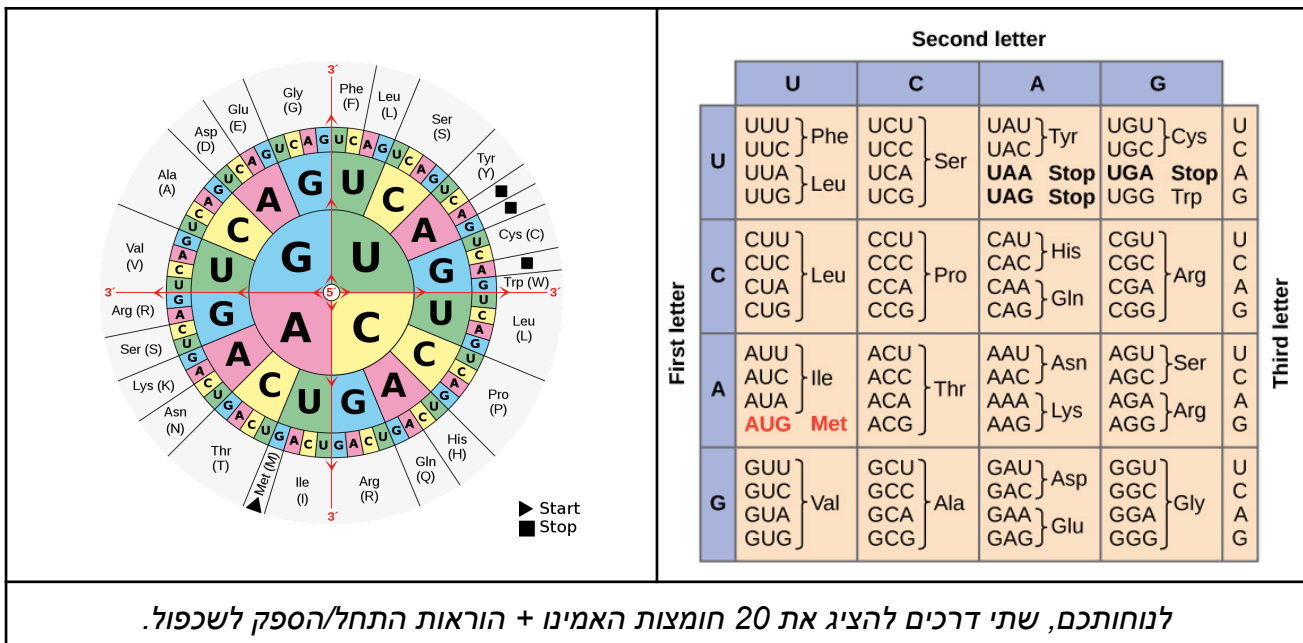


תרגיל בית - ביולוגיה חישובית

עבודה זו מסכמת את הנושאים שנלמדו בנושא ביולוגיה חישובית (Computational Biology):
עבודה עם רצפי DNA ■ אבולוציה של אוכלוסיות ■ יחסי טורף-נטרף (Predator-Prey Relations)

עבודה עם רצפי DNA - זיהוי רצף הגנום האנושי מחתיכות אקראיות שלו

1. בכיתה ראינו איך שלשות (רצף של 3 חומצות בסיס, למשל: AAG, או ACT) מייצגים חומצת אמינו אחת מתוך 20 ועוד 3 "הוראות הפעלה" לשכפול הגנטי. סה"כ, 23 ייצוגים שונים. חוקרים גילו שטעויות שכפול קורות, בעיקר, באות השלישית של הקודון (האות האחרונה ברצפים להלן). אם הבסיס האחרון בשלישיה משתנה:
 - 1.1. מה הסיכוי שנייצר את אותה חומצת אמינו שהקוד המקורי התכוון לייצר?
 - 1.2. האם אפשר להגיע לאפס תקלות, כלומר, שכל טעות בבסיס האחרון לא תגרום לטעות בייצוג של הקודון מבחינת חומצת אמינו/פעולת עריכה?
 - 1.3. אם ה-DNA היה מקודד את חומצות האמינו בעזרת בייט (Byte) כלומר, 8 סיביות של 0/1 לקידוד כל חומצה אמינית. האם בתנאים כאלו אפשר לייצר ייצוג ל-20 החומצות האמיניות (ועוד 2 קודי הפעלה לשכפול, סה"כ 22 איברים) כך שטעות בהעתקה של סיבית בודדת כלשהי בייצוג, לא תשנה ולא תגרום לייצור חומצה אחרת. אם אפשר - הסבר/י איך לעשות זאת. אם אי אפשר - נמק/י היטב את תשובתך. תשובה נכונה ללא הסבר מתאים לא תזכה במלוא הנקודות.
- הערה:** על מנת לענות על סעיף זה, צריך להכיר קודים שמזהים טעות וגם יודעים באיזו סיבית הטעות קרתה כך שאפשר לתקן כל טעות של סיבית בודדת בייצוג החומצה. אם אתם לא מכירים קוד כזה מהלימודים, עיינו בערך ויקידיפה: Hamming code



לנוחותכם, שתי דרכים להציג את 20 חומצות האמינו + הוראות התחל/הספק לשכפול.

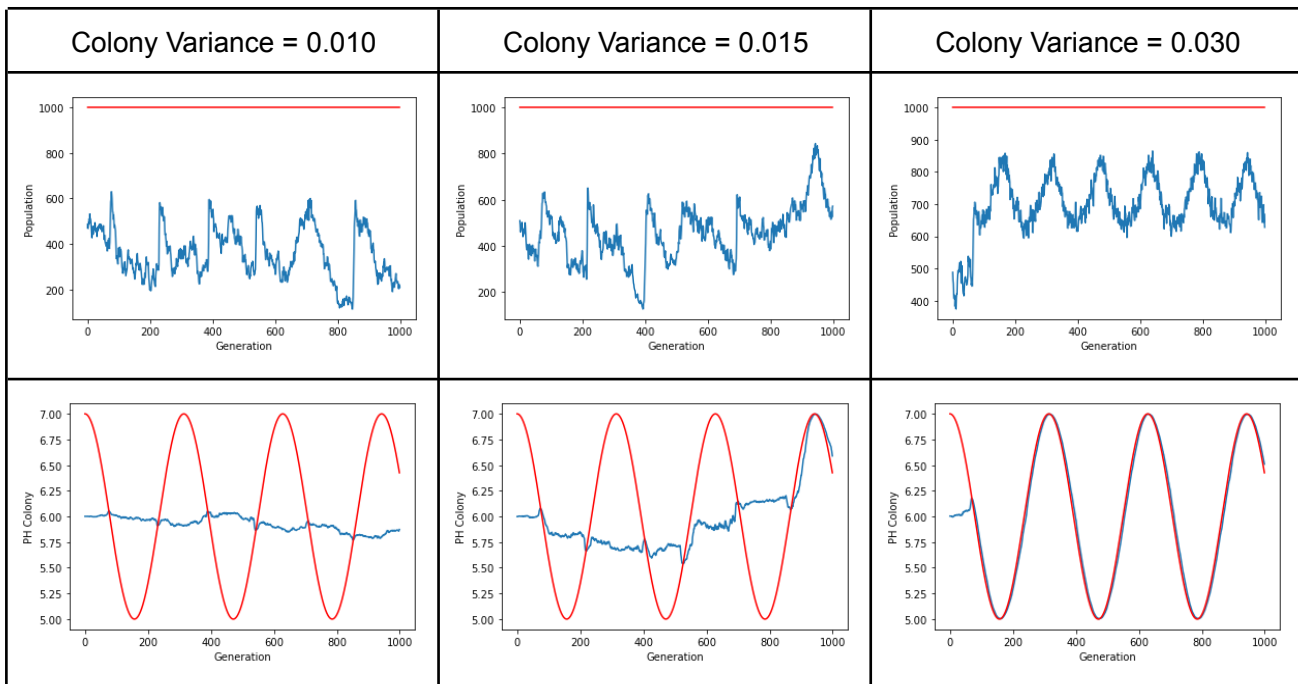
2. אנו רוצים לייצג 50,000 חלבונים שונים בגוף האדם על ידי רצף של חומצות אמינו (שבעצמן, כפי שלמדנו, מיוצגות כל אחת על ידי 3 בסיסים בקוד הגנטי, למשל: acc). עליכם להציע שיטה יעילה לקידוד החלבונים על ידי 20 חומצות האמינו השונות בגוף האדם. מותר לקודד כל חלבון ברצף ח. אמינו לבחירתם. התנאי היחיד הוא שכל חלבון חייב לקבל ייצוג שונה מכל חלבון אחר בגוף האדם. לצורך השאלה כאן, קידוד "יעיל" הוא קידוד שהאורך הארוך ביותר של קידוד חלבון הוא המספר הקטן ביותר האפשרי.

- 2.1. מה האורך של הקידוד הארוך ביותר של חלבון לפי השיטה שהצעת? יש להציג חישוב ברור עם הסברים מפורטים. תשובה לא ברורה, גם אם תהיה נכונה, לא תזכה במלוא הנקודות.
- 2.2. מה האורך הממוצע לקידוד חלבון בגוף האדם? הניחו שכל 50.000 החלבונים מופיעים באותם שכיחויות (זה לא נכון, אבל נעשה את ההנחה הזאת לצורך התרגיל).
- 2.3. במציאות, האורך הממוצע של חלבון בגוף האדם הוא כ-500 חומצות אמינו. מה לדעתך ההסבר להבדל בין האורך הממוצע שיצא בקידוד שלכם לעומת האורך הממוצע בטבע? מספיק לתת סיבה אחת עיקרית ולהסביר היטב מדוע היא הסיבה העיקרית לדעתכם.

אבולוציה של אוכלוסיות

בשיעור ראינו קוד המיישם התפתחות אוכלוסיית חיידקים (Bacteria) לאורך זמן בסביבה משתנה. ראינו איך אוכלוסיית החיידקים תשנה לאורך הזמן את העמידות שלה לרמות משתנות של PH כי חיידקים שמתאימים יותר לסביבת PH נתונה ישרדו ויתחלקו יותר, חיידקים עם ערכים פחות מתאימים, ישרדו פחות ובסוף יכחדו.

להלן תוצאות הרצת הקוד שראינו בכיתה כשלאוכלוסית החיידקים יש רמות שונות של יכולת התאמה לסביבה. כלומר, יש חיידקים שמבחינה גנטית מסוגלים לייצר שינוי מהותי יותר לדור הבא בהתאמתם לרמת PH הסביבה, ויש חיידקים שפחות. מידת השינוי נקבעת על ידי המשתנה Variance. להלן תוצאות הרצה ל-3 סימולציות עם Variance שונים:



3. הסבירו מילולית, מה אנו רואים בכל הרצה תוך ציון ההבדל בתוצאות בין ההרצות השונות. יש לציין (א) מה רואים בכל גרף. (ב) מה ההסבר שלכם לכך (ג) יש להתייחס לקשר בין גודל האוכלוסיה לאורך הזמן בגרף הראשון של ההרצה לבין התאמת, או אי התאמת ה-PH של החיידקים לסביבה כפי שרואים בגרף השני.

בכיתה ראינו סרטון קצר המסביר מדוע אוכלוסית הדובים בקוטב הפכה לאורך עשרות אלפי שנים לבעלת פרווה לבנה למרות שבמקור הדובים הראשונים שהגיעו לקוטב היו חומים. (לינק לסרטון מהכיתה [כאן](#))

4. כתבו קוד שמדגים את השינוי בצבע הפרווה של אוכלוסיית הדובים לאורך הדורות בדומה לקוד שראיתם בכיתה המדגים את התפתחות מושבת החיידקים והתאמתם ל-PH של הסביבה בה הם חיים.

להלן הקוד שראינו בכיתה (נמצא כקובץ באתר הקורס: Computational Biology-Evolution.ipynb)

```
# Initial conditions

N = 500                                # Population size
Max_Population = 1000                 # When is "too much" bacteria in the colony?
PH_Env = 2.0                          # External (Environment) PH
PH0_bacteria = 5.0                   # Set all Bacteria in colony to best live in PH0
                                      # that is different from the environment PH
Variance = 0.25                      # The variance in PH we allow when bacteria reproduce
                                      # (if =0, all generations will have same PH)
Simulation_time = 100                # How many generations we will simulate

Generations = range(0,Simulation_time)

# Create a population of N bacteria in the colony with individual PHx
Population = []
PopulationSize = []
Average_bacteri_PH = []

for i in range(0,N):
    PHx = PH0_bacteria + (random.random()*2-1)*Variance*2
    Population.append(PHx)

print(f'Population size: {len(Population)}\nAverage PH:
{np.mean(Population)}\nOutside PH: {PH_Env}')

# Start evolution generation by generation

for Generation in Generations:

    for x in Population:

        bacteri_PH = x

        Survive_PH = False
        if np.abs(PH_Env-bacteri_PH)<Variance:
            Survive_PH = True
        elif random.randint(1,2)==2:
            Survive_PH = True

        Survive_pop = False
        if len(Population)>=Max_Population:
```

```

        Survive_pop = False
    elif random.randint(1,len(Population))<Max_Population/2:
        Survive_pop = True

    if Survive_PH and Survive_pop:
        # This bacteria reproduce in this generation
        # Choose 2 new PH for the 2 outcomes
        PH_1 = bacteri_PH+(random.random()*2-1)*Variance
        PH_2 = bacteri_PH+(random.random()*2-1)*Variance

        Population.append(PH_1)
        Population.append(PH_2)

    # delete the old bacteria after it did, or did not, reproduce
    Population.remove(bacteri_PH)

# The whole colony was regenerated - Now we register few averages

PopulationSize.append(len(Population))
Average_bacteri_PH.append(np.mean(Population))

if DebugMode:
    print(f'PopulationSize = {len(Population)}')
    print(f'Average_bacteri_PH = {np.mean(Population)}')

print('Finished calculating the simulation, here are the results:')

plot.plot(Generations,PopulationSize)
plot.plot(Generations,Max_Population*np.ones(len(Generations)),"r-")
plot.xlabel("Generation")
plot.ylabel("Population")
plot.show()

plot.plot(Generations,Average_bacteri_PH)
plot.plot(Generations,PH_Env*np.ones(len(Generations)),"r-")
plot.xlabel("Generation")
plot.ylabel("PH Colony")
plot.show()

```

יחסי טורף נטרף (Predator Prey Relations)

בשיעור ראינו איך אוכלוסיית טורפים ונטרפים משפיעה הדדית זו על זו במרחב המחיה המשותף לטורפים והנטרפים. ראינו הדגמה על אוכלוסיית שועלים (טורפים) וארנבות (נטרפים).

לפניכם הקוד שמבצע סימולציה של השינוי באוכלוסיית הטורפים והנטרפים לאורך זמן במקרה של שועלים וארנבות.

```
from scipy.integrate import odeint
```

```

import matplotlib.pyplot as plt
import numpy as npy
import random

# timestep determines the accuracy of the euler method of integration
timestep = 0.0001
# amplitude of noise term
amp = 0.00
# the time at which the simulation ends
end_time = 50

# creates a time vector from 0 to end_time, seperated by a timestep
t = npy.arange(0,end_time,timestep)

# intialize rabbits (x) and foxes (y) vectors
x = []
y = []

""" definition of lotka-volterra parameters """
a = 1    # birth rate of rabbits
b = 0.1  # death rate of rabbits due to predation
c = 0.5  # natural death rate of foxes
d = 0.02 # factor that describes how many eaten rabbits give birth to a new fox

""" calculates the change in the two population over time """
# initial conditions for the rabbit (x) and fox (y) populations at time=0
x.append(100)
y.append(20)

for index in range(1,len(t)):

    # evaluate the differentials (change in population) for this iteration
    xd = x[index-1] * (a - b*y[index-1])
    yd = -y[index-1]*(c - d*x[index-1])

    # evaluate the next value of x and y populations
    next_x = x[index-1] + xd * timestep
    next_y = y[index-1] + yd * timestep

    # add the next value of x and y
    x.append(next_x)
    y.append(next_y)

""" visualization """

plt.plot(t, x)
plt.plot(t, y)
plt.xlabel('Time')
plt.ylabel('Population Size')
plt.legend(('Rabbits', 'Foxes'))
plt.title('Deterministic Lotka-Volterra')
plt.show()

```

```
plt.plot(x,y)
plt.xlabel('Rabbit Population')
plt.ylabel('Fox Population')
plt.title('Phase Portrait of Deterministic Lotka-Volterra')
plt.show()
```

5. שנו את הקוד כך שקצב הרבייה של הארנבות מושפע מגודל אוכלוסיית הארנבות. כלומר: במקום קצב קבוע a , הקצב יהיה $a = 3 - \log(X[i])$ כאשר \log הוא לוגריתם בבסיס 10.
- 5.1 הגישו את הקוד המתוקן ואת שני הגרפים, כפי שהם בקוד הנוכחי, ללא שינוי.
- 5.2 הסבירו את ההבדל בתוצאות
- 5.3 מה האוכלוסיה המירבית של ארנבים שתתכן ביער, הנובעת מההגדרה החדשה של קצב ההתרבות שלהם (שאלה זו קשורה רק להבנת מקדם הרבייה החדש שלהם, לא לסימולציה)

--- סוף התרגיל ---