

תרגיל 1 - עימוד רצפים

תאריך הגשה: 19 בנובמבר 2019

הגישו קובץ ex1.tar המכיל:

- קובץ pdf עם פתרון מוקלד של החלק התיאורטי של התרגיל
- קובץ/קבצי Python3 עם מימוש החלק התכנותי של התרגיל. בפרט, את הקובץ seq_align.py.

חלק תיאורטי

1 מספר העימודים האפשריים

כמוטיבציה לפיתוח אלגוריתמי עימוד חכמים, נרצה להעריך את מספר העימודים האפשריים שהאלגוריתם יצטרך לכסות. יהיו שני רצפים s, t , באורך n כל אחד. האלגוריתם מעמד באופן גלובלי, והשימוש ברווחים (gap) מותר. הניחו כי אורך הרצפים המעומדים לא עולה על $2n$. הראו כי מספר העימודים האפשריים הוא לפחות 3^n .

2 קנס ליניארי על רווחים - Linear Gap Penalty

באלגוריתמי העימוד שהוצגו בשיעור, הקנס על כל רווח היה קבוע, ללא תלות במספר הרווחים. כלומר, אם הקנס על רווח בודד הוא d , אז על רווח של g בסיסים נקנסים $p = gd$.

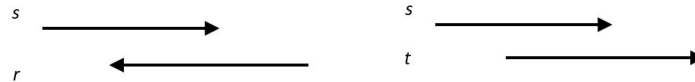
נתבונן במודל מולקולרי, בו כאשר DNA נשבר, עשוי להיכנס מספר גדול של בסיסים. נשתמש במודל ליניארי פשוט, לפיו ניתן קנס מסויים, d , על הרווח הראשון, וקנס אחר, e , על כל רווח נוסף. אז הקנס על g רווחים רצופים יהיה $p = d + e(g - 1)$. לרוב יתקיים $d > e$. כתבו פסאודו קוד לאלגוריתם עימוד גלובלי, עם קנס ליניארי על רווחים כמתואר לעיל. על הקוד לכלול אתחול, את נוסחת הרקורסיה, ואת אופן אחזור העימוד האופטימלי.

3 עימוד חפיפה - Overlap Alignment

בריצוף Paired-end מתקבלות לעתים קרובות שתי קריאות (reads) שיש ביניהן חפיפה. נסמן את שתי הקריאות ב- s, r , ונניח שיש ביניהן איזור חופף. נסמן ב- t את ההופכי המשלים של r . נרצה לעמד את האיזור החופף בלבד. כלומר, לעמד את הסיפא של s עם

הרישא של t . ניתן לעשות זאת באמצעות עימוד שתי הקריאות ללא קנס על רווחים בהתחלה ובסוף.

עבור רצפים s, t , מצאו את העימוד האופטימלי בין הסיפא של s לבין הרישא של t . כלומר, מצאו עימוד בין s לבין t ללא קנס על רווחים ברישא של s או בסיפא של t .



כתבו פסאודו קוד לעימוד חפיפה כנ"ל. על הקוד לכלול אתחול, את נוסחת הרקורסיה, ואת אופן אחזור העימוד האופטימלי.

4 עימוד גלובלי במקום ליניארי - Linear Space Global Alignment

עבור שני רצפים באורכים $n \approx m$, אלגוריתם העימוד הסטנדרטי משתמש במקום $n \times m$, כלומר ריבועי בגודל הקלט. עבור קלטים גדולים, זה עלול להיות לא מעשי. במקרים כאלה, ניתן לשדרג את האלגוריתם כך שיעשה שימוש במקום ליניארי בגודל הקלט. בכיתה ראיתם שיטה של "הפרד ומשול", שבה שומרים את הניקוד שאיתו חוצים את העמודה ה- $\lfloor \frac{n}{2} \rfloor$. כתבו פסאודו קוד לעימוד גלובלי המשתמש במקום ליניארי כמתואר לעיל. על הקוד לכלול אתחול, את נוסחת הרקורסיה, ואת אופן אחזור העימוד האופטימלי.

חלק תכנותי

5 עימוד באמצעות תכנון דינמי

בחלק זה תממשו Python מעמד רצפים (sequence aligner), שבהינתן שני רצפים, סוג עימוד, ומטריצת ניקוד, מדפיס את העימוד האופטימלי ואת הניקוד שלו. הרצפים יתקבלו בקבצי FASTA - פורמט מקובל עבור רצפים ביולוגיים. דוגמה לתוכן של קובץ fasta:

```
>name of seq1
ACACGGTGGACCGGAT
AACACGGTAATACCAG
```

קלט - רצפים ומטריצת ניקוד:

שני קבצי fasta. ניתן להניח שהרצפים בקבצים אלה מגיעים מהאלפבית $\Sigma = \{A, C, G, T\}$. מומלץ להשתמש בפונקציה `fastaread()` שמומשה עבורכם בקובץ `seq_align.py`. בתרגיל זה ניתן להניח שבכל קובץ fasta יש רצף אחד בלבד. מטריצת ניקוד S . השורה הראשונה והעמודה הראשונה מתארות את אותיות האלפבית, או רווח (-). שאר התאים בטבלה מתארים ניקוד עבור החלפה, התאמה או מחיקה. לדוגמה,

$S_{T,-} = \sigma(T, -)$, באופן דומה, A עם A . $S_{A,A} = \sigma(A, A)$ הוא הניקוד עבור עימוד של T מול רווח. המטריצה ניתנת כקובץ `tsv`. ראו קובץ לדוגמה, "score_matrix.tsv", עם מטריצה שנלקחה מ-NUC 4.2. ניתן להניח שהמטריצה נתונה בפורמט זה, ובסדר קבוע של עמודות/שורות (נוקלאוטידים): $[A, C, G, T]$. בנוסף, בתרגיל זה ניתן להניח שהמטריצה סימטרית, כלומר מתקיים $\sigma(X, Y) = \sigma(Y, X)$.

פלט - עימוד וניקוד:

בהינתן זוג רצפי קלט, $X = (x_1, \dots, x_n)$ ו- $Y = (y_1, \dots, y_m)$, על התוכנית לתמוך בשלושה סוגי עימוד:

עימוד גלובלי ("global") - בסוג עימוד זה, האלגוריתם מחפש את ההתאמה הטובה ביותר בין X לבין Y , כך שכל התווים ב- X וב- Y מועמדים אחד מול השני (כלומר x_i מול y_j , כאשר $1 \leq j \leq m, 1 \leq i \leq n$), או מול רווחים (x_i מול רווח או y_j מול רווח).

עימוד לוקאלי ("local") - בסוג עימוד זה, האלגוריתם מחפש את העימוד האופטימלי של תתי מחרוזות של X, Y . כלומר את העימוד הטוב ביותר של תתי מחרוזות $X_{s1:e1}$ עם $Y_{s2:e2} = (y_{s2}, \dots, y_{e2})$, כך ש- $1 \leq s1 \leq e1 \leq n$ ו- $1 \leq s2 \leq e2 \leq m$.

עימוד חפיפה ("overlap") - עימוד זה שימושי כאשר אחד מהרצפים הוא תתי מחרוזת של השני, או כאשר הרישא של אחד חופפת לסיפא של השני (ראו שאלה 3). עימוד זה דומה לעימוד גלובלי, אך אין קנס על רווחים בהתחלה ובסוף העימוד.

דרישות טכניות

ניתן יהיה להפעיל את התוכנה מהטרמינל, באמצעות השורה:

```
python3 seq_align.py a.fasta b.fasta --align_type global
--score score_matrix.tsv
```

על התוכנה להדפיס את העימוד האופטימלי, ומיד אחריו את סוג העימוד, ואת הניקוד, כמו בדוגמה הבאה:

```
TCGAATCG -- CACGCGCGGCTCTCCTTAGAACCGGCCGGCTCCCGAATAA
TTGGGTCGGTTTACCCGGTCTTCATCCG -- CCGACTGTTTAAAAACCAA
```

```
TGTTTCAGTGTGTTTGACAAACTCAATCGGAGGTCTCGGAAGA --- AGTATC
CAAGGTAAGAGGAGGGGAGCTTTGTTGTTGTTTAACTGTGTAGTGAC
```

```
AAAAAAAAAAAAAA
AAAAAAAAAAAAAA
```

```
global:20
```

אם ישנו יותר מעימוד אופטימלי אחד, מספיק להדפיס אחד מהם.
אופן ההדפסה: העימוד יודפס כבלוקים של שתי שורות (שורה לכל אחד משני הרצפים),
עם עד 50 תווים בשורה. לאחר כל בלוק תהיה שורה ריקה. לאחר הרצפים, יודפס
`type : score`, כאשר `type` הוא סוג העימוד (`global`, `local`, `overlap`), ו-`score` הוא הניקוד
(מספר).

סיכום הקבצים הנתונים לתרגיל התכנותי:

1. `seq_align.py` - שלד לסקריפט ה-Python שלכם.
2. `score_matrix.tsv` - מטריצת ניקוד לדוגמה.
3. תיקיית `fastas` - התיקייה מכילה מספר קבצי `fasta` לדוגמה.