Predicting Human Non-TATA Promoters from DNA Sequence with a Convolutional Neural Network

Klas Holmgren

13 June 2026

Abstract

I trained a convolutional neural network (CNN) to classify 251 bp human DNA sequences as non-TATA promoters versus non-promoters, using sequence alone. The model reaches **90.5% accuracy and 0.969 AUROC** on a held-out test set, matching or exceeding the published CNN baseline for this benchmark. Beyond raw performance, I show that the network's first-layer convolutional filters—its learnable "motif detectors"—recover biologically meaningful DNA patterns, including the **GC-box (Sp1/KLF) motif** and CpG-rich core-promoter elements that are hallmarks of TATA-less human promoters. This demonstrates that a small, fast CNN not only predicts regulatory function accurately but also learns interpretable, biologically grounded features.

1 Background and motivation

Gene transcription begins at *promoters*—regulatory DNA regions near transcription start sites where RNA polymerase and transcription factors assemble. Classically, many promoters contain a *TATA box*, but a large fraction of human promoters are *TATA-less* and instead rely on GC-rich elements and CpG islands. Identifying promoters directly from sequence is a long-standing problem in regulatory genomics.

Deep learning is well suited to this task because regulatory signals are encoded as short, position-flexible *sequence motifs*. Convolutional networks were introduced to genomics for exactly this reason [1, 2, 3]: a 1D convolution scanning a one-hot-encoded sequence is mathematically analogous to scanning with a *position weight matrix* (PWM), the standard representation of a transcription-factor binding motif. This project reproduces that idea on a clean benchmark and emphasizes *interpretability*: not just *whether* the model predicts well, but *what it learns*.

2 Data

I use the `human_nontata_promoters` dataset from Genomic Benchmarks [4], a curated suite designed for benchmarking deep models on genomics.

Property	Value
Sequence length	251 bp
Alphabet	A, C, G, T (one-hot)
Classes	promoter (positive) / non-promoter (negative)
Train	27,097 (14,742 pos / 12,355 neg)
Test	9,034 (4,915 pos / 4,119 neg)
Class balance	∼54% positive (roughly balanced)

Table 1: Dataset summary.

Encoding. Each sequence is converted to a 4×251 one-hot matrix, with channels ordered (A, C, G, T)—exactly the input layout a `Conv1d(in_channels=4)` expects. Ambiguous bases (e.g. `N`) map to an all-zero column. Encoded arrays are cached as compressed `.npz` for fast reuse. A 10% validation split is carved from the training set for model selection; the test set is held out entirely until final evaluation.

3 Methods

3.1 Architecture

The model (`PromoterCNN`, ∼108k parameters) follows the DeepBind/Basset design:

Stage	Layer
Input	one-hot 4×251
Motif scan	Conv1d($4 \to 128$, k=19, same-pad) → BN → ReLU → MaxPool(4)
Combine motifs	Conv1d($128 \to 64$, k=11, same-pad) → BN → ReLU
Pooling	AdaptiveMaxPool1d(1) ("is the motif present anywhere?")
Head	Dropout → Linear($64 \to 128$) → ReLU → Dropout → Linear($128 \to 1$)
Output	logit → sigmoid

Table 2: Network architecture.

Two choices are deliberate and central to the interpretability story:

- **First conv = motif detector.** Each of the 128 first-layer filters (width 19 bp) learns to recognize a short sequence pattern. Same-padding keeps activations aligned with input positions, so a filter's activation at position i corresponds to a known input window.
- **Global max-pooling** asks, per filter, *"does this motif occur anywhere in the sequence?"*—the correct inductive bias for regulatory elements whose exact position varies, and one that makes the model robust to *where* the signal sits.

3.2 Training

Binary cross-entropy on a single logit (`BCEWithLogitsLoss`); AdamW (lr = 10^{-3}, weight decay 10^{-5}); reduce-on-plateau scheduler on validation AUROC; best checkpoint selected by validation AUROC over 20 epochs. Hardware: NVIDIA RTX 4070, **∼0.5 s/epoch** (∼10 s total). The code is device-agnostic and falls back to CPU automatically.

4 Results

4.1 Headline metrics (held-out test set)

Metric	Value
Accuracy	**0.905**
AUROC	**0.969**
Promoter precision	0.890
Promoter recall (sensitivity)	0.941
Promoter F1	0.915
Specificity (non-promoter recall)	0.861

Table 3: Test-set performance. Results match or exceed the published CNN baseline (∼0.85 accuracy) for this benchmark [4].

4.2 ROC curve and confusion matrix

The model separates the classes cleanly (Fig. 1), with a steep rise indicating strong performance even at conservative thresholds. The confusion matrix (Fig. 2) shows mildly asymmetric errors: few missed promoters (289 false negatives) but somewhat more false positives (572). This higher sensitivity than specificity is reasonable for a screening-style task and is tunable along the ROC curve.

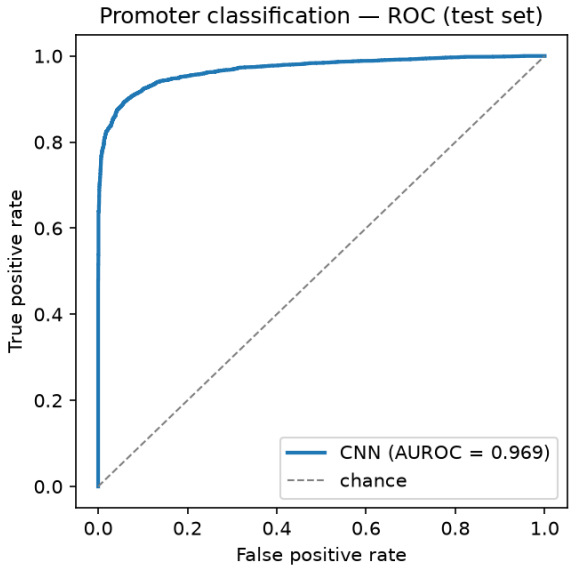

Figure 1: ROC curve on the test set (AUROC = 0.969).

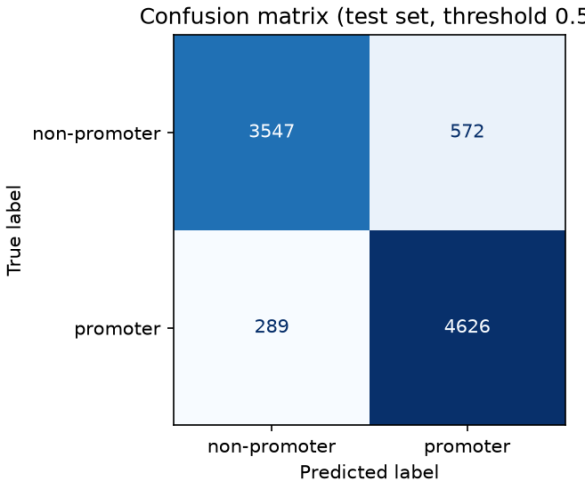

Figure 2: Confusion matrix at threshold 0.5.

5 Interpretability: what did the model learn?

A predictive model is more convincing if its learned features are biologically meaningful. I recover motifs from the trained first-layer filters using the standard DeepBind/Basset procedure: (1) run

many real sequences through the first convolution; (2) for each filter, collect the input windows that activate it most strongly; (3) stack those windows into a *position frequency matrix* (PFM); (4) render each PFM as an information-content *sequence logo* (letter height in bits); (5) rank filters by total information content to surface the sharpest, most specific motifs.

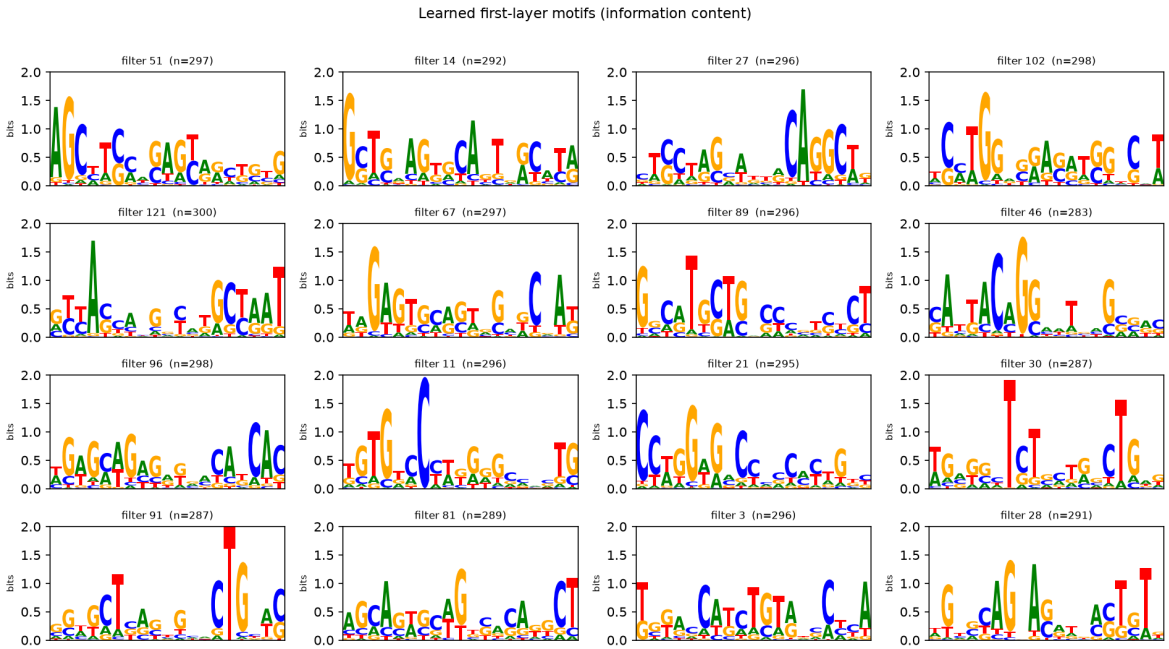

Figure 3: The 16 most informative learned first-layer filters, rendered as information-content sequence logos. Many are crisp, GC-rich patterns; several show clear GC-box (`GGGGCGGGG`) motifs, the canonical Sp1/KLF transcription-factor binding site.

Findings. The most informative filters are crisp and well-structured (approaching the 2-bit maximum at key positions). They are dominated by **G/C-rich and CpG patterns**, including clear **GC-box motifs** (`GGGGCGGGG`)—the canonical binding site of the Sp1/KLF family of transcription factors. This is exactly what biology predicts: non-TATA human promoters are GC-rich and frequently overlap CpG islands, where Sp1/KLF and related factors nucleate transcription. In other words, the network independently rediscovered known regulatory grammar *purely from labeled sequence*, with no motif supervision.

6 Discussion and limitations

Strengths. A compact (∼108k-parameter) CNN achieves strong accuracy in seconds of training, and its learned features are directly interpretable as biological motifs—a clean illustration of why CNNs became the workhorse of regulatory genomics.

Limitations.

- *Single benchmark.* Results are specific to `human_nontata_promoters`; generalization to other promoter definitions, species, or assays is untested.
- *No strand symmetry.* DNA is double-stranded, but the model sees only one strand. A reverse-complement-aware model or augmentation would likely improve robustness.

- *Motifs are unnamed.* The logos are visually consistent with GC-boxes, but I have not formally matched them to a database (e.g. JASPAR via TomTom) to assign identities with statistical confidence.
- *Benchmark caveats.* Negative sets can carry compositional biases; high accuracy may partly reflect base-composition differences rather than precise motif logic.

7 Conclusion and future work

A small convolutional network predicts human non-TATA promoters from sequence at **0.905 accuracy / 0.969 AUROC**, and its first-layer filters recover the GC-rich, Sp1/KLF-style motifs characteristic of TATA-less promoters. The project demonstrates an end-to-end, reproducible pipeline—encoding, modeling, training, evaluation, and interpretation—on real genomic data. Future directions: (1) reverse-complement augmentation or a strand-equivariant architecture; (2) formal motif identification by matching learned PFMs to JASPAR (TomTom); (3) attribution methods (saliency, Integrated Gradients, DeepLIFT) for per-base importance; (4) fine-tuning a DNA language model (DNABERT-2, Nucleotide Transformer) for comparison; (5) cross-task evaluation on other Genomic Benchmarks datasets.

References

[1] Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* 33, 831–838.

[2] Kelley, D.R., Snoek, J., Rinn, J.L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research* 26, 990–999.

[3] Zhou, J., Troyanskaya, O.G. (2015). Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* 12, 931–934.

[4] Grešová, K., Martinek, V., Čechák, D., Šimeček, P., Alexiou, P. (2023). Genomic Benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data* 24, 25.

A Reproducing this report

```
cd ~/genomics-dl && source .venv/bin/activate
uv pip install -r requirements.txt
python -c "from genomic_benchmarks.loc2seq import download_dataset; \
          download_dataset('human_nontata_promoters', version=0)"
python src/data.py                                # build encoded caches
python src/train.py --epochs 20                   # -> outputs/best_model.pt
python src/report_figs.py                         # -> roc_curve.png, confusion_matrix.png
python src/interpret.py --top 16 --n-seqs 4000  # -> motifs.png
pdflatex report.tex && pdflatex report.tex      # -> report.pdf
```

Environment: Python 3.12, PyTorch 2.12 (CUDA 13), NVIDIA RTX 4070. Exact versions pinned in `requirements.txt`.