GET: A foundation model of transcription across human cell types

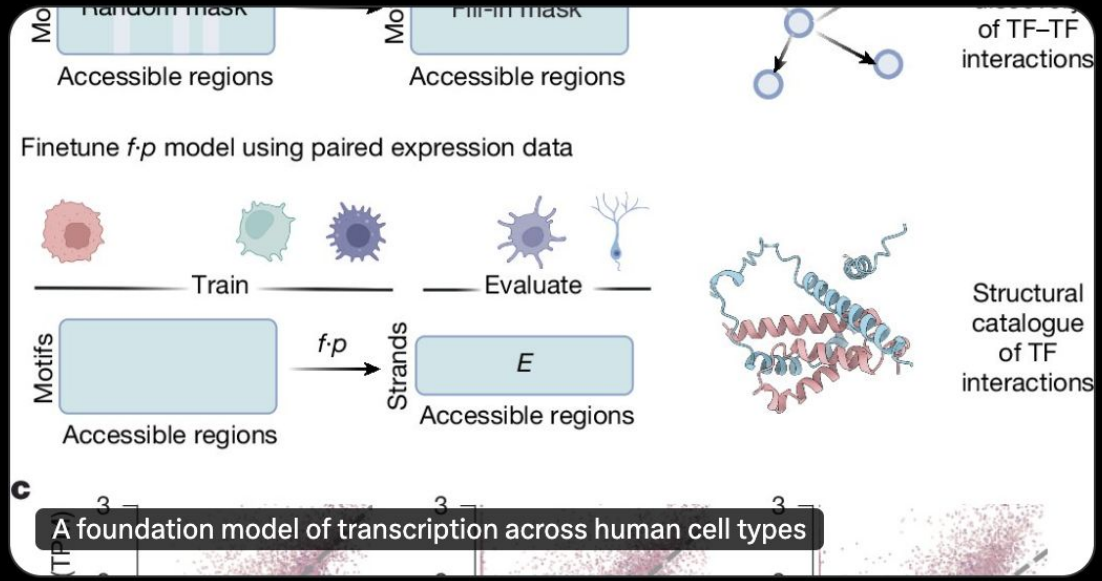
(Fu et al. 2025)

Celine Hoh

02/11 Deep learning reading group

Steven Salzberg 💙💛 @StevenSalzberg1 · Jan 9

Is this real, or is it excessive AI hype? New paper claims that an AI foundation model achieves experimental-level accuracy in predicting gene expression even in previously unseen cell types." I'll read the paper, but don't believe it for a second



From nature.com

13 65 440 61K

Background

Challenge: Predict gene expression from transcription factor information

Past models not good enough because:

- Expecto, Basenji and Enformer
  - can only predict gene expression of the cell types they were trained on

- Geneformer, scGPT and scFoundation
  - Works on single cell, do not learn deep underlying transcriptional grammar

Quick bio review

🧬 Promoters → Gene On/Off Switch

✨ Enhancers → Boosts Gene Expression

🧑‍🔬 Transcription Factors (TFs) → Messengers that bind at promoters and enhancers

GET

Goal: Predict gene expression from chromatin information

GET (general expression transformer)
- Input: Chromatin data from 213 fetal + adult cell types
- Output: Gene expression prediction

Why is GET so cool:

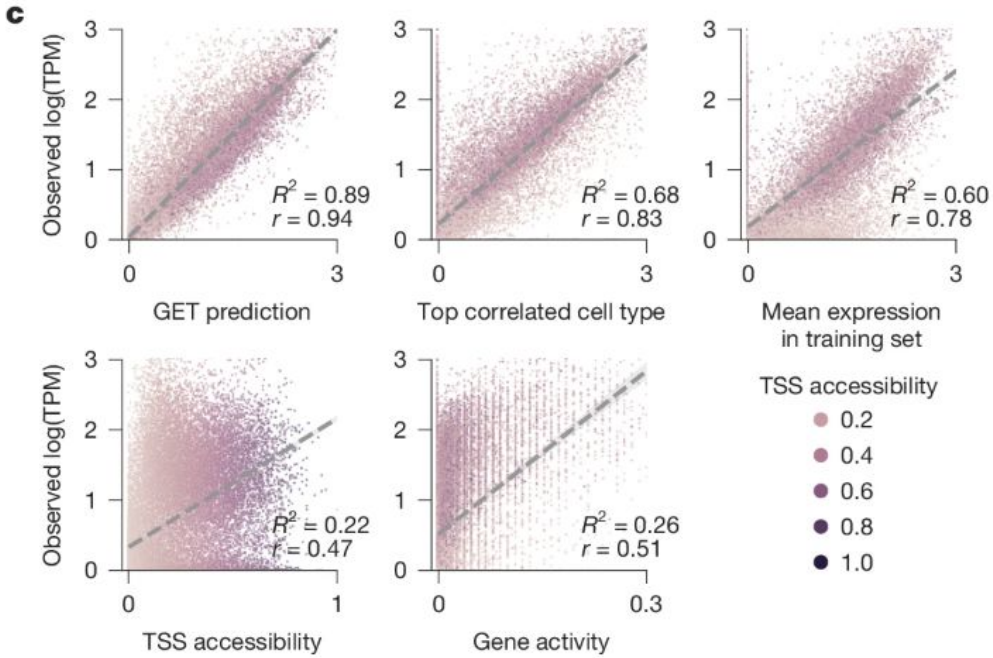
✅ Zero-shot prediction → Can predict gene expression without prior training on specific cell types.
✅ Identifies cis-regulatory elements → Finds important DNA regions controlling gene expression.
✅ Constructed GET Catalog → A public database of transcription factor (TF) interactions.
✅ First foundation model to predict transcription directly from chromatin landscapes!
✅ Interpretable!

Content

1. What can GET do???
2. How was GET trained???
3. How does GET generalize so well???
4. How did they make GET interpretable???

1. What can GET do???

1) GET predicts gene expression really well even on unseen cell types!

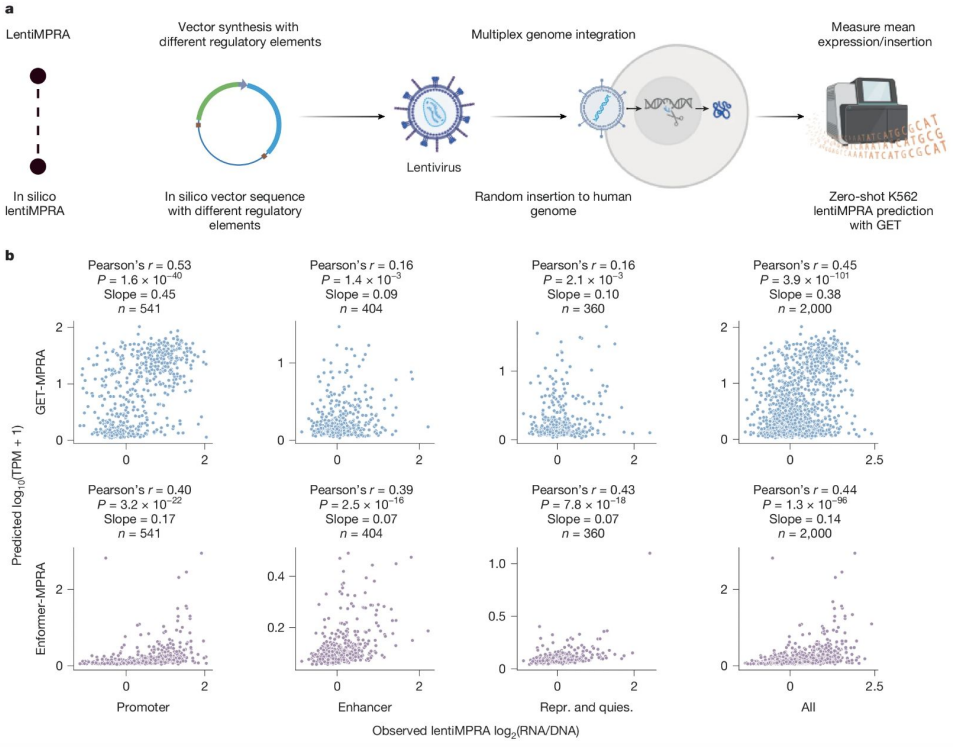


Trained GET without astrocyte data, test on astrocyte data.

GET achieved a Pearson correlation of 0.94 (R^2 = 0.88).

Comparable to real experimental replicates of astrocytes, which have r = 0.92–0.99! 🤯

2) GET identifies regulatory elements well without any prior training!



LentiMPRA integrates DNA sequences into the genome, and outputs activity

Recently inserted and tested 226,243 sequences in K562 leukemia cells, creating a gold-standard benchmark

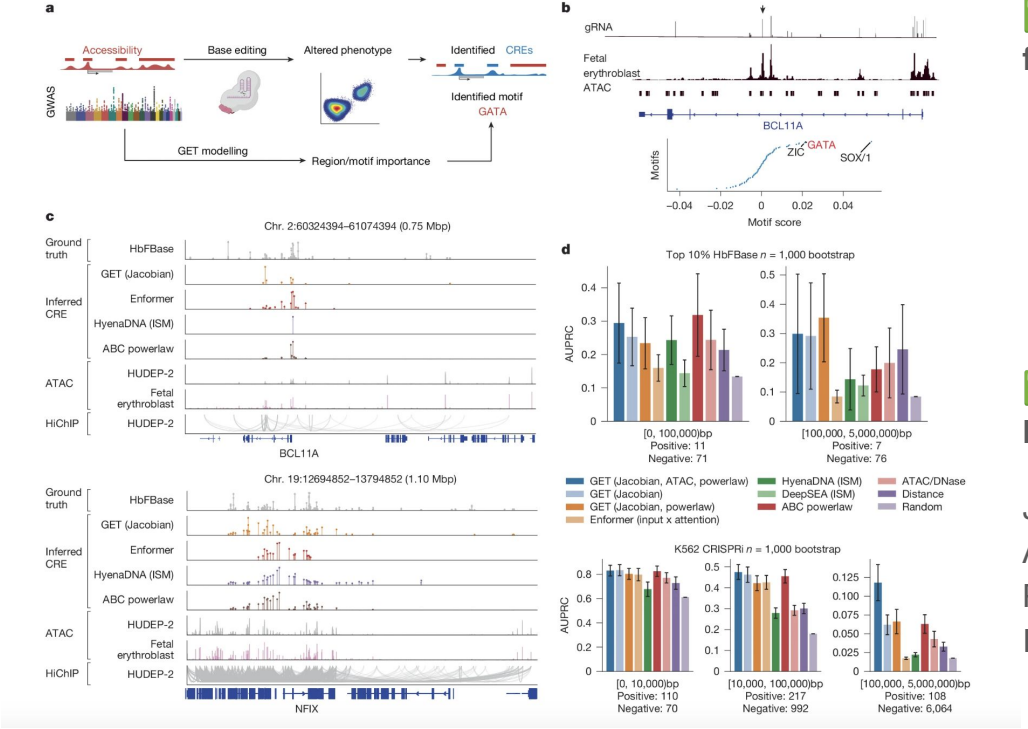
GET was fine-tuned on K562 chromatin accessibility & gene expression data, **never trained on LentiMPRA data.**

Enformer was trained on 486 different K562 assays, including TF binding, histone marks, and chromatin accessibility.

Enformer does better in low expression regions, by GET does better overall!

GET is significantly faster, allowing it to screen all 226,243 elements in the same time Enformer needed for just 2,000!

3) GET predicts long range enhancer–promoter pairs better than others



✅ **GET successfully rediscovered key transcription factors (TFs):**

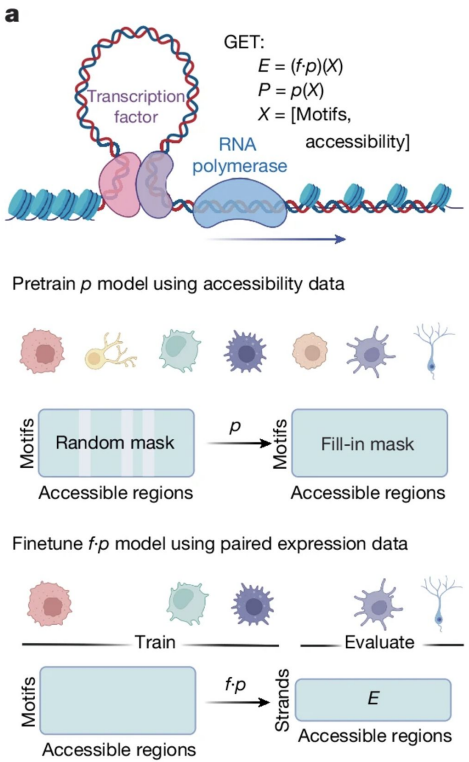
- **GATA TF** → Activates **BCL11A** (confirmed known science).
- **SOX TF family** → Linked to fetal hemoglobin before, but GET found **a new regulatory role in a specific enhancer**.

✅ **GET outperformed other models** at detecting **long-range enhancer–promoter interactions.**

Jacobian → Measures Feature Importance
ATAC → Measures Chromatin Accessibility
Powerlaw → Models Long-Range Enhancer-Promoter Interactions

2. How was GET trained??

GET model



Inputs for a local region (~2Mbp)

X = input features (TF binding scores, chromatin accessibility score)

p(X) = chromatin environment

Output

E = gene expression

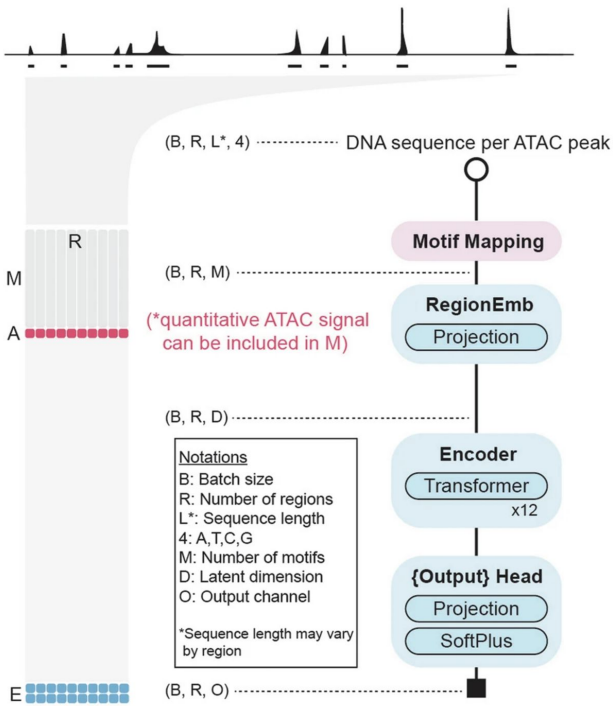
Pretraining (From X to p(X))

- Learns relationships among Xs by predicting missing (masked) TF binding, accessibility scores
- 16 NVIDIA V100 GPUs over 1 week (800 epochs).

Fine-tuning

- Trained to predict E from p(X)
- Same architecture but output head changed to focus on predicting E
- 8 A100 GPUs over 1 day (100 epochs).

GET Architecture



1. **Regulatory Element Embedding (RegionEmb)**
   a. Converts DNA regulatory elements into vectors
   b. 1 regulatory element = 282 transcription factor (TF) binding scores + 1 chromatin accessibility score

2. **Regulatory-Element-Wise Attention Layers (Encoder)**
   a. 12 layers of attention to analyze how regulatory elements interact with each other
   b. Learns both short and long-range gene regulation.

3. **Expression Prediction Output Layer**
   a. Final layer converts learned info into gene expression prediction
   b. Other specialized outputs like identifying cis-regulatory elements can also be generated

Honorable mention: LoRA!

a. Adapts large ML model for specific uses without retraining the entire model.
b. Reduces 99% of parameters

LoRA (Low-Rank Adaptation of Large Language Models)

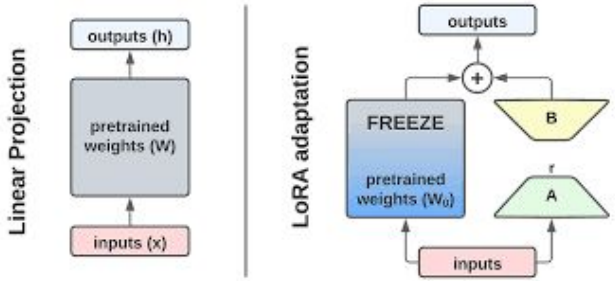
1) **Freeze the Pretrained Model**
   Instead of modifying the original model, LoRA keeps the pretrained weights unchanged so that general knowledge is intact.

2) Introduce Trainable Low-Rank Matrices (A & B)

- Instead of updating the full weights W, LoRA adds and trains small matrices (A & B) to capture adjustments.
  Mathematically: **W′=W+AB**
  - W = Original frozen model weights
  - A & B = Small trainable matrices

3) Fine-Tune Only the Small Matrices (A & B)

- Since A and B are much smaller than W, fewer parameters need to be updated.
- Drastically reduces memory usage with near full fine-tuning accuracy



3. How does GET generalize so well???

Leave things out so that GET do transfer learning better!

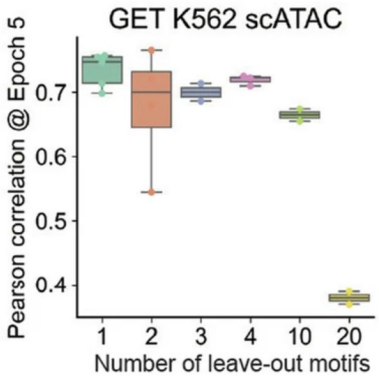
Leave-One-Chromosome-Out Testing

- Trained without one chromosome at a time, then tested on the missing chromosome.
- Tested on:
  - Fetal astrocytes (r=0.78), Tumor cells (r= 0.75), K562 cells (blood cancer cell line) (r=0.81)
    ✅ GET still made accurate predictions even when entire chromosomes were missing.
- Simulates real-world scenarios! where GET encounters new genomes/mutations/missing regulatory regions.

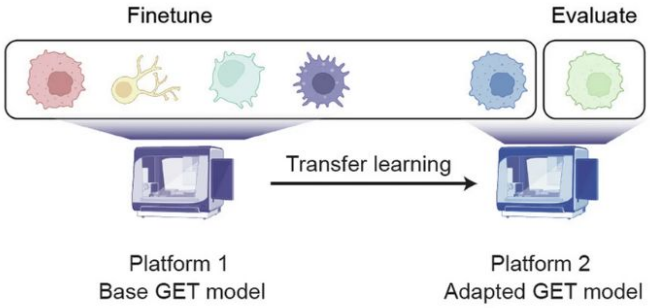
Leave-Out-Motif Testing

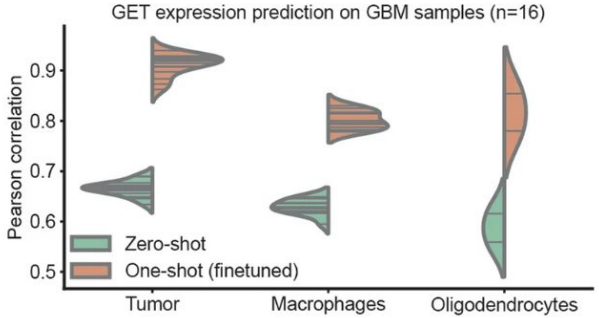
- Some TF motifs were hidden to see if GET could still predict gene expression.
- Forces GET to **learn how different motifs interact** rather than relying on individual TFs.
- Tested with 1, 2, 3, 4, 10, and 20 missing motifs.

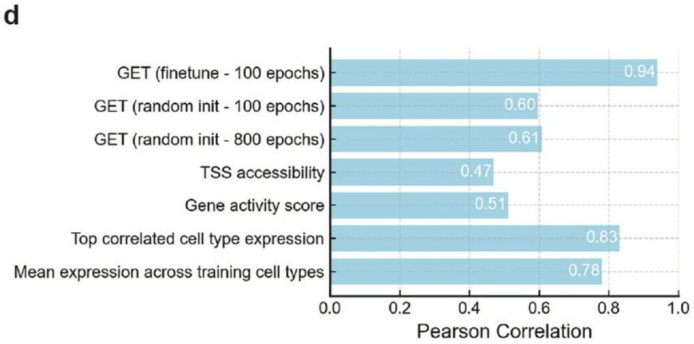
✅ GET was still accurate with up to 10 missing motifs.
⚠️ With 20 missing motifs, accuracy dropped because too much regulatory information was removed.

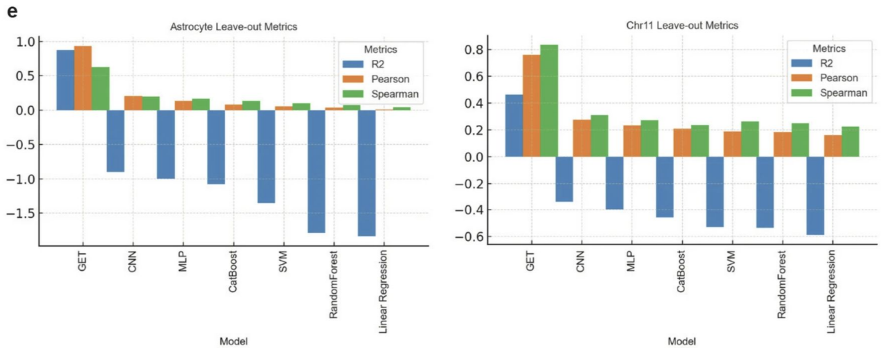


Fine tuning and transfer learning is what makes GET good!









4. How did they make GET interpretable???

1) **Feature Attribution**

- Quantitative ATAC Model → Uses chromatin accessibility strength (numerical score).
- Binary ATAC Model → Uses accessibility as a yes/no feature

  ✅ Binary model preferred → Ensures GET learns regulatory sequences, not just accessibility strength.
- Jacobian Matrix → Measures how changes in a motif affect gene expression, to identify important TF motifs

2) 🤨 **Which enhancer regulate what genes?**

  - GET ranks enhancer importance using Jacobian scores.
  - Uses distance-based predictions (e.g., Hi-C contact maps) to improve accuracy.
  - Experimental Validation with LentiMPRA

3) 🤨 **Do TF expressions match target gene's expression?**

- Gene-by-Motif Matrix → Mapped TFs to their target genes using Jacobian scores.
- Regulatory Embedding Space
  - Built using transformer attention layers.
  - Genes clustered based on regulation patterns, not just sequence similarity.

4) 🤨 **Find cause-and-effect relationships between transcription factors.**

- Built TF interaction networks using:
  - Spearman correlation & causal inference algorithms.
  - Compared GET's predictions to known TF interaction databases (STRING, ChIP-seq).

Conclusion

- GET - real or overhyped?
- Performance is good!
- DNA + ATAC data allows good predictions on gene expression
- Transcriptional regulators across different cell types are similar, GET learns this underlying grammar, which makes it good at predicting unseen cell types
- Data leakage?

Go try it out!

- Paper: https://t.ly/iQct_
- Model: https://t.ly/4jnUI
- Analysis package: https://t.ly/OqLAL
- Demo: https://t.ly/rbFQB
- Docker: https://t.ly/86n_i