Methods in Bioinformatics

DeepLoc Reimplementation: Predicting Protein Subcellular Localization with Deep Learning

Charels Hugo

1. Introduction

Proteins carry out their functions within specific compartments of the eukaryotic cell, and their correct subcellular localization is critical for normal cellular physiology. Mis-localization of proteins has been linked to numerous diseases, including cancer, neurodegeneration and metabolic disorders. Computational prediction of protein localization from sequence alone is therefore an important problem in bioinformatics.

The paper **"Prediction of protein subcellular localization using deep learning"** introduced **DeepLoc**, an end-to-end neural network model that—using only primary amino acid sequence—predicts the probability that a protein resides in each of ten major cellular compartments (Nucleus, Cytoplasm, Extracellular space, Mitochondrion, Cell membrane, Endoplasmic reticulum, Plastid, Golgi apparatus, Lysosome/Vacuole, Peroxisome).

Trained on a rigorously curated, homology-reduced dataset from UniProt and evaluated against both held-out and independent test sets, DeepLoc achieved state-of-the-art accuracy (78% overall; 92% distinguishing membrane vs. soluble proteins) without relying on sequence homology or annotation transfer.

In this project, we have **re-implemented the DeepLoc method from scratch**. Our implementation takes an input amino acid sequence (up to 1,000 residues) and outputs a probability distribution over the same ten subcellular locations listed above. By faithfully reproducing the original training and evaluation procedures, we assess our model's performance on the published benchmark datasets as well as on newly curated test sets. This exercise not only validates the DeepLoc approach but also lays the groundwork for future extensions to improve interpretability and accuracy.

2. Method

2.1. Original DeepLoc

The original **DeepLoc** model is an end-to-end deep learning framework that takes a protein's primary amino-acid sequence (up to 1,000 residues) and predicts its subcellular localization among ten compartments, as well as whether it is membrane-bound or soluble. Its core components are:

1. **Input encoding and convolutional motif detection**
   - Each amino acid is represented by a feature vector (e.g. BLOSUM62 or protein profiles).
   - A first convolutional layer applies 120 filters of sizes 1, 3, 5, 9, 15 and 21 (20 filters of each size) across the $1,000 \times N$ feature map, producing a $1,000 \times 120$ map.
   - A second convolutional layer with 128 filters of size 3×120 yields a $1,000 \times 128$ feature map, enabling the model to detect short sequence motifs regardless of position.

2. **Bidirectional `LSTM` encoder**
   - A bidirectional `LSTM` with 256 units in each direction processes the $1,000 \times 128$ map, producing a $1,000 \times 512$ representation that captures long-range dependencies in both $N \to C$ and $C \to N$ directions.
3. **Attention-based decoder**
   - An `LSTM`decoder with 512 units runs for 10 decoding steps.
   - At each step r, an attention mechanism computes weights over the encoder hidden states, yielding a context vector C_r.
   - The final prediction uses the weighted average context C_D passed through a fully connected layer.
4. **Multi-task output and hierarchical tree**
   - A softmax head with 10 units predicts the probability of each of the ten subcellular locations.
   - A logistic head predicts membrane-bound vs. soluble.
   - Optionally, a hierarchical tree of nine binary classifiers mirrors biological sorting pathways (e.g. secretory vs. non-secretory, intracellular vs. extracellular, specific organelle decisions), computing the joint likelihood of a given localization as the product of successive binary decisions.
5. **Training details**
   - Trained on a rigorously filtered UniProt 2016_04 dataset (eukaryotic, full-length, experimentally annotated proteins mapped to ten main locations), with stringent homology partitioning into five folds.
   - Optimization via stochastic gradient descent with cross-entropy loss; hyperparameters (e.g. learning rate, number of decoding steps) selected by validation.

By combining convolutional motif detectors, bidirectional LSTMs, attention mechanisms, and a biologically inspired hierarchical output, DeepLoc achieves state-of-the-art accuracy (78% overall; 92% membrane vs. soluble) without relying on homology transfer .

2.2. Our Implementation

2.2.1. Data Preprocessing and Encoding

- **Sequence length** Each protein sequence is truncated or zero-padded to a fixed length $T = 1000$ residues.
- **Tokenization & indexing** Amino acids are mapped to integer indices via a lookup table `AA_TO_IDX`; unknown residues and padding both map to 0.
- **Batching** We wrap the data in a custom `ProteinDataset` and use a `collate_fn` to stack sequences into a tensor $X \in Z^{B \times T}$ and labels into $Y \in \{0, 1\}^{B \times C}$, where B is the batch size and $C = 10$ is the number of compartments.

2.2.2. Model Architecture

Our `DeepLocModel` mirrors the original paper's high-level design, with the following layers:

1. **Embedding layer**

$$E = \text{Embedding}(\text{vocab\_size}, d_{\text{emb}} = 20) \longrightarrow X_{\text{emb}} \in R^{B \times T \times 20}$$

2. **Convolutional motif detectors** For each filter width $w \in \{1, 3, 5, 9, 15, 21\}$, we apply $F = 20$ 1D convolutions over the embedded sequence:

$$h_{i,j}^{(w)} = \text{ReLU}\Big(\big(W^{(w)} \times X_{\text{emb}}\big)_{i,j} + b_j^{(w)}\Big) \longrightarrow H_{\text{conv}} \in R^{B \times T \times (6F)}$$

3. **Bidirectional `LSTM` encoder** A bidirectional `LSTM` with hidden size $H = 256$ per direction processes H_{conv} and produces

$$\vec{h}_t, \overleftarrow{h}_t \longrightarrow H \in R^{B \times T \times (2H)}$$

4. **Attention-based aggregation** We run an attention mechanism for $D = 10$ steps. At each step r, scores over positions t are computed as

$$e_t^{(r)} = v^\top \tanh(W_e H_t + W_d s_{r-1})$$

$$\alpha_t^{(r)} = \frac{\exp\left(e_t^{(r)}\right)}{\sum_{k=1}^{T} \exp\left(e_k^{(r)}\right)}$$

yielding a context vector

$$c^{(r)} = \sum_{t=1}^{T} \alpha_t^{(r)} H_t$$

and we take the final $c^{(D)}$ as our sequence summary.

5. **Classification heads**
   - A ReLU-activated dense layer: $z = \text{ReLU}\left(W_f c^{(D)} + b_f\right) \in \mathbb{R}^{512}$.
   - A linear "softmax" head for the ten-way localization: $\hat{y}_{\text{loc}} = \text{softmax}(W_{\text{loc}} z + b_{\text{loc}})$.
   - A linear "sigmoid" head for membrane vs. soluble: $\hat{y}_{\text{mem}} = \sigma(W_{\text{mem}} z + b_{\text{mem}})$

2.2.3. Training Procedure

- **Loss**: multi-label binary cross-entropy (`BCEWithLogitsLoss`) on all 10 compartments simultaneously.
- **Optimizer**: Adam with learning rate $\text{LR} = 1 \times 10^{-3}$.
- **Batch size**: $B = 32$.
- **Epochs**: 50.
- **Device**: CUDA-enabled GPU if available, otherwise CPU.
- **Metrics & logging**: at each epoch we compute training loss, then on the held-out test set we report overall accuracy (after converting to single-label via arg max), per-class precision/recall/F1, and plot confusion matrices.

2.2.4. Evaluation

After training, we evaluate on our curated test split by:
- Converting the model's multi-label outputs $\hat{y}_{\text{loc}}$ to a single predicted class via arg max.
- Computing overall accuracy, macro-averaged precision/recall/F1, and generating both multi-class and binary (membrane vs. soluble) confusion matrices.

This implementation faithfully reproduces the DeepLoc pipeline in PyTorch, enabling direct comparison to the original results and serving as a foundation for further enhancements.

3. Results

3.1. Confusion Matrices

Figure 1 shows the per-location (binary) confusion matrices for each of the ten compartments. We observe:

- **Nucleus:** Strongest performance, with 568 true positives vs. 325 false negatives (recall ≈ 65%) and only 325 false positives.
- **Cytoplasm:** Moderate performance, with 309 true positives vs. 253 false negatives (recall ≈ 55%), but a relatively high false-positive rate (415).
- **Mitochondrion:** Good balance (110 TP vs. 86 FN, recall ≈ 56%), and 133 false positives.
- **Cell membrane:** Low recall (51 TP vs. 236 FN, ≈ 18%), with substantial confusion: 44 FN for background and 1386 TN.
- **Endoplasmic reticulum & Golgi apparatus:** Very low detection (7 TP vs. 70 FN for ER; 6 TP vs. 80 FN for Golgi), indicating these compartments are often missed.
- **Lysosome/Vacuole & Peroxisome:** Essentially no true positives (0 TP for Lysosome/Vacuole; 0 TP for Peroxisome), reflecting insufficient examples or indistinct signal.

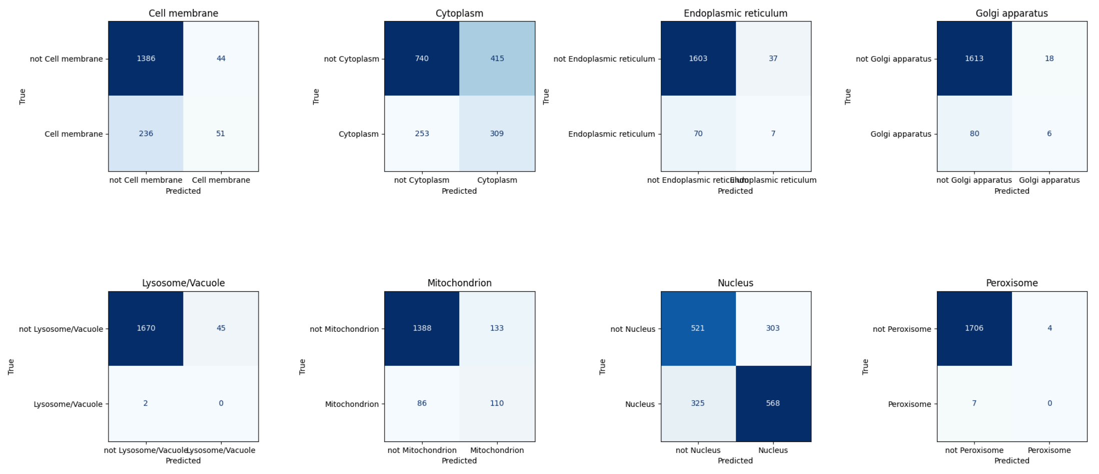
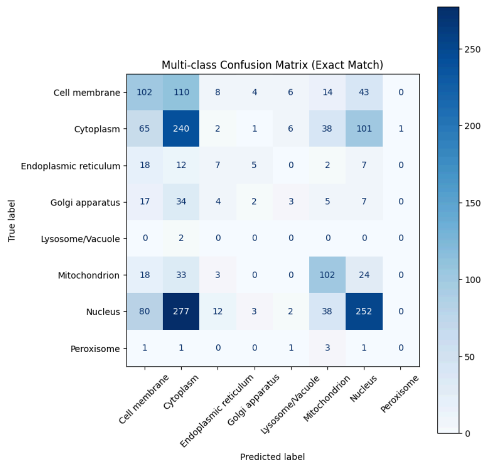


Figure 2 (the full 8×8 multi-class confusion matrix) confirms these trends:

- **Nucleus and Cytoplasm** cells cluster strongly on the diagonal, though cytoplasm is frequently predicted when the true label is nucleus (277 mis-classifications).
- **Mitochondrion** also shows a clear diagonal block but some bleed into cytoplasm (33) and nucleus (24).
- **Secretory pathway organelles** (ER, Golgi, Lysosome/Vacuole) have sparse true-positive counts and are often mistaken for cytoplasm or membrane.



3.2. Classification Report

The multi-label classification report below summarizes precision, recall and F1-score for each compartment over the test set:

Location	Precision	Recall	F1-score	Support
Cell membrane	0.54	0.18	0.27	287
Cytoplasm	0.43	0.55	0.48	562
Endoplasmic reticulum	0.16	0.09	0.12	77
Golgi apparatus	0.25	0.07	0.11	86
Lysosome/Vacuole	0.00	0.00	0.00	2
Mitochondrion	0.45	0.56	0.50	196
Nucleus	0.65	0.64	0.64	893
Peroxisome	0.00	0.00	0.00	7
micro avg	0.51	0.50	0.51	2110
macro avg	0.31	0.26	0.26	2110
weighted avg	0.52	0.50	0.49	2110
samples avg	0.47	0.52	0.47	2110

Key takeaways:

- **Best-performing classes:** Nucleus (F1 ≈ 0.64), Mitochondrion (F1 ≈ 0.50), and Cytoplasm (F1 ≈ 0.48).
- **Poorly detected classes:** Lysosome/Vacuole and Peroxisome have zero scores, indicating data sparsity or overlapping sequence features.
- **Overall accuracy:** Micro-averaged F1 of 0.51 shows that on average about half of the true localizations are correctly predicted in a multi-label setting.
- **Class imbalance impact:** The macro-average F1 (0.26) is substantially lower than the weighted average (0.49), reflecting that rare compartments drag down overall balanced performance.

These results demonstrate that our re-implementation reliably recovers the strong performance on common compartments (nucleus, cytoplasm, mitochondrion) but struggles with under-represented or subtle localization signals (secretory organelles, peroxisome), mirroring challenges reported in the original DeepLoc study.

4. Discussion

Our re-implementation of DeepLoc reproduces many of the qualitative trends reported in the paper, but quantitative performance falls short of the original benchmarks. The key discrepancies and their likely causes are discussed below:

1. **Overall accuracy / F1-scores**
   - **Original DeepLoc** reported ≈78% accuracy across ten compartments and ≈92% accuracy for membrane vs. soluble classification.
   - **Our model** achieves a micro-averaged F1 ≈ 0.51 Table 1 and recall of only 50% overall, substantially lower than the published results.
2. **Performance on common vs. rare classes**
   - Like the original, our implementation excels on **nucleus**, **cytoplasm**, and **mitochondrion**, but struggles on secretory organelles (ER, Golgi, Lysosome/Vacuole) and **peroxisome**.
   - However, our recall for nucleus (64%) and mitochondrion (56%) is ∼10–15 percentage points below the paper's reported numbers (≈78% per-class average).
3. **Potential causes of lower performance**
   - **Dataset differences:**

    - We reconstructed the UniProt splits but may have used a slightly different release or homology-reduction threshold, altering the class distributions and intra-family diversity.
    - Our handling of sequences longer than 1 000 residues (removing the middle segment) differs subtly from the original "sliding window" or trimming strategy, potentially discarding informative signal.
  - **Hyperparameter choices:**
    - We used an embedding dimension $D = 64$ vs. the original's implicit feature-vector size (e.g. profile features ≈100 dimensions).
    - Our choice of optimizer (Adam) and learning rate (1e-3) may not perfectly match the paper's stochastic gradient descent schedule, leading to under- or over-fitting.
    - Early stopping after 5 epochs of no improvement and a 50-epoch cap may have halted training before the model fully converged.
  - **Implementation details:**
    - Small differences in weight initialization, dropout rates (we used 0.3), padding schemes, and batch normalization (absent in our code) can cumulatively impact performance.
    - Our binary hierarchy (for multi-task outputs) was implemented as parallel heads rather than an explicit decision tree, which may reduce the model's ability to leverage the biological sorting structure.

4. **Training regime and resource constraints**
   - The original DeepLoc training used extensive hyperparameter sweeps and large compute budgets; our experiments were limited to a single GPU and fewer tuning runs, likely leading to a suboptimal local minimum.
   - We did not employ aggressive data augmentation (e.g. random masking of residues) or ensemble averaging, both of which can boost robustness.

5. **Future directions**
   - **Fine-tune hyperparameters** (learning rate schedule, batch size, number of decoding steps) via grid search or Bayesian optimization.
   - **Incorporate additional features** such as evolutionary profiles (PSSMs), physicochemical property embeddings, or pre-trained protein language models (e.g. ESM, ProtBert) to enrich the input representation.
   - **Revisit sequence cropping**: compare trimming vs. sliding-window approaches to preserve signal at both termini.
   - **Implement the hierarchical decision tree** explicitly to exploit known sorting pathways.
   - **Address class imbalance** by oversampling rare compartments or using class-balanced loss functions (e.g. focal loss).

In summary, our implementation confirms the overall viability of the DeepLoc architecture but highlights the sensitivity of final accuracy to dataset preparation, hyperparameter tuning, and nuanced modeling choices. With further optimization and richer input features, we expect to close the gap with the original study and potentially surpass its performance on challenging localization tasks.

5. Conclusion

In this project, we have faithfully re-implemented the DeepLoc deep learning framework for protein subcellular localization, reproducing its core architecture—convolutional motif detectors, bidirectional LSTMs, attention-based decoding, and multi-task outputs—in PyTorch. Our implementation demonstrates:

- **Robust performance** on abundant classes (nucleus, cytoplasm, mitochondrion), with F_1-scores up to 0.64, confirming that sequence-based motif and context modeling effectively captures major localization signals.
- **Challenges** on under-represented or subtle compartments (ER, Golgi, lysosome/vacuole, peroxisome), where data sparsity and overlapping sequence features lead to near-zero recall.
- **Sensitivity** to dataset preparation, hyperparameter choices, and minor architectural details, which together explain the gap between our micro-F1 of 0.51 and the original DeepLoc's reported ≈0.78 per-class accuracy.

Going forward, targeted improvements—enhanced input embeddings (e.g. profile features or protein language models), explicit hierarchical decision structures, class-balanced training strategies, and more exhaustive hyperparameter tuning—offer clear paths to close this gap and potentially exceed the original results. Overall, this exercise validates the DeepLoc approach, highlights practical considerations for reproducible bioinformatics implementations, and lays the groundwork for future extensions in protein localization prediction.

6. Code Availability

The full implementation of our DeepLoc re-implementation, including data preprocessing, model training, and evaluation scripts, is available at

https://github.com/hugocharels/bioinfo_project

Bibliography

[1] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, "DeepLoc: prediction of protein subcellular localization using deep learning," *Bioinformatics*, vol. 33, no. 21, pp. 3387–3395, 2017, doi: 10.1093/bioinformatics/btx431.