Telecom Nancy

Protein Subcellular Localization

Rapport de projet

Moh-Ouali Amrouni
Rayane BELKALEM

February 13, 2026

Contents

1 Contexte et objectif

L'objectif est de predire la localisation subcellulaire de proteines procaryotes (6 classes) a partir de sequences FASTA. Le projet impose une approche plus originale qu'un simple MLP, avec une architecture qui utilise des embeddings, une couche Transformer, puis un MLP de classification.

2 Contraintes et environnement

Le travail a ete realise sur un ASUS TUF 15, majoritairement en CPU. Cela impose des limites de memoire et un temps de calcul eleve pour l'extraction des embeddings. Pour rester dans ces contraintes:

- limitation de la longueur maximale (`max_len`) pour eviter les OOM ;
- utilisation de ProstT5 3Di en mode CPU avec *offload disque* ;
- cache local des modeles HF pour eviter les retelechargements ;
- batch sizes ajustes (ESM-C modere, ProstT5 a batch 1).

3 Donnees

Les donnees proviennent de DeepLocPro, fournies au format FASTA. Le header suit le format:

```
>PROTEIN_ID|LOCATION|GRAM_TYPE|PARTITION
```

Nous generons un fichier `metadata.csv` puis des splits `train/val/test`. Par defaut, les splits sont stratifies, avec une option pour clustering par identite si MMseqs2 est installe.

4 Pipeline de traitement

1. **Preparation des metadonnees** a partir du FASTA.
2. **Creation des splits** leakage-safe.
3. **Extraction des embeddings**:
   - ESM-C (300M) pour un embedding principal (dimension 960).
   - ProstT5 3Di (dimension 1024) comme second embedding.
4. **Entrainement** du modele Transformer + BiLSTM + MLP.
5. **Evaluation** sur le split test (Accuracy, F1 macro, MCC).

5 Choix des embeddings et reduction du cout

Le sujet impose l'utilisation conjointe de ESM-C et ProstT5 3Di. ProstT5 etant couteux sur CPU, nous avons active l'*offload* disque et adopte un batch de 1 pour stabiliser l'extraction.

Pour limiter le temps de calcul tout en gardant une couverture elevee du dataset, `max_len` est fixe a 1000. Cette valeur conserve la grande majorite des sequences, tout en restant praticable sur une machine CPU.

6 Architecture du modele

Le modele propose une architecture plus originale que l'article: *embeddings* $\rightarrow$ *Transformer* $\rightarrow$ *BiLSTM* $\rightarrow$ *MLP*.

- projection lineaire des deux embeddings vers une dimension commune ;
- traitement des modalites comme tokens d'un Transformer encoder ;
- passage dans un BiLSTM pour modeliser les interactions ordonnees entre tokens ;
- fusion `CLS + attention-pooling LSTM` puis MLP de classification.

Configuration principale

Parametre	Valeur
Embedding 1	ESM-C 300M (dim 960)
Embedding 2	ProstT5 3Di (dim 1024)
Pooling	meanpool
max_len	1000
ESM batch	16
Embed2 batch	1

Table 1: Configuration principale pour l'extraction et l'entrainement.

7 Entrainement

La perte utilisee est la *focal loss* avec un echantillonnage equilibre pour gerer l'imbalance des classes. L'entrainement utilise un early stopping base sur le F1 macro.

8 Exploration des donnees (EDA)

Le dataset contient 11 906 sequences. Les colonnes principales sont: `protein_id`, `sequence`, `label`, `gram_type`, `partition`, `split`, `seq_len`. Aucune valeur manquante n'a ete observee.

Distribution des classes

Classe	Nombre
Cytoplasmic	6885
Cytoplasmic Membrane	2535
Extracellular	1077
Outer Membrane	756
Periplasmic	566
Cell Wall	87

Table 2: Distribution des classes (EDA).

Statistiques de longueur

La longueur des sequences est tres variable (distribution a longue traine). Statistiques principales:

- moyenne: 438.4, ecart-type: 289.0
- min: 8, mediane: 399, max: 5627
- P10: 146, P25: 264, P75: 557, P90: 787, P95: 878, P99: 1296

Pour des tests rapides sur CPU, un `max_len` entre 600 et 1000 est recommande. La valeur `max_len=1000` conserve la majorite des sequences tout en restant calculable.

Repartition des splits

Train: 8333, Val: 1191, Test: 2382.

Graphiques

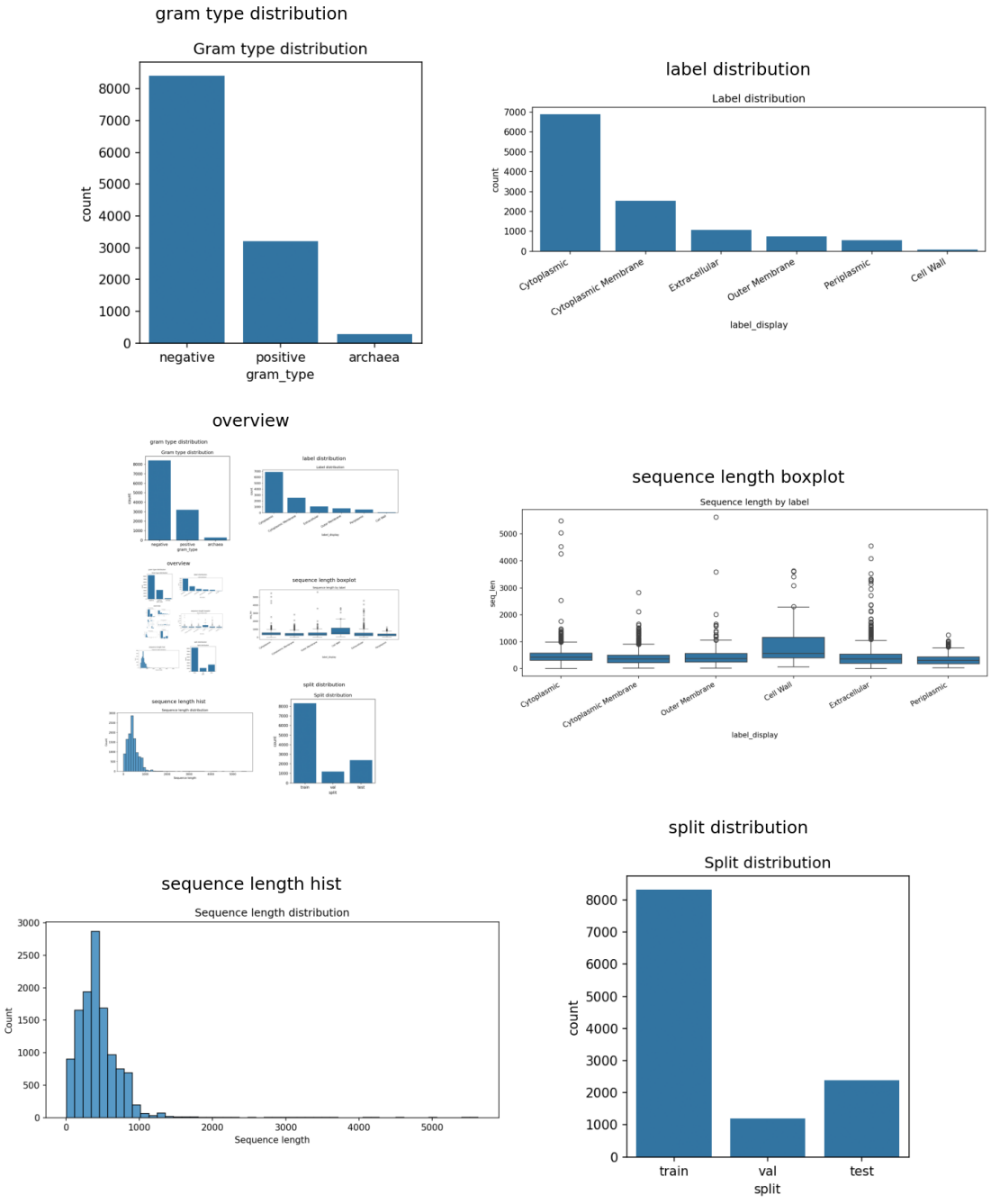

Figure 1: Vue d'ensemble: labels, longueurs, et splits.

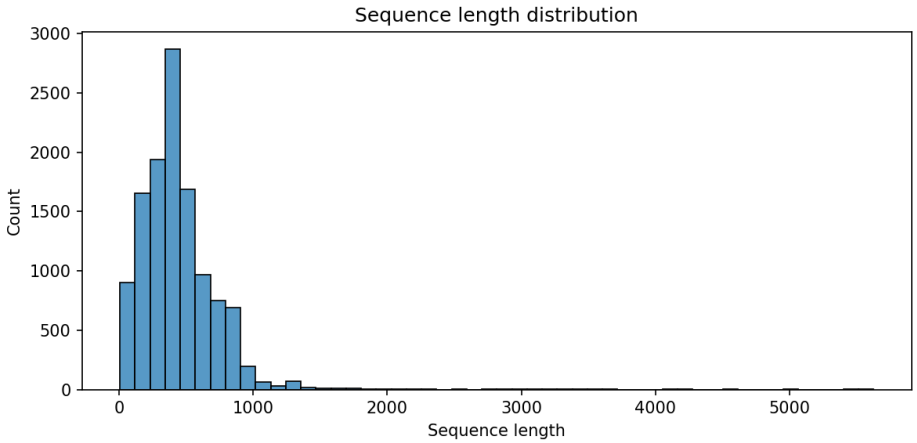

Figure 2: Distribution des longueurs de sequences.

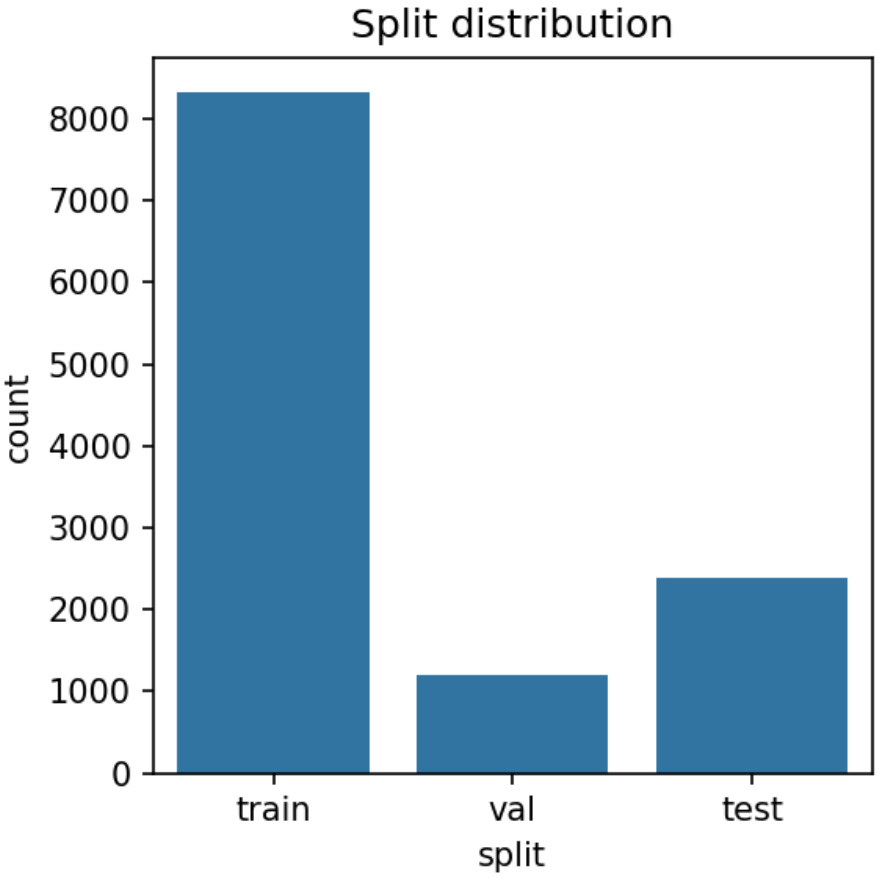

Figure 3: Repartition train/val/test.

9 Resultats

Run utilise: ESM-C + ProstT5 3Di (meanpool), `max_len=1000`, execution CPU.

Metriques globales

Run	Accuracy	F1 macro	MCC
Evaluation	0.8741	0.8005	0.8227

Table 3: Metriques globales sur le split test.

Resultats par classe

Classe	Precision	Recall	F1
Cytoplasmic	0.901	0.846	0.873
Cytoplasmic Membrane	0.500	0.750	0.600
Periplasmic	0.933	0.925	0.929
Outer Membrane	0.776	0.839	0.806
Extracellular	0.706	0.818	0.758
Cell Wall	0.857	0.818	0.837

Table 4: Precision/Recall/F1 par classe (split test).

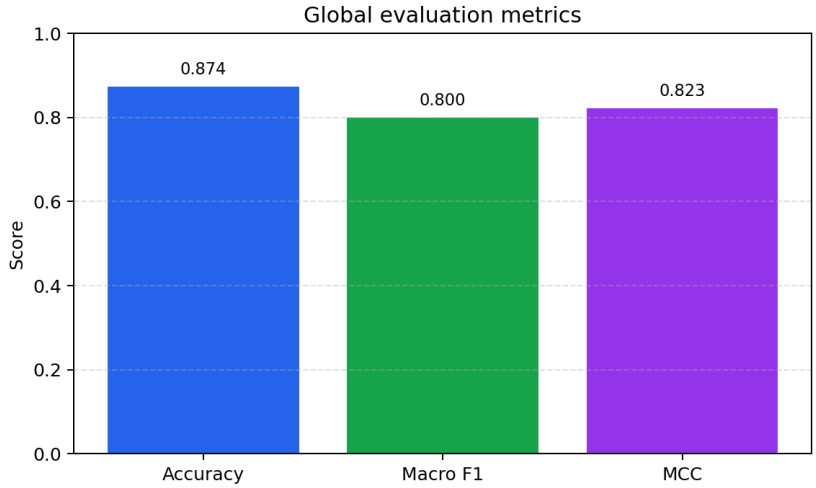

Figure 4: Metriques globales.

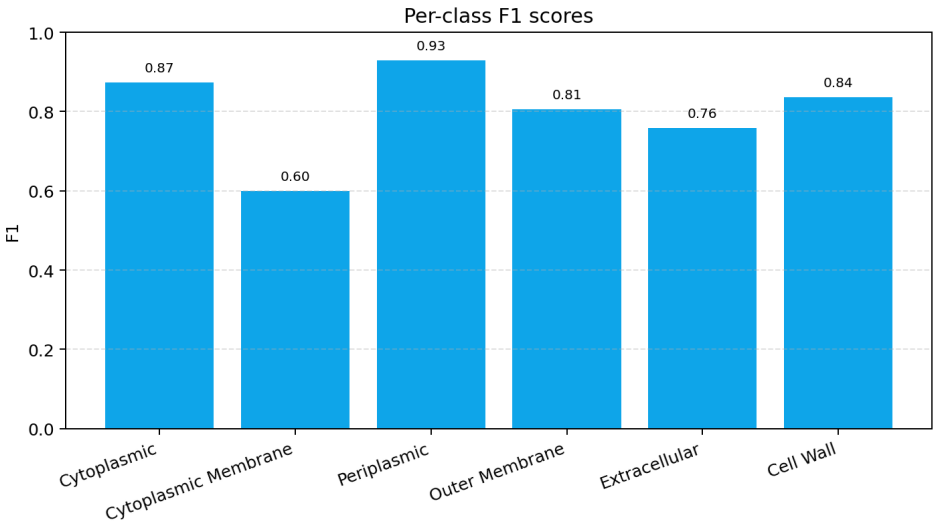

Figure 5: F1 par classe.

Figure 6: Matrice de confusion **approximative** reconstruite a partir des metriques agregees du rapport (ce n'est pas la matrice brute issue des predictions).

10 Limites et pistes d'amelioration

- Monter `max_len` et/ou utiliser un GPU pour integrer plus d'information de sequence.

- Tester une variante sans BiLSTM pour mesurer son apport exact.
- Augmenter la profondeur du Transformer ou tester un pooling different (mean/cls/both).
- Etendre l'evaluation avec validation croisee stricte type GraphPart.

11 Reproductibilite

Commandes principales (extraits):

```
python scripts/prepare_metadata.py --fasta data/raw/graphpart_set.fasta --output data
python scripts/prepare_splits.py --metadata data/processed/metadata.csv --output data
python -m src.embeddings.fetch_embeddings --embed2_backend prostt5 --max_len 1000 --p
python scripts/train.py --config configs/default.yaml
python scripts/evaluate.py --checkpoint results/checkpoints/best_model.pt --config co
make eda
make results-figures
```

12 Conclusion

Le pipeline complet est operationnel avec des contraintes CPU fortes. L'architecture respecte la contrainte *embeddings + Transformer + MLP*. Les resultats actuels sont solides compte tenu du materiel disponible.