פיתוח תרופות בעידן המידע

תרגיל 2

מועד הגשה: 30.11.2021

- **תרגיל הבית מיועד להגשה עצמאית. כל העתקה שתימצא תגרור פסילה של התרגיל כולו, גם עבור הסטודנט המעתיק, וגם עבור הסטודנט שהתרגיל הועתק ממנו. אנא הימנעו מאי נעימות.**
- **כל תכנית צריכה להכיל תיעוד מסודר של מטרת התכנית, ושל פקודות חשובות בגוף התוכנית.**
- **יש לרשום בראש כל קובץ פייתון הערה עם שם הסטודנט ות"ז.**
- **אין להשתמש בחומר שטרם נלמד. מתוך הנושאים שנלמדו, מותר להיעזר בכל תיעוד רלוונטי.**
- **על כל ההערות ותיעוד התכנית להכתב בשפה האנגלית בלבד!**
    - מומלץ לתעד בפורמט הרלוונטי לפייתון, כולל שימוש ב-docstrings
- **יש להקפיד על כתיבת קוד תקין ונכון בהתאם לכללים שנלמדו. קוד שאינו תואם את העקרונות שנלמדו יקבל ניקוד חלקי בלבד.**
    - יש לתת שמות משמעותיים למשתנים.
    - ניתן להעזר בפונקציות משנה כרצונכם, בהתאם לעקרונות תכנות נכון ומודולרי.
- **יש להקפיד לעבוד על פי הפורמט המבוקש, רווחים מיותרים ופיסוק שגוי יגררו הורדת ניקוד.**

התרגיל יוגש בקובץ יחיד בשם exercise2_id.py דרך תיבת ההגשה הרלוונטית במודל. ניתן להגיש מספר פעמים עד למועד ההגשה הסופי. הגרסה האחרונה היא שתיבדק.

- לדוגמה exercise2_123456789.py

על התרגיל להיות מסוגל לרוץ באופן הבא:

```
python exercise1_partX_id.py argument1 argument2 …
```

- כדי לקבל משתנים לתוכנית, ניתן להשתמש בתחביר המפורט בקישור הבא:

```
import sys
# print first argument
print(sys.argv[1])
```

הסבר כללי:

- התרגיל בנוי כך שראשית אנו משדרגים פונקציות מתרגיל קודם לעבודה מתאימה עבורנו, ולאחר מכן בונים על גביהן קומה נוספת. לפיכך, מומלץ לקרוא את כל התרגיל בטרם תחילת העבודה כדי לאפשר הבנה מיטבית של הדרוש, תכנון קוד נכון ומניעת עבודה מיותרת.
- על בדיקות תקינות להיות רלוונטיות לקלט – באם מדובר ברצף מא"ב מסויים, יש לבדוק שכל הא"ב בקלט הנתון מוכר לפונקציה ואין אות מסוג שונה.
  - עבור רצף, יש לוודא כי הקלט מכיל רק אותיות מותרות. לדוגמה, עבור רצף דנ"א ניתן לקבל גם אותיות קטנות וגם גדולות שכן יש להן משמעות דומה, אך אין לקבל רצף שמכיל אותיות מעבר לא"ב {A, T, C, G, a, t, c, g} שכן הפונקציה לא אמורה להתמודד עימן.
  - עבור מספרים, יש לוודא שהמספר עומד בדרישות המתאימות.
  - עבור קלט מתוך סט מוגדר של קלטים, יש לוודא כי מדובר בקלט מוכר לנו.
- באם קלט אינו עומד בסטנדרטים, יש לזרוק AssertionError ואת הקלט הסורר. לדוגמה

```
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AssertionError: 3.5
```

חלק 1: שדרוג פונקציות קודמות

1. בהינתן רצף דנ"א, שדרגו את הפונקציה ***find_srr(dna_seq)*** המוצאת את כל החזרות הפשוטות ברצף לפי ההגדרה הנ"ל כך שתחזיר לכל SRR אפשרי את מספר הפעמים המקסימלי שבו הוא מופיע. רצף דנ"א יחיד עשוי להכיל מספר SRR שונים, יש להחזיר לכל האפשרויות ולכל האורכים (תתי-מיקרוסטליטים) את אורך החזרה הארוכה ביותר. באם אין חזרות יש להחזיר None. אין להניח תקינות קלט.
   החזרות יוחזרו מהפונקציה באיזה אופן שתרצו.
   לכשיודפסו, על החזרות להיות בסדר א"ב כפי שהוא מוגדר עבור מחרוזות בפייתון:

   repeat_sequence1,num_repeats1;repeat_sequence2,num_repeats2

   דוגמה:
   עבור הרצף `ATCAAATCAAATCAAGAGAGAG` הפונקציה תחזיר משתנה, שניתן להדפיסו כך

   A,3;AG,4; ATCAA,3;GA,3

2. בהינתן רצף דנ"א, נרצה להמירו לרצף רנ"א. שדרגו את הפונקציה בשם ***transcribe(dna_seq)*** המבצעת זאת כך שהתרגום יתבצע באמצעות מילון זיווגים, המכיל לכל אות מקורית את התרגום הרצוי. יש להחזיר את הרצף המשועתק לפי הכיווניות המתאימה (5' ל-3'). יש להקפיד על אותיות uppercase בפלט המוחזר. אין להניח תקינות קלט.

3. שדרגו את הפונקציה בשם ***translate(rna_seq)*** המתרגמת את הרנ"א החל מקודון ההתחלה הראשון כך שתחזיר רשימת קודונים (ולא מחרוזת), וכך שתסתמך על סט נתון של קודוני התחלה וסיום, שיוגדר מראש (לא מועבר לפונקציה, אלא נתון כמשתנה פנימי\חיצוני).
   ניתן להניח שהתרגום נמשך עד השלשה המלאה האחרונה או עד לקודון העצירה הראשון (המוקדם מביניהם, לא כולל קודון העצירה עצמו). יש לבחור במסגרת הקריאה הארוכה ביותר. באם מדובר ברנ"א שאינו מקודד, יש להחזיר None. באם לשתי מסגרות קריאה אותו אורך חלבון יש לבחור בראשונה מביניהן. אין להניח תקינות קלט.

חלק 2: פולימראז

הן השכפול והן השעתוק נעשים בד"כ ע"י אנזים הנקרא פולימראז. ישנם מספר סוגים של פולימראזות, ביניהם אנזים משכפל הנקרא דנ"א פולימראז, ואנזים משעתק הנקרא רנ"א פולימראז. צרו מחלקה בשם ***Polymerase*** לפי הדרישות הבאות:

1. פולימראז מאותחל עם סוג הפולימראז (RNA או DNA) וקצב הטעויות (מספר עשרוני בין 0 ל-1, כאשר ברירת המחדל היא 0) שלו באופן הבא:

Polymerase(self, type, error_rate)

2. פולימראז יודע לשעתק רצף בהתאם לסוג המתאים. היעזרו בפונקציה המשודרגת ***transcribe(dna_seq)*** מהחלק הקודם והכניסו אותה לתוך המחלקה תוך ביצוע התאמות רלוונטיות. שימו לב כי הפולימראז יכול להיות דנ"א פולימראז <u>או</u> רנ"א פולימראז, אך לא שניהם. ודאו כי הפונקציה עומדת בכך.

<u>אין להניח תקינות קלט עבור אף אחד מהסעיפים.</u>

חלק 3: ריבוזום

תרגום מתבצע באופן סטנדרטי ע"י ריבוזום. כתבו מחלקה בשם ***Ribosome*** לפי הדרישות הבאות:

1. ריבוזום מאותחל עם הקוד הגנטי הרלוונטי וקודוני ההתחלה ליצור בו הוא פועל באופן הבא:

Ribosome(self, genetic_code, start_codons)

2. ריבוזום יכול לתרגם רצף רנ"א נתון ע"י זיהוי מסגרת הקריאה הארוכה ביותר האפשרית, ותרגומה עד קודון העצירה האחרון באמצעות פונקציית ***synthesize(self, rna_seq)*** המחזירה מחרוזת של רצף חלבון מלא. באם מדובר ברצף שאינו מקודד יש להחזיר None. העזרו בפונקציה ***translate(rna_seq)*** המשודרגת מהחלק הקודם והכניסו אותה לתוך המחלקה, תוך ביצוע התאמות רלוונטיות. שימו לב לשימוש בקוד הגנטי וקודוני ההתחלה המתאימים.

<u>ניתן להניח כי הקוד הגנטי נתון כמילון בו המפתחות הם הקודונים והערכים הם חומצות האמינו התואמות. עבור קודון עצירה הערך יהיה None. אין לבצע בדיקה כפולה של תקינות קלט – מספיק שאחת הפונקציות בודקת.</u>

חלק 4: תא

תא סטנדרטי מכיל גנום, דנ"א פולימראז, רנ"א פולימראז, וריבוזום. תא יודע להתחלק במיטוזה, ולתרגם חלבונים. כתבו מחלקה בשם ***Cell*** לפי הדרישות הבאות:

1. תא מאותחל עם שם התא, גנום, מספר העותקים מהגנום (מספר שלם הגדול מאפס), קוד גנטי סטנדרטי, סט של קודוני התחלה אפשריים, וקצב חלוקה (מספר שלם הגדול מאחד):

Cell(self, name, genome, num_copies, genetic_code, start_codons, division_rate)

2. גנום הוא רשימה של רצפי דנ"א.
3. תא בונה לעצמו את הפולימראזות שלו וכן את הריבוזום בשלב האתחול.
4. כאשר תא מודפס, הוא מיוצג ע"י מחרוזת המכילה את שם התא, את מספר העותקים בגנום וקצב החלוקה. הוסיפו בקצוות <> לשם זיהוי המידע כיחידה של אובייקט יחיד. שימו לב להבדלים בין __str__() והפונקציה __repr__(), לצורך התרגיל הנוכחי מספיק להדפיס בהתאם לפורמט. אין צורך שדריסה תהיה רלוונטית להשוואות בשלב זה.

<name, num_copies, division_rate>

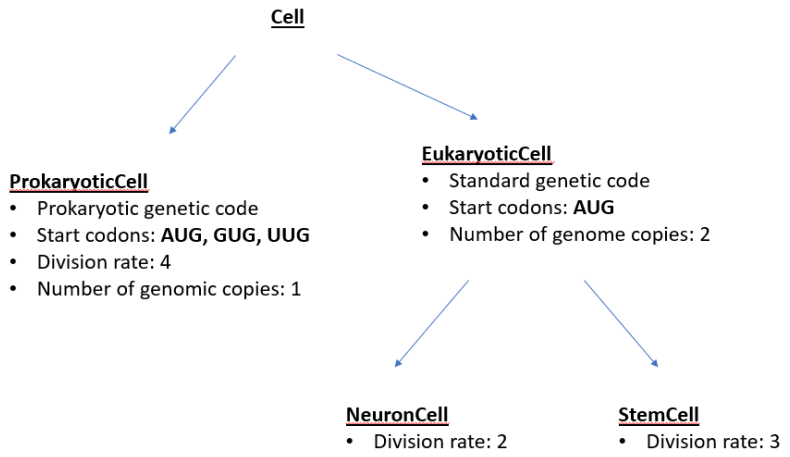
5. תא יכול לעבור מיטוזה ע"י קריאה לפונקציה ***mitosis()*** המחזירה רשימה עם תאים זהים במדויק שאורכה בהתאם לקצב החלוקה של התא, או על ידי **אופרטור הכפל** המחזיר רשימה עם מספר העותקים הרצוי בפעולה.
6. תא בעל מספר זוגי של עותקי גנום יכול לעבור מיוזה ע"י קריאה לפונקציה ***meiosis()*** המחזירה רשימה של שני תאים, אחד עם הגנום המקורי ואחד עם הגדיל המשלים שלו. לכל אחד מספר עותקי גנום המחולק בשניים. אם התא אינו יכול לעבור מיוזה הפונקציה מחזירה None.
7. תא יכול לתת את רפרטואר הגנום שלו ע"י קריאה לפונקציה ***repertoire()*** המחזירה רשימה, בה לכל רצף בגנום טאפל לפי הסדר הבא:
   - החזרות המופיעות באותו רצף (לפי ההגדרות בחלק 1.1). אם אין חזרות יש להחזיר No simple repeats in DNA sequence
   - רצף הרנ"א המשועתק
   - החלבון שנוצר מכל אחד מהרצפים בגנום, לפי חוקי הריבוזום שהוגדרו לעיל (בחלק 1.3). אם מדובר ברצף שאינו מקודד יש להחזיר Non-coding RNA

ניתן להניח כי הקוד הגנטי, וקודוני ההתחלה נתונים כדרוש. לא ניתן להניח תקינות קלט עבור הגנום, מספר העותקים וקצב החלוקה.

יש להעזר בחלקים הקודמים בהם בנינו את המחלקות הדרושות למימוש תא, וכן בפונקציה לזיהוי חזרות שניתן להכניס ישירות למחלקת התא.

חלק 5: סוגי תאים ספציפיים

ישנם מגוון תאים בעלי חלק מהתכונות הידועות מראש. לכל אחד מהתאים האלו, צרו מחלקה היורשת מ-Cell ומכילה את התכונות הידועות לפי התרשים הבא:



שימו לב להדפיס את שם המחלקה בהתאם לסוג התא.

תוכנית ראשית (15 נק'):

מומלץ להיעזר בתיעוד כאן בנוגע לשימוש ב-__name__ למעוניינים אך אין חובה.

התוכנית הראשית תתפקד כמעין תרבית תאים בהדגרה. התאים ייבחרו מתוך שלושת הסוגים הנמצאים בקצות העץ ההיררכי: תאים פרוקריוטים (ProkaryoticCell), נוירונים (NeuronCell) ותאי גזע (StemCell, הקפידו לרשום בהתאם!)

על התוכנית הראשית לקבל שם של תא (קלט ראשון, מחרוזת), מספר מחזורי חלוקה כולל (קלט שני, מספר שלם גדול מאפס), מספר תאים מקסימלי אפשרי (קלט שלישי, מספר שלם גדול מאחד) ורצפים גנומיים (החל מקלט רביעי והלאה).

1. התוכנית תבחר את סוג התא המתאים לקלט ותאתחל אותו (מומלץ להעזר בפונקציה המתפקדת כ-factory אך אין חובה).
2. כל עוד נותרו מחזורי חלוקה מיוטית וגם לא הגענו למספר התאים המקסימלי האפשרי, יש לבצע חלוקה של התא המקורי וצאצאיו (כלומר כל התאים בתרבית).
3. על התוכנית להדפיס את התא הקיים בתרבית, את מספר התאים שהתקבל בסיום, ואת הרפרטואר של אחד מהם.
4. לבסוף על התא לבצע מיוזה ולהדפיס את הגנום והרפטואר של שני תאי הבת.

<u>אין להניח תקינות קלט, אין צורך לבדוק פעמיים רצפים הגנומיים וניתן להסתפק בבדיקה של פונקציה פנימית.</u>

דוגמת הרצה 1:

python exercise2_123456789.py StemCell 3 100 TTGATCTGCATGTTCATGAT ATCAAATCAAATCAAATCAA TTACATCAT

פלט:

Original cell: <StemCell, 2, 3>

Final number of cells: 27

Repertoire: [('No simple repeats in DNA sequence', 'AUCAUGAACAUGCAGAUCAA', 'MNMQI'), ('A,3;AAATC,3;AATCA,3;ATCAA,4;CAAAT,3;TCAAA,3', 'UUGAUUUGAUUUGAUUUGAU', 'Non-coding RNA'), ('No simple repeats in DNA sequence', 'AUGAUGUAA', 'MM')]

Undergoing meiosis...

First cell genome: ['TTGATCTGCATGTTCATGAT', 'ATCAAATCAAATCAAATCAA', 'TTACATCAT']

First cell repertoire: [('No simple repeats in DNA sequence', 'AUCAUGAACAUGCAGAUCAA', 'MNMQI'), ('A,3;AAATC,3;AATCA,3;ATCAA,4;CAAAT,3;TCAAA,3', 'UUGAUUUGAUUUGAUUUGAU', 'Non-coding RNA'), ('No simple repeats in DNA sequence', 'AUGAUGUAA', 'MM')]

Second cell genome: ['ATCATGAACATGCAGATCAA', 'TTGATTTGATTTGATTTGAT', 'ATGATGTAA']

Second cell repertoire: [('No simple repeats in DNA sequence', 'UUGAUCUGCAUGUUCAUGAU', 'MFM'), ('ATTTG,3;GATTT,3;T,3;TGATT,3;TTGAT,4;TTTGA,3', 'AUCAAAUCAAAUCAAAUCAA', 'Non-coding RNA'), ('No simple repeats in DNA sequence', 'UUACAUCAU', 'Non-coding RNA')]

דוגמת הרצה 2:

python exercise2_123456789.py ProkaryoticCell 5 999 AGCTAGTCAATTGATCTGCATGTTCATGAT ATCGAGTTTGGGTTGTGTGTGAAGTCATC ATCATCGCTACTA

פלט:

Original cell: <ProkaryoticCell, 1, 4>

Final number of cells: 997

Repertoire: [('No simple repeats in DNA sequence', 'AUCAUGAACAUGCAGAUCAAUUGACUAGCU', 'MNMQINULA'), ('G,3;GT,3;T,3;TG,4', 'GAUGACUUCACACACAACCCAAACUCGAU', 'MTSHTTQTR'), ('No simple repeats in DNA sequence', 'UAGUAGCGAUGAU', 'M')]

Cannot undergo meiosis