אלגו' בביולוגיה חישובית (76558)
תרגיל 3: שחזור עצי אבולוציה מרצפי דנ"א

תאריך הגשה: 28/1/2025

בתרגיל זה תקבלו כקלט מספר רצפים ביולוגיים בעלי אב קדמון משותף, ותתבקשו לחשב את העץ הפילוגנטי שלהם (דנדרוגרמה), כולל אורכי ענפים.

תיאור הדאטה

מצורפים לתרגיל חמישה קבצי FASTA המכילים רצפים סינתטיים שיצרנו (באורכים שונים). כך למשל, הקובץ ex3.5000.fa מכיל רצפים מועמדים באורך 5000 בסיסים. את הרצפים יצרנו באופן סטוכסטי, על ידי דגימה מאותו העץ (לכל הקבצים), בעזרת מטריצת קצב Jukes-Cantor הבאה:

R	A	C	G	T
A	-1	⅓	⅓	⅓
C	⅓	-1	⅓	⅓
G	⅓	⅓	-1	⅓
T	⅓	⅓	⅓	-1

לנוחיותכם, אין במודל איתו יצרנו את הרצפים indels (כלומר הכנסות ומחיקות). כמו כן, נניח (1) אי-תלות בין העמדות, (2) שמירות אבולוציונית זהה לכל העמודות, ו-(3) רברסיביליות בזמן.

מטרת התרגיל

עליכם לשחזר את העץ שיצר את הרצפים כולל מבנה ואורכי הענפים.

אנו ממליצים לשחזר את העץ באופן הבא:

1. **חשבו את הזמן האבולוציוני שמפריד בין כל שני רצפים**. לשם כך, הניחו שאחד הרצפים הוא האב הקדמון של השני, וחשבו את המרחק האופטימלי (ביחידות זמן) בין זוג רצפים. השתמשו במודל Jukes-Cantor עם מטריצת הקצב R שמופיעה למעלה.
   ניתן לחשב את המרחק בכמה דרכים. תוכלו אולי לחשוב על הבעיה בתור בעיית אופטימיזציה חד-מימדית (line search), בה הקלט הוא הזמן t, והפלט הוא הניראות המותנית של הרצף השני בהינתן הרצף הראשון והזמן שחלף t. ואז אתם מחפשים את ה-t שממקסם את הניראות.
   גירסא יותר פשוטה של זה תחפש את האופטימום בעזרת grid search. כלומר להניח סט בדיד וקבוע מראש של ערכי t אפשריים, ואז חישוב הניראות בכל אחד מהם ובחירת האופטימלי.
   או שתוכלו להיות חכמים יותר, ולגזור את פונקצית הניראות לפי הזמן, וכך למצוא באופן ישיר את הזמן t שממקסם את הניראות, לכל זוג רצפים. כמו שראינו בכיתה, הסטטיסטי המספיק הוא אחוז העמדות הזהות בין שני הרצפים.
   אנא פרטו בצורה ברורה ופורמלית את תשובתכם.
2. אחרי חישוב מטריצת המרחקים בזוגות, **תוכלו לקבץ את הרצפים בצורה אגלומרטיבית**. אנא ישמו את אלגוריתמים ה-UPGMA ואת אלגוריתם ה-Neighbor Joining שראינו בכיתה (Saitou-Nei, להלן NJ), כדי לאחד את הרצפים באופן איטרטיבי, עד להרכבת עץ מלא, כולל חישוב אורכי הענפים.

דרישות

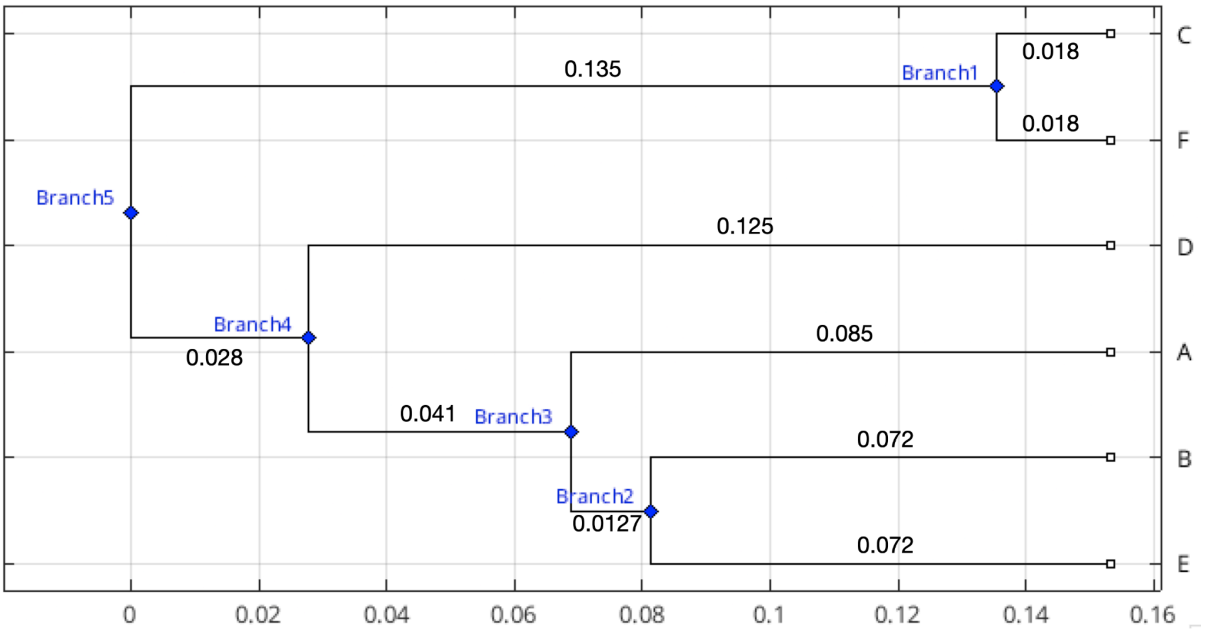
עליכם להגיש הסבר פורמלי של מה שעשיתם, כולל הצדקות והוכחות מתמטיות לכל שלב (למשל, מציאת המרחקים האופטימלי או חישוב אורכי הענפים). לכל קובץ FASTA, עליכם להגיש עץ פילוגנטי בפורמט Newick, כפי המודגם פה (ללא ציון שמות הקודקודים הפנימיים).

https://en.wikipedia.org/wiki/Newick_format

למשל, הסטרינג הבא:

((C:0.018,F:0.018):0.135,(D:0.125,(A:0.085,(B:0.072,E:0.072):0.0127):0.041):0.028);

מייצג את העץ:



ניתוח התוצאות

אנא בנו עצים לכל אחד מקבצי הקלט, ונתחו את ההבדלים ביניהם. השוו בין האלגוריתמים השונים ובין אורכי העימודים השונים. אתם גם מוזמנים לדגום עמדות מהרצפים - כלומר לקצר או להאריך אותם ולנתח את השינויים. אנא שימו לב גם להבדלים איכותיים (מבנה העץ) וגם להבדלים כמותיים (אורכי הענפים).

עליכם לתאר את המודל על פיו אתם עובדים בצורה ברורה ופורמלית, ברמה שתאפשר לנו ליישם מחדש את המודל שלכם, אם נרצה. אנא הקפידו על שרטוטים, תיאור השלבים השונים במהלך העבודה, וכן הלאה. אנא הקפידו לציין בצורה ברורה מה ההנחות שהנחתם וציינו מה הרציונל שלכם.

אנא הקפידו על תיאור ותיעוד ברורים של הקוד שלכם, שיאפשרו לנו לעיין בו ולהבין מה אתם חושבים שעשיתם בקוד.

ניתן להגיש בזוגות.

מה להגיש?

קובץ tar המכיל:

- קובץ PDF ובו תיאור מפורט (בעברית או באנגלית) של הפתרון שלכם, כולל גרפים.
- קבצי טקסט בפורמט Newick, ובהם העצים ששיחזרתם (מכל אחד מקבצי הקלט, בעזרת שני האלגוריתמים). אנא קיראו לקבצי הפלט: ex3.1000.NJ.tree, ex3.50.UPGMA.tree, וכן הלאה.
- קובץ python אחד בשם ex3.py, עם התכנית. אנו נריץ את התכנית על קבצי FASTA נוספים בפורמט זהה, באמצעות הפקודה הבאה:

```
$ python3 ex3.py --fasta_path file.fa --algo algo
```

כאשר algo יקבל ערך אחד מבין "upgma", "nj".

התוכנית שלכם תשחזר עץ לפי הרצפים של העלים בקובץ fasta_file.fa בעזרת אלגוריתם UPGMA או NJ, ותדפיס את העץ ואורכי הענפים בקידוד newick, עם דיוק של שלוש ספרות אחרי הנקודה. אנא הקפידו להדפיס את העץ לפי הפורמט, כולל שמות העלים המקוריים (ללא שמות הקודקודים הפנימיים), ושהתוכנה **לא תדפיס משהו נוסף במהלך הריצה.**

השימוש בכלי עזר לתיכנות מבוססי בינה מלאכותית (דוגמת copilot) מותר, אולם עליכם **להצהיר על כך בקובץ ה-PDF**, לפרט באיזה כלים השתמשתם, מה הפרומפטים שהכנסתם, ולתאר במילים באופן מפורט את תהליך העבודה על התרגיל.