אלגוריתמים בביולוגיה חישובית / תרגיל 1 - עימוד רצפים

תאריך הגשה: 18 בינואר 2024

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 I

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 אקספוננציאלי ב-n.

2 קנס אפיני על רווחים - Affine 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 אפיני 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 ניתוח זמן ריצה

תרגיל זה מתקשר לחלק התכנותי. בתרגיל זה תתבקשו לממש אלגוריתם עימוד גלובלי ולנתח באופן אמפירי את זמן הריצה. צרו רצפים באורכים שונים ומדדו את זמן הריצה ואת כמות הזיכרון שהתכנית משתמשת בו עבור קלטים באורכים שונים. נסו להגדיל את אורך הרצף הראשון כאשר אורך הרצף השני קבוע, להגדיל את אורך הרצף השני כאשר אורך הרצף הראשון קבוע, ולהגדיל את אורכי שני הרצפים יחד. הציגו את מסקנותיכם בגרפים ובפסקה קצרה (עד 4 משפטים), כחלק מקובץ ה-pdf.

טיפ - בלינוקס אפשר למדוד זמן ריצה ושימוש בזיכרון באמצעות הפקודה הבאה:

```
$ command time -f "max_mem=%M elapsed=%E" python seq_align.py ...
```

חלק II

חלק תכנותי

4 עימוד באמצעות תכנון דינמי

בחלק זה תממשו בPython3 תוכנית לעימוד רצפים (sequence aligner), שבהינתן שני רצפים, סוג עימוד, ומטריצת ניקוד, מדפיסה את העימוד האופטימלי ואת הניקוד שלו.

הרצפים יתקבלו בקבצי FASTA - פורמט מקובל עבור רצפים ביולוגיים.

דוגמה לתוכן של קובץ fasta עם רצף אחד ארוך המתפרס לשתי שורות:

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

קלט - רצפים ומטריצת ניקוד:

1. <u>שני קבצי fasta.</u> ניתן להניח שהרצפים בקבצים אלה מגיעים מהאלפבית $\Sigma = \{A, C, G, T\}$. ניתן להשתמש בספריה biopython, או בפונקציה $fastaread()$ שמומשה עבורכם בקובץ seq_align.py. בתרגיל זה ניתן להניח שבכל קובץ fasta יש רצף אחד בלבד.

2. <u>מטריצת ניקוד S.</u> השורה הראשונה והעמודה הראשונה מתארות את אותיות האלפבית, או רווח ($-$). שאר התאים בטבלה מתארים ניקוד עבור החלפה, התאמה או מחיקה. לדוגמה, $S_{A,A} = \sigma(A, A)$ הוא הניקוד עבור התאמה של A עם A. באופן דומה, $S_{T,-} = \sigma(T, -)$ הוא הניקוד עבור עימוד של T מול רווח. המטריצה ניתנת כקובץ tsv. ראו קובץ לדוגמה, "score_matrix.tsv". 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, $S = (s_1, ..., s_n)$ ו-$T = (t_1, ..., t_m)$, על התוכנית לתמוך בשלושה סוגי עימוד:

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

עימוד לוקאלי (local) - בסוג עימוד זה, האלגוריתם מחפש את העימוד האופטימלי של תתי מחרוזות של S, T. כלומר את העימוד הטוב ביותר של תת מחרוזת $S_{b1:e1} = (s_{b1}, ..., s_{e1})$ עם תת מחרוזת $T_{b2:e2} = (t_{b2}, ..., t_{e2})$, כך ש-$1 \leq b1 \leq e1 \leq n$ ו-$1 \leq b2 \leq e2 \leq m$. יש להדפיס רק את תתי המחרוזות שנמצאו, ולא את העימוד של המחרוזות במלואן.

עימוד חפיפה (overlap) - עימוד זה שימושי כאשר אחד מהרצפים בשלמותו הוא תת מחרוזת של השני, או כאשר הרישא של אחד חופפת לסיפא של השני. עימוד זה דומה לעימוד גלובלי, אך אין קנס על רווחים בהתחלה ובסוף העימוד. רמז: התבוננו בשורה/עמודה הראשונה והאחרונה. בתרגיל זה יש להניח שהמחרוזת הראשונה בקלט היא הראשונה בעימוד. כלומר, הסיפא שלה חופף לרישא של המחרוזת השניה.

מוטיבציה לעימוד חפיפה: בריצוף Paired-end של מולקולות דנ"א דו-גדילי 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 ברישא של s או בסיפא של t (ללא הגבלת הכלליות).

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 (בה העימוד והניקוד מבוססים על המטריצה score_matrix.tsv):

```
TCGAATC-G-CACGCGCGGCTCTCCTTAGAACCGGCCGGCT---CCCGAA
TTGGGTCGGTTTCACCCGG-TCTTCAT-CCGCCGACTGTTTAAAAACCAA

TAATGTTTCAGTGTTTGACAAACTCAATCGGAGGTCT--CG-GAAGAAGT
CAA-G-GTAAGAG-GAGGGGAGCTTTGTTGTTGTTTTAACGTGTGTTAGT

ATCAAAAAAAAAAAA
GACAAAAAAAAAAAA

global:19
```

אם ישנו יותר מעימוד אופטימלי אחד, מספיק להדפיס אחד מהם.

אופן ההדפסה: העימוד יודפס כבלוקים של שתי שורות - שורה לכל אחד משני הרצפים, לפי הסדר הניתן בקלט, 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), הם יהיו מופרדים ע"י נקודתיים ובלי רווחים.

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.