3η Προγραμματιστική Εργασία

Υπολογιστική Βιολογία & Αναζήτηση Δεδομένων

Χειμερινό εξάμηνο 2025-26

Αναζήτηση ”Απομακρυσμένων” Ομολόγων με Προσεγγιστικές Μεθόδους & ESM-2

Η άσκηση θα υλοποιηθεί σε σύστημα Linux με χρήση **Python 3.10+** και των κωδίκων που αναπτύχθηκαν στις Εργασίες 1 και 2.

Προθεσμία παράδοσης: Παρασκευή 16/1, 23.59

Περιγραφή του Προβλήματος

Τα παραδοσιακά εργαλεία στοίχισης ακολουθιών (sequence alignment), όπως το **BLAST**, αδυνατούν συχνά να ανιχνεύσουν **απομακρυσμένες ομόλογες** (remote homologs) πρωτεΐνες — πρωτεΐνες που διατηρούν παρόμοια τρισδιάστατη δομή και λειτουργία παρόλο που έχουν χαμηλή ομοιότητα αλληλουχίας (<30%, γνωστή ως ”Twilight Zone”).

Στόχος της εργασίας είναι η χρήση διανυσματικών αναπαραστάσεων (embeddings) πρωτεϊνών από το μοντέλο **ESM-2** και η εφαρμογή αλγορίθμων **Approximate Nearest Neighbor (ANN)** για τον εντοπισμό απομακρυσμένων ομολόγων. Ως μέθοδοι αναζήτησης θα χρησιμοποιηθούν όλες οι μέθοδοι που υλοποιήθηκαν στις προηγούμενες εργασίες.

Ζητούμενα

Βήμα 1: Παραγωγή Embeddings (ESM-2)

Θα χρησιμοποιήσετε το προεκπαιδευμένο μοντέλο `facebook/esm2_t6_8M_UR50D` για να μετατρέψετε τις πρωτεϊνικές ακολουθίες σε διανύσματα.

- Για κάθε πρωτεΐνη, το διάνυσμα προκύπτει από τον μέσο όρο (mean pooling) των αναπαραστάσεων του τελευταίου επιπέδου.
- Τα διανύσματα πρέπει να αποθηκευτούν σε κατάλληλη μορφή για να τροφοδοτήσουν τους αλγορίθμους σας.

Βήμα 2: Προσαρμογή Αλγορίθμων (Domain Adaptation)

Θα πρέπει να προσαρμόσετε και να εφαρμόσετε τους παρακάτω αλγορίθμους (από τις Εργασίες 1 & 2) στα νέα δεδομένα των πρωτεϊνών:

1. **Euclidean LSH:** Πρέπει να ρυθμίσετε κατάλληλα τις παραμέτρους (k, L, w) για τον χώρο των embeddings.
2. **Hypercube Projection:** Τυχαία προβολή σε υπερκύβο. Αντίστοιχα ρύθμιση των παραμέτρων.
3. **IVF-Flat & IVFPQ:** Χρήση των μεθόδων με ευρετήριο συσταδοποίησης. Αντίστοιχα ρύθμιση των παραμέτρων.
4. **Neural LSH:** Εκπαίδευση του μοντέλου διαμέρισης στα δεδομένα των πρωτεϊνών.

Βήμα 3: Πειραματική Σύγκριση & Βιολογική Αξιολόγηση

Για ένα σύνολο ερωτημάτων (queries), το σύστημα θα εκτελεί αναζήτηση με όλες τις μεθόδους ANN και, για τα ίδια queries, στοίχιση με **BLAST** (local alignment) έναντι της ίδιας βάσης.

- **Ποσοτική σύγκριση:**
  - Υπολογίστε για κάθε μέθοδο χρόνους εκτέλεσης, L2 αποστάσεις των γειτόνων και τη μετρική Recall@N έναντι των κορυφαίων χτυπημάτων του BLAST.
  - Συγκρίνετε τις μεθόδους ως προς τη σχέση ταχύτητας/ακρίβειας, με βάση το Recall@N και το πλήθος των αναζητήσεων ανά δευτερόλεπτο (QPS).
- **Βιολογική αξιολόγηση (remote homologs):**
  - Εστιάστε σε ένα μικρό αλλά αντιπροσωπευτικό υποσύνολο ερωτημάτων (π.χ. 5–10 πρωτεΐνες-στόχους).
  - Για κάθε τέτοιο query, εντοπίστε πρωτεΐνες που:
    - βρίσκονται ψηλά (Top-N) στα αποτελέσματα μιας ή περισσότερων embedding-based μεθόδων (μικρή L2 απόσταση)
    - αλλά παρουσιάζουν **χαμηλό BLAST identity** (π.χ. < 30%).
  - Χρησιμοποιήστε τις σημειώσεις της SwissProt/UniProt (λειτουργική περιγραφή, EC αριθμούς, GO όρους, Pfam domains κ.λπ.) για να ελέγξετε αν αυτές οι πρωτεΐνες:
    - ανήκουν στην ίδια οικογένεια ή
    - έχουν παρόμοια βιολογική λειτουργία ή/και κοινά δομικά domains.
  - Τεκμηριώστε **3–5 χαρακτηριστικά παραδείγματα** όπου:
    - το BLAST δίνει χαμηλή ομοιότητα ακολουθίας (Twilight Zone),
    - οι embedding-based μέθοδοι τις φέρνουν κοντά στο χώρο των διανυσμάτων,
    - και οι λειτουργικές/δομικές σημειώσεις υποστηρίζουν την ύπαρξη ομολογίας.
  - Σχολιάστε επίσης περιπτώσεις πιθανών **false positives** (μικρή L2 αλλά χωρίς προφανή βιολογική συνάφεια) και πώς αυτές επηρεάζουν την ερμηνεία των αποτελεσμάτων.

ΕΙΣΟΔΟΣ & ΕΚΤΕΛΕΣΗ

1. Δεδομένα

- `swissprot.fasta`: Βάση δεδομένων πρωτεϊνών.
- `targets.fasta`: Πρωτεΐνες-στόχοι (queries).

2. Εκτέλεση

Θα παραδώσετε δύο εκτελέσιμα:

A. Generating Embeddings: `python protein_embed.py -i swissprot.fasta -o protein_vectors.dat`

B. Search Benchmark: `python protein_search.py -d protein_vectors.dat -q targets.fasta -o results.txt -method <all|lsh|hypercube|neural|ivf>`

ΕΞΟΔΟΣ (Αναφορά Αξιολόγησης)

Για κάθε πρωτεΐνη-ερώτημα (*query*) το πρόγραμμα πρέπει να παράγει έξοδο δύο επιπέδων:

- **[1] Συνοπτική σύγκριση μεθόδων**:
  - Για κάθε μέθοδο (Euclidean LSH, Hypercube, IVF-Flat, IVFPQ, Neural LSH) να εκτυπώνονται οι αναζητήσεις ανά δευτερόλεπτο (QPS) και η τιμή της μετρικής `Recall@N` σε σχέση με τα κορυφαία χτυπήματα του BLAST (Top-N).
- **[2] Αναλυτικοί Top-N γείτονες ανά μέθοδο**:
  - Για κάθε μέθοδο να εκτυπώνεται ένας πίνακας με τους Top-N γείτονες (π.χ. $N = 10$ για αναγνώσιμη εκτύπωση), με πληροφορίες για:
    - `Neighbor ID`
    - L2 απόσταση
    - `BLAST identity`
    - ένδειξη αν ο γείτονας ανήκει στο Top-N του BLAST (`In BLAST Top-N?`)
    - σύντομο βιολογικό σχόλιο όπου είναι σχετικό (`Bio comment`).
  - Το N που χρησιμοποιείτε στον υπολογισμό της `Recall@N` (π.χ. $N = 50$) πρέπει να αναφέρεται ρητά στην έξοδο.

Μορφή εξόδου Οι παρακάτω τιμές δεν είναι αντιπροσωπευτικές αλλά έχουν επιλεχθεί τυχαία.

```
Query Protein: <ID>
N = 50  (μέγεθος λίστας Top-N για την αξιολόγηση Recall@N)

[1] Συνοπτική σύγκριση μεθόδων
-----------------------------------------------------------------------
Method        | Time/query (s) | QPS  | Recall@N vs BLAST Top-N
-----------------------------------------------------------------------
Euclidean LSH | 0.020          | 50   | 0.92
Hypercube     | 0.030          | 33   | 0.88
Neural LSH    | 0.010          | 100  | 0.95
IVF-Flat      | 0.008          | 125  | 0.93
IVF-PQ        | 0.005          | 200  | 0.90
BLAST (Ref)   | 1.500          | 0.7  | 1.00   (ορίζει το Top-N)
-----------------------------------------------------------------------


[2] Top-N γείτονες ανά μέθοδο (εδώ π.χ. N = 10 για εκτύπωση)

Method: Euclidean LSH
Rank | Neighbor ID | L2 Dist | BLAST Identity | In BLAST Top-N? | Bio comment
-------------------------------------------------------------------------------
1    | <Prot_A>    | 0.15    | 22%            | Yes             | Remote homolog? (κοινή Pfam)
2    | <Prot_D>    | 0.16    | 19%            | No              | Πιθανό false positive
3    | <Prot_E>    | 0.17    | 25%            | Yes             | --
...
10   | <Prot_X>    | 0.25    | 12%            | No              | --


Method: Neural LSH
Rank | Neighbor ID | L2 Dist | BLAST Identity | In BLAST Top-N? | Bio comment
-------------------------------------------------------------------------------
1    | <Prot_B>    | 0.18    | 18%            | Yes             | Rem. Hom.? (GO description)
2    | <Prot_A>    | 0.19    | 22%            | Yes             | --
...
10   | <Prot_Y>    | 0.27    | 15%            | No              | --
```

Για τις περιπτώσεις που, με βάση την περαιτέρω ανάλυση στην αναφορά, θεωρείτε **υποψήφιες απομακρυσμένες ομόλογες** πρωτεΐνες, μπορείτε να χρησιμοποιείτε στο πεδίο `Bio comment` ενδεικτικές σημειώσεις (π.χ. «Remote homolog (ίδια οικογένεια, κοινό Pfam domain)»), ώστε να συνδέεται το ποσοτικό αποτέλεσμα με τη βιολογική ερμηνεία.

Παραδοτέα

1. Κώδικας Python (συμπεριλαμβανομένων των modules από προηγούμενες εργασίες, προσαρμοσμένα όπου χρειάζεται).
2. **Αναφορά (Report):**
   - Ποσοτική σύγκριση χρόνων εκτέλεσης και ακρίβειας μεταξύ των 5 μεθόδων (LSH, Hypercube, IVF-Flat, IVFPQ, Neural) με χρήση της μετρικής Recall@N και QPS.
   - **Ενότητα βιολογικής αξιολόγησης:**
     - Σαφής διατύπωση του πώς ορίζετε στην πράξη μία *remote homolog*.
     - Παρουσίαση και σχολιασμός 3–5 χαρακτηριστικών παραδειγμάτων όπου τα embeddings εντοπίζουν υποψήφιες απομακρυσμένες ομόλογες πρωτεΐνες που δεν εμφανίζονται ψηλά στα αποτελέσματα του BLAST, με αναφορά σε λειτουργικές σημειώσεις (UniProt), domains (π.χ. Pfam) και άλλες σχετικές πληροφορίες.
     - Συζήτηση των ορίων της προσέγγισης: περιπτώσεις όπου οι μέθοδοι ANN παράγουν βιολογικά αμφίβολους γείτονες (false positives) καθώς και πιθανές κατευθύνσεις βελτίωσης.

Επιπρόσθετες απαιτήσεις

1. **Οργάνωση Κώδικα**: Τα δύο προγράμματα πρέπει να είναι καλά οργανωμένα, κάνοντας χρήση ξεχωριστών modules.
2. **Αρχείο `readme.md`**: Το `readme.md` πρέπει να περιλαμβάνει: α) τίτλο/περιγραφή, β) κατάλογο αρχείων Python και περιγραφή τους, γ) οδηγίες εγκατάστασης εξαρτήσεων (`pip install -r requirements.txt` ή αντίστοιχα για `conda`), και δ) λεπτομερείς οδηγίες χρήσης/εκτέλεσης και για τα δύο σενάρια (`embed` και `search`).
3. **Αναφορά**: Η τεκμηρίωση της πειραματικής μελέτης (που θα περιλαμβάνει αναλυτική τεκμηρίωση για την επιλογή των υπερπαραμέτρων ανά μέθοδο στον χώρο των embeddings) αποτελεί ξεχωριστό παραδοτέο (markdown ή PDF).
4. **Git**: Η υλοποίηση θα πρέπει να γίνει με χρήση Git.