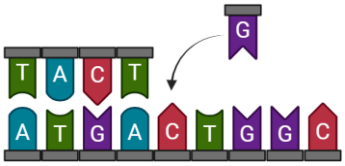
- מבין המתודות 4, 6 ו-7, יש לממש לפחות 2 מתודות כ-one-liners.
- מומלץ לממש את פונקציית העזר **'Complement'** שמקבלת תו המייצג בסיס ב-DNA ומחזירה את הבסיס המשלים לו.
- לא חובה להשתמש בכל המתודות בסעיפים הבאים.

3. מחלקות אנזימים

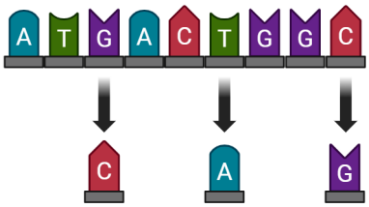
3.1. רקע

בטבע, אנזימים הם חלבונים שגורמים לתגובות כימיות ביצורים חיים. בתרגיל זה, הריאקציות ישפיעו על מבנה ה-DNA (שלב 2.2).

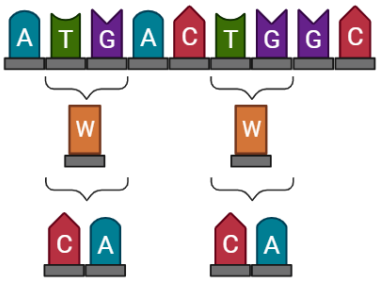
האנזים ***DNA Polymerase*** הוא אנזים אמיתי שבהינתן רצף DNA, מייצר את הגדיל המשלים לו:



האנזים ***Mutase*** הוא אנזים אמיתי גם כן אבל אנחנו נתעלם מהפעולה האמיתית שלו. בתרגיל זה, האנזים ***Mutase*** יגרום למוטציות (שינויים ב-DNA) בתדירות קבועה – ה-***Mutase*** ישנה כל בסיס n-י ב-DNA לבסיס המשלים שלו. למשל, עבור n=3:



הקומפלקס ***CRISPR/Cas9*** הוא שילוב של האנזימים ***Cas9*** ו-***CRISPR*** שיחד יכולים לזהות רצפים ספציפיים ב-DNA ולערוך אותם. בתרגיל זה, האנזים ***CRISPR*** יחליף את רצף המטרה שלו עם הבסיס W והקומפלקס ***CRISPR/Cas9*** יחליף את הנוקלאוטיד W ברצף חדש. למשל עבור רצף המטרה TG והרצף החדש CA:



<u>הערה:</u> שימו לב שהבסיס המשלים ל-W הוא הבסיס M. במקרה והקומפלקס ***CRISPR/Cas9*** נתקל בבסיס M, הוא מחליף אותו ברצף המשלים לרצף החדש.

3.2. מחלקת Enzyme והמחלקות היורשות ממנה

3.2.1. המחלקה Enzyme

ממשו את המחלקה **'Enzyme'** שמכילה את המתודות הבאות:

1. **__init__(self)** – בנאי, למחלקה **'Enzyme'** אין שדות.
2. **process(self, dna_sequence)** – מתודה שמקבלת עצם מטיפוס **'DNASequence'** ומבצעת עליו פעולה (בהתאם לסוג האנזים, האנזים הבסיסי לא מבצע אף פעולה).

3.2.2. המחלקות היורשות מ-Enzyme

עדכנו את הבנאים של המחלקות הבאות:

- עבור המחלקה '**Mutase**' – הבנאי צריך לקבל את המספר **'freq'** שמציין את תדירות המוטציות שהאנזים יגרום.
- עבור המחלקה **'CRISPR'** - הבנאי צריך לקבל את המחרוזת **'seq'** שמציינת את הרצף ש-***CRISPR*** יחליף ב-DNA.

בנוסף, ממשו את המתודה process עבור המחלקות הבאות:

- עבור המחלקה **'Polymerase'** – המתודה מדמה את הפעולה של אנזים ה-***Polymerase*** ומחזירה את הרצף המשלים ל-DNA שהתקבל.
- עבור המחלקה '**Mutase**' - המתודה מדמה את הפעולה של אנזים ה-***Mutase*** ומחליפה כל בסיס n-י (כאשר n הוא הפרמטר **'freq'** שהתקבל בבנאי) ב-DNA שהתקבל, בבסיס המשלים לו.
- עבור המחלקה **'CRISPR'** - המתודה מדמה את הפעולה של אנזים ה-***CRISPR*** ומחליפה את כל המופעים של רצף המטרה (**'seq'** שהתקבל בבנאי) ב-DNA שהתקבל, בבסיס W.

3.2.3. המחלקה CRISPR_Cas9 היורשת מ-CRISPR

עדכנו את הבנאי של המחלקה **'CRISPR_Cas9'** כך שיקלוט את המחרוזת **'new_seq'** המציינת את הרצף שאליו יחליף הקומפלקס את בסיסי ה-W.

בנוסף, ממשו את המתודה process למחלקה **'CRISPR_Cas9'** כך שתבצע את הפעולות הבאות:

- המתודה תקרא למימוש של process שמופיע במחלקה **'CRISPR'** (באמצעות super) על מנת להחליף את כל המופעים של **'seq'** בבסיס W.
- המתודה תחליף את כל המופעים של הבסיס W ברצף **'new_seq'**.
- המתודה תחליף את כל המופעים של הבסיס M ברצף המשלים ל-**'new_seq'.**

4. מערכת לביצוע פרוטוקול ניסוי

בחלק זה, נממש מערכת שתקרא אנזימים ורצפי DNA מתוך קבצי טקסט ותבצע פעולות על פי פרוטוקול ניסוי.

ממשו את הפונקציה **'processData(dir_path)'** שמבצעת את הפעולות הבאות:

1. הפונקציה קוראת את הקובץ **'DNA.json'** שנמצא בתיקייה **'dir_path'**. הקובץ מכיל שמות של רצפי DNA ואת הרצפים עצמם.
   *חובה להשתמש במחלקה **'DNASequence'** מחלק 2.2 על מנת לאחסן את הרצפים.
2. הפונקציה קוראת את הקובץ **'protocol.txt'** שנמצא בתיקייה **'dir_path'**. כל שורה בקובץ מכילה **שם** של רצף DNA ושם של אנזים (ואת הארגומנטים לאנזים אם יש צורך) באופן הבא:

DNA1 Polymerase
DNA2 Mutase [freq]
DNA3 CRISPR [seq]
DNA4 CRISPR/Cas9 [seq] [new_seq]

הערה: ניתן להשתמש בפונקציה ()split בחלק זה – נסו אותה!

3. עבור כל שורה בקובץ **'protocol.txt'**, הפונקציה תבצע את הפעולות הבאות:
   a. תיצור עצם מהמחלקה הרלוונטית שיורשת ממחלקת **'Enzyme'** בהתאם לשם האנזים שמופיע בקובץ.
   b. תפעיל את מתודת ה-process של האנזים על רצף ה-DNA ששמו מופיע באותה השורה – יש להחליף את ה-DNA הקיים ב-DNA החדש (קחו בחשבון שאותו DNA יכול להופיע מספר פעמים בקובץ והשינויים בו מצטברים).
4. הפונקציה תיצור קובץ ששמו **'ModifiedDNA.json'** בתוך התיקייה **'dir_path'**. הקובץ יהיה זהה במבנהו ל-**'DNA.json'** אך יכיל את הרצפים המעודכנים.

5. הרצת הקוד

כתבו פונקציית main שתקבל path לתיקייה כארגומנט ותריץ את הפונקציה **'processData(dir_path)'**.
שימו לב שפעולה זו תתבצע רק אם קובץ ה-Python שבו ה-main נמצאת, מורץ ישירות.

6. בונוס

חלק מרצפי ה-DNA מהטסטים שנמצאים ב-git הם למעשה הודעות מוצפנות! האם תוכלו לפענח אותן?

רמז: TAGC=1032="N"

7. הערות

- בכל התרגיל אין להשתמש במספרי קסם למעט 0/1.
- ניתן להניח כי הקלט תקין בכל התרגיל.
- וודאו כי אתם מריצים פייתון גרסה 3.6. שימו לב כי גרסה זו אינה גרסת ברירת המחדל על השרת. כדי להריץ פייתון 3.6 השתמשו בפקודה python3.
- פתרון התרגיל צריך לעבוד בכל מערכת הפעלה.
- מסופקים לכם עם התרגיל מספר טסטים אשר נועדו לבדוק בכלליות את התוכנית שלכם.
  - אל תסמכו על טסטים אלו! כתבו טסטים ובדקו את התוכנית שלכם.

8. הגשה

את ההגשה יש לבצע דרך את המודל של הקורס. הקפידו על הדברים הבאים:

- יש להגיש את קבצי הקוד מכווצים לקובץ zip (לא פורמט אחר).
- אין להגיש קבצים נוספים מלבד קובץ אחד בשם ex5.py בתוך ה-zip.
- ניתן להגיש את התרגיל מספר פעמים, רק ההגשה האחרונה נחשבת.
- על מנת לבטח את עצמכם נגד תקלות בהגשה האוטומטית, שימרו עותק של התרגיל בענן לפני ההגשה ואל תשנו אותו אחריה (שינוי של הקובץ יגרום לשינוי חתימת הזמן של העדכון האחרון).
  - כל אמצעי אחר לא יחשב הוכחה לקיום הקוד לפני ההגשה.

בהצלחה!