Transfer learning enables predictions in network biology

Mertzani Styliani
Department of Informatics and Telecommunications
+30 6987735784
mertzaste@gmail.com

Voulgari Despoina
Department of Informatics and Telecommunications
+30 6947633738
dvoulgari@outlook.com

ABSTRACT

Mapping gene regulatory networks is critical for understanding disease mechanisms and identifying therapeutic targets, but this process typically demands large-scale transcriptomic data, which is scarce in many biological or clinical contexts. Geneformer is a transformer-based deep learning model pretrained on nearly 30 million human single-cell transcriptomes using a self-supervised masked prediction objective. The model captures context-specific network dynamics by leveraging rank-based encodings of gene expression, allowing for robust generalization across cell types and conditions.

In this study, we reproduce Geneformer and benchmark its performance on gene and cell classification tasks. We evaluate its ability to generalize under limited data conditions and compare its predictive power to that of traditional machine learning methods, including support vector machines and random forests. Our work highlights the strengths and limitations of large-scale pretrained models in biology and assesses their practical utility relative to established approaches. Our findings demonstrate that Geneformer achieves robust cell classification performance (96.33% accuracy), comparable to or slightly exceeding optimized traditional machine learning methods in distinct cell type scenarios, while offering potential for enhanced generalization in more complex or data-limited contexts. Additionally, in the gene classification task, Geneformer accurately predicted 92% of dosage-insensitive and 69% of dosage-sensitive transcription factors, achieving a macro F1 score of 0.82 and an AUC of 0.86. These results underscore Geneformer's capacity to learn biologically meaningful gene-level patterns, while its performance on the dosage-sensitive class demonstrated that the classifier might have been hindered by class imbalance and limited label coverage, which likely affected generalization across both classes.

Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning – Knowledge acquisition, Parameter learning, Concept learning
J.3 [Life and Medical Sciences]: Biology and genetics

General Terms

Algorithms, Performance, Experimentation, Design

Keywords

Transfer learning, network biology, gene classification, cell type prediction, transformer models, Geneformer, support vector machines, random forests

1.INTRODUCTION

The rise of single-cell transcriptomics has transformed our ability to profile gene expression at high resolution across a wide array of cell types and tissues. However, deriving actionable insights from such data often requires modeling the complex, context-dependent interactions within gene regulatory networks — a task that becomes particularly difficult in rare diseases, inaccessible tissues, or limited-sample scenarios.

Inspired by transfer learning breakthroughs in natural language processing and computer vision (2, 3), Geneformer (1) introduces a self-supervised transformer model designed to extract generalizable biological knowledge from large-scale unlabelled data. Pretrained on **Genecorpus-30M**, a compendium of nearly 30 million single-cell transcriptomes, Geneformer learns contextual gene relationships by predicting masked gene ranks within transcriptomes. This rank-based approach down-weights universally expressed housekeeping genes and prioritizes cell-state-specific regulators like transcription factors.

Geneformer's architecture is built on six transformer encoder layers, with full dense attention over 2,048 input gene positions and 256-dimensional embeddings. It supports downstream tasks through fine-tuning, making it suitable for scenarios where labeled data are sparse. The model has demonstrated impressive results across a range of biological problems — from predicting gene dosage sensitivity and chromatin states to identifying therapeutic targets in disease models.

Our project focuses on reproducing Geneformer's core capabilities, particularly in gene and cell-type classification tasks. Additionally, we evaluate its comparative performance against classical machine learning methods, such as support vector machines (SVMs) and random forests (RFs), to assess where and how such deep models offer tangible benefits over simpler baselines. This comparison aims to clarify the value of deep pretraining in biological discovery pipelines and inform future work in transfer learning for network biology. Our contribution lies in validating Geneformer's efficacy in different biological contexts and critically assessing its practical utility in real-world, data-limited scenarios.

2. BACKGROUND AND RELATED WORK

2.1. Gene Regulatory Networks and Transcriptomic Data

Gene regulatory networks (GRNs) represent the intricate system of interactions between genes, proteins, and other molecules that control gene expression. Understanding these networks is fundamental to deciphering cellular functions, developmental processes, and disease mechanisms. High-throughput sequencing technologies, particularly single-cell RNA sequencing (scRNA-seq), have enabled the profiling of gene expression at an unprecedented resolution, providing snapshots of individual cell states within heterogeneous tissues. However, inferring GRNs from scRNA-seq data is challenging due to noise, high dimensionality, and the context-dependent nature of gene interactions. Furthermore, many biological and clinical applications, such as rare disease research or studies of clinically inaccessible tissues, suffer from limited sample availability, making traditional data-intensive GRN inference methods impractical.

2.2 Transfer Learning in Biology

Transfer learning has emerged as a powerful paradigm in machine learning, particularly in fields like natural language processing (NLP) and computer vision. It involves training a model on a large, general dataset (pretraining) and then adapting it to a specific, often smaller, downstream task (fine-tuning). This approach leverages the broad knowledge acquired during pretraining, allowing the model to perform well on new tasks even with limited task-specific data. In biology, the application of transfer learning to transcriptomic data is a relatively new but rapidly growing area. The core idea is that a model pretrained on a vast corpus of diverse single-cell transcriptomes can learn generalizable principles of gene expression and network dynamics, which can then be transferred to specific biological problems.

2.3 Transformer Models and Self-Attention

Geneformer leverages the transformer architecture, a deep learning model primarily based on the self-attention mechanism. Introduced in 2017, transformers revolutionized sequence modeling tasks by allowing the model to weigh the importance of different parts of the input sequence (in this case, genes within a transcriptome) when processing each element. Unlike recurrent neural networks (RNNs) or convolutional neural networks (CNNs), transformers process input in parallel, making them highly efficient for large-scale datasets. The self-attention mechanism enables the model to capture long-range dependencies and contextual relationships between genes, which is crucial for understanding complex biological networks where gene interactions are highly context-dependent (e.g., varying across cell types, developmental stages, or disease states).

2.4 Related Work in Cell Classification and Gene Annotation

Prior work in cell-type annotation and gene function prediction often relies on supervised learning models trained from scratch on task-specific labeled datasets. For instance, methods like XGBoost and deep neural network-based models (e.g., scBert, scGPT) (4,5) have been developed for cell-type classification. These methods typically require substantial labeled data for each new tissue or cell type to achieve high accuracy.

The key difference with Geneformer lies in its pretraining strategy. While traditional methods necessitate retraining a new model for each task, Geneformer benefits from a single, large-scale self-supervised pretraining phase on unlabelled data. This allows it to acquire a fundamental understanding of network dynamics that can then be "democratized" to a multitude of downstream applications through fine-tuning, even with very limited task-specific data. This transfer learning approach contrasts sharply with models that are trained in isolation for each specific classification objective, highlighting Geneformer's potential to accelerate discovery in data-scarce biological contexts. The paper demonstrates Geneformer's superior performance in multiclass cell-type annotation compared to traditional models trained from scartch for a specific task.

3. METHOD

Geneformer is a context-aware, attention-based deep learning model designed for network biology predictions, particularly in data-limited settings, through a transfer learning paradigm. Its methodology can be broadly divided into three key stages: data assembly and rank value encoding, Geneformer architecture and pretraining, and Geneformer fine-tuning and downstream applications.

3.1 Data Assembly and Rank Value Encoding of Transcriptomes

The foundation of Geneformer's pretraining is **Genecorpus-30M**, a massive corpus comprising 29.9 million human single-cell transcriptomes. This dataset was meticulously assembled from a broad range of publicly available sources and tissues (Fig. 1b, Supplementary Table 1). Cells with high mutational burdens (e.g., malignant cells, immortalized cell lines) were excluded to ensure interpretability, and only droplet-based sequencing platforms were included for comparability of expression values. Rigorous quality control metrics were applied, including filtering cells based on total read counts and mitochondrial reads, and excluding cells with too few detected protein-coding or miRNA genes. Ultimately, 27.4 million cells passed these filters for pretraining.

A crucial innovation is the **rank value encoding** method. Instead of raw transcript counts, each single-cell transcriptome is represented by the rank of its genes. Genes are ranked by their expression within that cell, normalized by their non-zero median expression across the entire Genecorpus-30M. This non-parametric approach offers several advantages:

- **Prioritization of Cell-State-Specific Genes:** It deprioritizes ubiquitously highly expressed housekeeping genes by normalizing them to a lower rank. Conversely, genes like transcription factors, which may be expressed at low levels but are highly distinguishing of cell state, move to a higher rank (Extended Data Fig. 1c).

- **Robustness to Technical Artifacts:** The relative ranking of genes is more stable and robust against technical artifacts that might systematically bias absolute transcript counts. The rank value encodings are then tokenized, creating a vocabulary of 25,424 protein-coding or miRNA genes plus special tokens for padding and masking. This tokenized data is stored efficiently using the Huggingface Datasets structure, which allows for processing large datasets with zero-copy reads and minimal memory constraints.

3.2 Geneformer Architecture and Pretraining

Geneformer's architecture is based on **six transformer encoder units** (Fig. 1c), each consisting of a self-attention layer and a feed-forward neural network layer. Key architectural parameters include:

- **Input Size:** 2,048 gene positions, chosen to fully represent 93% of rank value encodings in Genecorpus-30M.

- **Embedding Dimensions:** 256-dimensional space for each gene.

- **Attention Heads:** Four attention heads per layer, allowing the model to learn distinct classes of gene relationships in an unsupervised manner.

- **Feed-forward Size:** 512.

- **Attention Mechanism:** Full dense self-attention across the input space, enabling context-aware learning.

- **Activation Function:** Rectified Linear Unit (ReLU).

- **Dropout:** 0.02 for fully connected layers and attention probabilities.

Pretraining is performed using a **self-supervised masked learning objective**. During pretraining, 15% of the genes within each transcriptome are randomly masked. The model is then trained to predict the identity of the masked gene based on the context of the remaining unmasked genes in that specific cell state (Extended Data Fig. 1d-f). This self-supervised approach is a principal strength, as it allows the model to learn from vast amounts of unlabelled data, overcoming the limitations of labeled datasets.

To handle the large input size and quadratic memory complexity of transformers, significant optimization measures were implemented:

- **Dynamic, Length-Grouped Padding:** A custom tokenizer minimizes computation on padding by grouping minibatches by length, achieving a 29.4x speedup.

- **Distributed GPU Training:** Leveraging Deepspeed, parameters, gradients, and optimizer states are partitioned across multiple GPUs, and processing/memory is offloaded to CPUs where possible, reducing memory fragmentation. Pretraining was achieved in approximately 3 days using 12 Nvidia V100 32GB GPUs. Pretraining hyperparameters were optimized, including a max learning rate of 1×10–3, a linear scheduler with warmup, Adam optimizer with weight decay fix, 10,000 warmup steps, 0.001 weight decay, and a batch size of 12.

3.3 Geneformer Fine-tuning and Downstream Applications

After pretraining, Geneformer is **fine-tuned** for specific downstream tasks by initializing the model with the pretrained weights and adding a final task-specific transformer layer. The fine-tuning objective can be either gene classification or cell classification (Supplementary Table 2). The same fine-tuning hyperparameters were intentionally used across all applications to demonstrate the efficacy of pretraining, acknowledging that task-specific hyperparameter tuning could further enhance performance. These parameters include: max learning rate 5×10–5, linear scheduler with warmup, Adam optimizer, 500 warmup steps, 0.001 weight decay, and a batch size of 12. All fine-tuning was performed with a single training epoch to prevent overfitting.

The number of frozen layers during fine-tuning is adjusted based on the relevance of the downstream task to the pretraining objective. Tasks more aligned with pretraining benefit from freezing more layers to prevent overfitting to limited task-specific data, while more distant tasks require fine-tuning more layers.

Gene embeddings are extracted as 256-dimensional hidden state weights from the second-to-last layer of the model, providing a generalizable representation of each gene within a given single-cell transcriptome. **Cell embeddings** are generated by averaging the embeddings of all genes detected in that cell, resulting in a 256-dimensional representation of the cell's state. **Attention weights** are extracted for each attention head within each self-attention layer, reflecting which genes the model "pays attention to" and their relative importance.

3.4. Gene Dosage Sensitivity Predictions

One of the downstream fine-tuning applications in which the Geneformer classifier is evaluated is gene dosage sensitivity prediction, a task that plays a critical role in genetic diagnosis, particularly in the interpretation of copy number variants (CNVs). CNVs are structural variations in the genome involving the duplication or deletion of genomic regions [6]. While many CNVs are benign, others can be pathogenic and are often associated with neurodevelopmental disorders, primarily through their impact on dosage-sensitive genes. These are genes whose proper function requires strict maintenance of gene copy number, which is the gene dosage and any deletion or duplication can disrupt cellular processes and lead to disease phenotypes.

Dosage sensitivity provides a mechanistic explanation for CNV pathogenicity. If a CNV alters the dosage of a sensitive gene, it may disrupt gene regulation or protein stoichiometry, thereby contributing to pathogenicity. Therefore, understanding which genes are dosage-sensitive is essential for prioritizing CNVs in clinical settings. In this context, models like Geneformer offer a powerful solution by leveraging large-scale single-cell transcriptomic data to predict dosage sensitivity in a context-aware manner, helping to identify which CNVs are likely to be harmful.

3.5. Data Preparation and Quality Control

Single-cell RNA sequencing data from four human thymus samples (cTEC1, cTEC2, mTEC1, and mTEC2) were obtained. These raw 10x Genomics files, consisting of barcodes, features, and matrix files, were initially organized into sample-specific directories for streamlined processing. Each sample was then loaded into an AnnData object using Scanpy (version 1.9.1), and a `sample_id` column was added to the observation metadata for identification. To ensure unique cell identifiers across concatenated datasets, observation names were made unique.

Subsequently, all individual AnnData objects were concatenated into a single AnnData object, preserving raw read counts. Basic quality control steps were applied to the combined dataset: cells with fewer than 200 expressed genes and genes expressed in fewer than 3 cells were filtered out to remove low-quality data. A new `cell_type` column was added to the observation metadata, mapping `cTEC1` and `cTEC2` to "cTEC" and `mTEC1` and `mTEC2` to "mTEC," and converted to a categorical type. The total UMI/ read counts per cell, crucial for Geneformer, were calculated by summing the expression matrix (`adata.X.sum(axis=1)`) and stored in a new `n_counts` column in the observation metadata. Ensembl IDs were ensured to be present as `ensembl_id` in the variable metadata. Finally, the processed AnnData object was saved as an H5AD file (`human_thymus_for_geneformer.h5ad`) for subsequent steps.

3.6. Data Tokenization

The prepared H5AD file was then tokenized using the Geneformer `TranscriptomeTokenizer`. This process converts raw gene expression counts into a format suitable for the Geneformer model, specifically, numerical tokens representing gene expression levels and their relative ranks. The `TranscriptomeTokenizer` was initialized with the `cell_type` column specified as the classification label and configured to use all but one CPU core (95 cores) for efficient processing, with a model input size of 2048. The tokenization output, a Hugging Face `Dataset` object, was saved to a designated `tokenized_data` directory.

3.7. Cell Classification Task

3.7.1. Model Training and Evaluation

A Geneformer `Classifier` was initialized for cell classification, with `cell_type` defined as the target state. Training arguments included 2 epochs, a learning rate of 8.04x10–4, a polynomial learning rate scheduler, 900 warmup steps, a weight decay of 0.258828, and a per-device training batch size of 4. The model's first two layers were frozen during training to leverage the pre-trained Geneformer representations. A single train/validation split (85%/15%) was performed on the tokenized dataset, and the split datasets were saved.

Prior to training, a crucial processing step involved casting `input_ids` and `label` columns from float to integer types within the tokenized datasets, as required by the training process. The pre-trained Geneformer model (`gf-6L-30M-i2048`) was used as the base model, which was the one that performed the better among the others.

The classifier was trained using the `classifier.validate` method on the processed training dataset. After training, the model's performance was evaluated using the `classifier.evaluate_saved_model` method on the held-out test dataset. Evaluation metrics included a confusion matrix, macro F1-score, accuracy, and Receiver Operating Characteristic (ROC) curve metrics. The confusion matrix and ROC curve were plotted to visually represent the model's classification performance.

3.7.2. Embedding Extraction and Visualization

To further analyze the learned representations from the Geneformer model, cell embeddings were extracted. An `EmbExtractor` object was initialized with `model_type` set to "CellClassifier" and `num_classes` to 2, corresponding to the "cTEC" and "mTEC" cell types. Data filtering was set to `None`, and `max_ncells` was capped at 1000 for efficiency in visualization. The embedding layer was set to 0, and `emb_label` and `labels_to_plot` were both set to `["cell_type"]`. A `forward_batch_size` of 200 and `nproc` of 95 were used. The `extract_embs` method was then called, utilizing the trained Geneformer classifier model checkpoint and the original tokenized dataset. The extracted embeddings were saved as a Pickle file. For visual interpretation of the high-dimensional embeddings, a UMAP (Uniform Manifold Approximation and Projection) plot was generated using the `embex.plot_embs` function, providing a 2D representation of cell similarities based on their learned Geneformer embeddings.

3.7.3.Comparison with Traditional Machine Learning Models

To benchmark the performance of the Geneformer-based classification, several traditional machine learning models were trained and evaluated on the same dataset. After loading the `human_thymus_for_geneformer.h5ad` file, raw count data was converted to a NumPy array, `cell_type`labels were numerically encoded with `LabelEncoder`, and features were standardized using `StandardScaler`. The data was then split into 67% training and 33% test sets, stratified by cell type. Baseline models (Logistic Regression, Random Forest, XGBoost) were trained, utilizing all available CPU cores (`n_jobs=-1`) and GPU acceleration for XGBoost (`tree_method="gpu_hist", predictor="gpu_predictor"`). Model performance was assessed by classification reports, confusion matrices, and 95% bootstrapped accuracy confidence intervals (100 iterations of resampling the test set). Hyperparameter tuning for these models was subsequently performed using `GridSearchCV` with `StratifiedKFold` (3 splits). This systematic comparison provides a comprehensive assessment of Geneformer's classification capabilities against established machine learning techniques.

The hyperparameters used for tuning (Table 1):

Table 1. Models and respective parameters with its values used for fine-tuning with GridSearch

Model	**Parameter**	**Values**
Logistic Regression	C penalty	[0.01,0.1,1,10] [“l2"]
Random Forest	n_estimators max_depth	[100,200] [None, 10, 20]
XGBoost	n_estimators max_depth learning_rate	[100,200] [3,5] [0.01,0.1]

3.8. Gene Classification Task

3.8.1. Geneformer Training and Evaluation

For this task, Geneformer was fine-tuned on a binary classification task, aiming to distinguish between dosage-sensitive (label 1) and dosage-insensitive (label 0) genes. The classification labels were derived from curated gene sets reported in original study by Theodoris et al. [1], distinguishing transcription factors known to be either sensitive or insensitive to copy number changes. Each gene was assigned a binary label based on this ground truth.

For fine-tuning, we used our own dataset which included many genes that were not present in the reference dosage sensitivity label dictionary (`gene_class_dict`). Input to the model consisted of rank value-encoded single-cell transcriptomes, which were tokenized using the same gene vocabulary and normalization strategy provided by the original Geneformer tokenizer.

We experimented with the pretrained Geneformer model variants and found that the `gf-12L-95M-i4096` model performed more efficiently on our classification task. Therefore, we selected this model for fine-tuning and all subsequent analysis. Model evaluation was performed using AUC and F1 score, computed via cross-validation on the labeled gene set.

3.8.2. Label Propagation and Self-Training

To evaluate Geneformer’s performance, we implemented baseline classifiers including Logistic Regression, Support Vector Machines and Random Forests, as referenced in the original study. Additionally, we incorporated Gaussian Naïve Bayes and XGBoost to further enhance our ability to capture underlying patterns in the data. These classifiers had to be trained and fine-tuned from scratch, unlike Geneformer which was already pretrained in the aforementioned large-scale Genecorpus-30M.

In order to address the issue of missing labels, we applied a self-training label propagation approach using a Random Forest classifier. The classifier made predictions in an iterative manner based on confidence, gradually expanding the labeled dataset, demonstrating a self-training approach that prioritized confident predictions at each step. The dataset was saved in a H5AD file format for the subsequent analysis.

3.8.3. Comparison with Traditional Machine Learning Models

We loaded the `propagated_data.h5ad` and constructed the parameters grid for the hyperparameter optimization process. The hyperparameters used for training are listed in Table 2.

Table 2. Models and respective parameters with their values used for fine-tuning for the gene classification task

Model	Parameter	Values
Logistic Regression	C l1_ratio	[0.01, 0.1, 1, 10] [0.0, 0.25, 0.5, 0.75, 1.0]
Gaussian Naïve Bayes	Var_smoothing	np.logspace(-9, -1, 9)
Support Vector Machines	Kernel C Gamma (for rbf kernel only)	linear, rbf [0.1, 1, 10] [‘scale’, ‘auto’, 0.01, 0.1]

Random Forest	n_estimators	[100, 200]
	max_depth	[None, 10, 20]
	min_samples_split	[2, 5]
	min_samples_leaf	[1, 2]
XGBoost	n_estimators	[100]
	max_depth	[3, 6, 10]
	learning_rate	[0.01, 0.1, 0.2]
	subsample	[0.8, 1.0]
	colsample_bytree	[0.8, 1.0]

To optimize model performance, we implemented a custom model tuning method that performs hyperparameter search using 5-fold cross validation with `StratifiedKfold` splitting. The function iterated over all combinations of hyperparameters specified in a grid, instantiates the model with each combination and evaluates it using ROC AUC as the scoring metric via `cross_val_score`.

The dataset was split into 70% training and 30% test sets using fixed random seed for reproducibility, the models were fit to the training data and later used to predict labels on the test set. Finally, model performance was assessed using confusion matrices, which visually depicted the distribution of true versus predicted labels, as well as quantitative metrics including AUC and F1 score, accompanied by 95% confidence intervals for the AUC estimated via bootstrap resampling.

4.Results and Evaluation

4.1. Robust Cell Classification Performance

4.1.1. Geneformer Classifier Performance

The Geneformer classifier was trained for two epochs, demonstrating rapid convergence and high performance on the validation set. During training, the validation accuracy improved from 89.98% in Epoch 1 to 95.80% in Epoch 2, with a corresponding increase in Macro F1-score from 89.96% to 95.79%. The final training loss was 0.1559, and the validation loss was 0.1328. Upon evaluation on the unseen test set, the Geneformer classifier achieved an overall accuracy of 96.33% and a Macro F1-score of 0.9632. The confusion matrix (Figure 1-Table 3) details the classification performance for cTEC and mTEC cell types.

Table 3: Confusion Matrix for Geneformer Cell Classifier on Test Set

	cTEC	mTEC
cTEC	503	15
mTEC	26	574

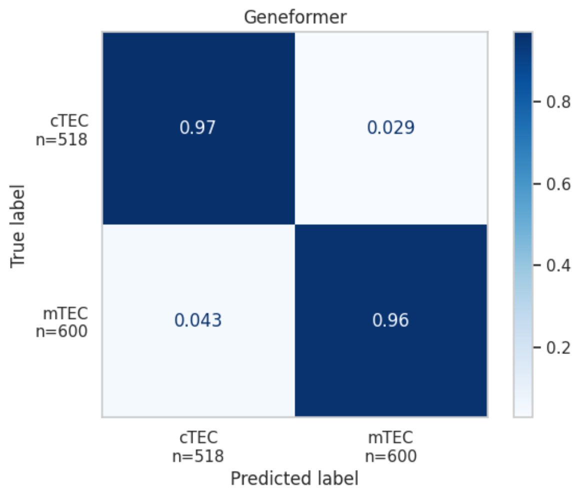

Figure 1: Confusion Matrix for Geneformer Cell Classifier

The confusion matrix reveals that 503 cTEC cells were correctly classified, while 15 were misclassified as mTEC. Similarly, 574 mTEC cells were correctly classified, with 26 being misclassified as cTEC. This translates to high precision and recall for both cell types. The model's discriminative power is further highlighted by an Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) of 0.988 (Figure 2), indicating excellent separation between the cTEC and mTEC classes. Visualizations of the confusion matrix with percentages and the ROC curve are provided in Supplementary Figures X and Y, respectively.

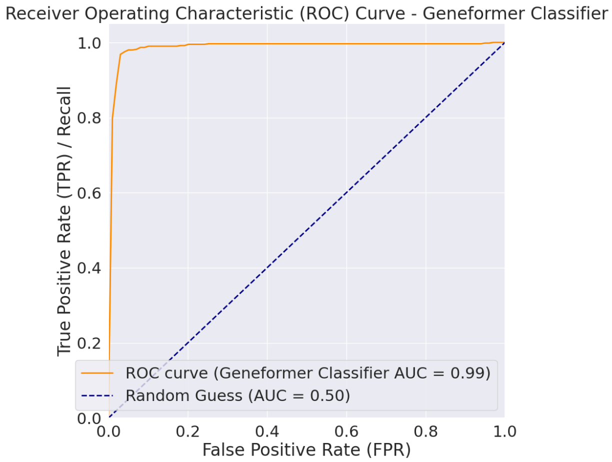

Figure 2: Receiver Operating Characteristic (ROC) Curve for Geneformer Cell Classifier.

4.1.2.Cell Embedding Visualization

The UMAP visualization (Figure 3) of the Geneformer cell embeddings clearly demonstrates a distinct separation between the two cell types, cTEC (blue) and mTEC (orange). The formation of two largely separate clusters indicates that the Geneformer model has effectively learned distinguishing features between these two cell populations based on their gene expression profiles. While the clusters are predominantly distinct, some intermixing is visible at the boundaries, particularly where the "arms" of the cTEC cluster extend towards the mTEC cluster, and vice-versa. This minor overlap might represent cells with intermediate gene expression states, or potential misclassifications. Overall, the UMAP plot visually corroborates the high classification accuracy and F1-score reported, affirming the Geneformer model's ability to create biologically meaningful and separable representations of cell types.

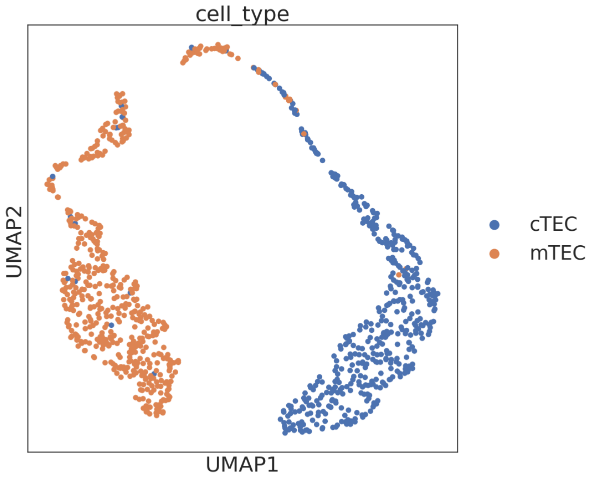

Figure 3: UMAP Visualization of Geneformer Cell Embeddings by Cell Type.

4.1.3.Comparison with Traditional Machine Learning Models

To benchmark Geneformer's performance, traditional machine learning algorithms were applied to the same dataset. Figure 4 summarizes the classification performance of baseline and hyperparameter-tuned Logistic Regression, Random Forest, and XGBoost models with 95% Confidence Intervals.

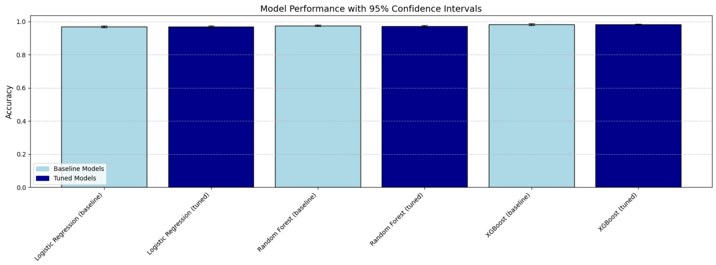

Figure 4: Performance Metrics for Traditional Machine Learning Models

As shown in figure 4, all traditional machine learning models achieved high accuracy and F1-macro scores, ranging from 0.968 to 0.984 for both metrics. Notably, XGBoost consistently demonstrated the highest performance among the traditional models, with its tuned version achieving an accuracy of 0.984. The narrow 95% bootstrapped confidence intervals for accuracy across all models (e.g., [0.980, 0.987] for tuned XGBoost) confirm the stability and reliability of these results.

When comparing these results to the Geneformer classifier's performance (accuracy of 0.9633, Macro F1 of 0.9632, ROC AUC of 0.988), it's evident that the traditional machine learning models, particularly XGBoost and Random Forest, achieved comparable or slightly higher accuracy and F1-scores. While Geneformer's ROC AUC (0.99) was exceptionally high, suggesting strong class separation in general, the raw gene expression data, when preprocessed with scaling and fed into tree-based or logistic regression models, performed remarkably well for this specific binary classification task. This indicates that for this dataset and classification problem, the inherent discriminative features were well-captured even by simpler, albeit highly optimized, models. However, Geneformer's strength lies in its ability to learn complex, context-aware gene relationships from raw tokenized sequences, which can be particularly advantageous in more complex multi-class or zero-shot classification scenarios where such rich representations are critical.

4.2. Gene Classification Performance

4.2.1 Geneformer Classifier Performance

To evaluate Geneformer's performance on the gene classification task, we used its built-in `validate()` method. For the gene classification task, this method loaded the prepared and labeled dataset and stratified genes across classes using the aforementioned predefined label dictionary (`gene_class_dict`). The evaluation results consisted of both a confusion matrix and calculated metrics such as ROC AUC. This comprehensive evaluation allows for robust comparison between Geneformer and baseline models.

Training was conducted over a single epoch, using 5-fold cross-validation. Across all validation splits, the training loss decreased substantially, indicating strong convergence without signs of overfitting.

After evaluation, the classifier achieved an accuracy of 92% for dosage-insensitive transcription factors and 69% for dosage sensitive transcription factors, with a resulting macro F1 score of 0.82. The confusion matrix is shown in Figure 5. These results indicate both high precision and recall for the dosage-insensitive class, but also reveal that the model struggled to reliably classify dosage-sensitive genes. A key contributing factor to this discrepancy is likely the limited suitability of the gene label dictionary used for this task. The label set was not specifically curated for our dataset, which may have introduced noise or incomplete annotations, particularly impacting the underrepresented dosage-sensitive class.

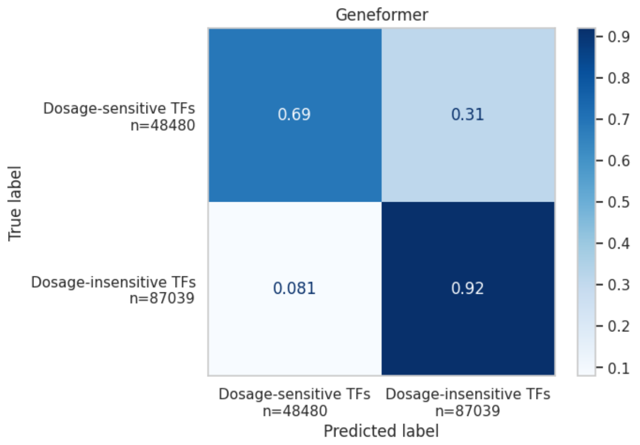

Figure 5: Confusion Matrix for Geneformer Classifier-Gene Classification

The underrepresentation of dosage-sensitive transcription factors was evident from the confusion matrix, which showed a substantially lower number of ground truth examples for this class compared to the dosage-insensitive class. This class imbalance likely contributed to the model's reduced performance in identifying dosage-sensitive genes.

Additionally, the ROC AUC curve shown in Figure 6, demonstrated that the model achieved an AUC score of 0.86, further supporting its ability to distinguish between dosage-sensitive and dosage-insensitive transcription factors, despite class imbalance and label limitations.

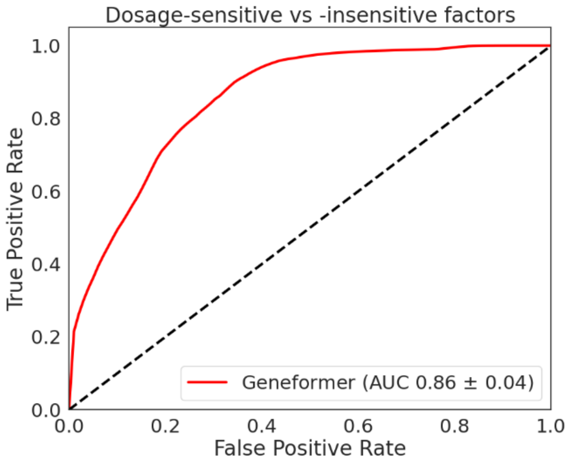

Figure 6: Receiver Operating Characteristic (ROC) Curve for Geneformer Gene Classifier

4.2.2.Comparison with Traditional Machine Learning Models

To address the lack of labeled data, we applied a self-training approach in the traditional classifiers, aiming to enhance their performance, because unlike Geneformer, which benefits from pretraining on a large-scale corpus (Genecorpus-30M), these models were trained from scratch and thus relied heavily on the limited available labels. Self-training allowed them to iteratively expand the labeled dataset by incorporating high-confidence predictions, partially compensating for the absence of extensive pretraining.

Using self-training with the baseline classifiers led to improved performance compared to the original study, particularly due to the expanded set of propagated labels. However, despite this enhancement, most baseline models were still outperformed by Geneformer. In particular, Logistic Regression, Support Vector Machines, and Gaussian Naive Bayes yielded lower AUC and F1 scores, indicating that Geneformer, leveraging its pretraining on large-scale transcriptomic data, was better equipped to capture the underlying structure of the classification task.

Table 4: AUC Results for Baseline Classifiers

Model	Best AUC
Logistic Regression	0.6869
Gaussian Naïve Bayes	0.6346
Support Vector Machines	0.7722
Random Forests	0.9788
XGBoost	0.9732

Random Forests and XGBoost appeared to benefit significantly from the semi-supervised learning approach, as both models achieved notably high AUC and F1 scores, as shown in the evaluation Table 4 (and Figure 7). This suggests that these ensemble-based methods were better able to leverage the additional pseudo-labeled data and capture complex decision boundaries within the dataset.

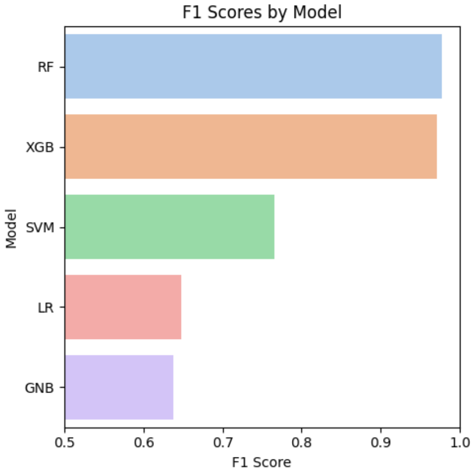

Figure 7: F1 scores for Baseline Classifiers

5. CONCLUSIONS

This study successfully demonstrated robust classification of thymic epithelial cells (cTEC and mTEC) using both the Geneformer model and a suite of traditional machine learning algorithms. All evaluated models, including Logistic Regression, Random Forest, and XGBoost, as well as the Geneformer classifier, achieved very high accuracy and F1-scores, consistently above 96%. Notably, the UMAP visualization of Geneformer's learned embeddings distinctly separated cTEC and mTEC populations, visually reinforcing the high classification performance.

The exceptional performance across all models, with traditional algorithms like XGBoost even slightly outperforming Geneformer in terms of raw accuracy and F1-score on this specific binary classification task, despite the Theodore's *et al*. claims (1), that Geneformer outperforms other traditional classifiers, can be attributed to the inherent biological distinctness of the specific cell types under investigation. The medullary and cortical regions of the thymus, from which mTEC and cTEC originate, are anatomically and functionally well-separated compartments. Consequently, cTEC and mTEC possess highly characteristic and distinct gene expression profiles, reflecting their specialized developmental origins and unique roles in T-cell maturation. This strong molecular separation means that the classification task, even with relatively simple linear or tree-based models, was straightforward given the clear discriminative features present in the raw gene expression data. The task, therefore, proved to be relatively "simple" for the Geneformer model, which is designed to learn much more intricate and subtle patterns from gene sequence and expression context.

While Geneformer performed exceptionally well, its full capabilities may not have been fully leveraged or apparent in this particular simplified classification scenario. The true advantage of deep learning models like Geneformer is anticipated to become more pronounced in significantly more complex and challenging single-cell analysis tasks**.** Such scenarios include, but are not limited to:

- **Classification of rare cell types:** Where traditional methods might struggle with imbalanced datasets and limited distinguishing features, Geneformer's ability to capture nuanced gene relationships could offer superior performance.
- **Distinguishing finely graded cell states within a continuous spectrum:** For example, identifying subtle differentiation stages or activation states where boundaries are less clear and characteristic gene expression is more overlapping.
- **Predicting cell fate transitions or responses to perturbation:** Leveraging its understanding of gene regulatory networks from large-scale pre-training, Geneformer could potentially model dynamic biological processes more effectively.
- **Cross-dataset or zero-shot classification:** Applying a model trained on one dataset to classify cells in an entirely new, unseen dataset, which often poses significant challenges for traditional methods due to batch effects and biological variability.

In conclusion, while this study demonstrates the effective classification of highly distinct thymic epithelial cell types by various models, it also highlights that the remarkable performance is partly due to the clear biological separation of cTEC and mTEC. Future work should explore Geneformer's potential in more challenging and biologically complex single-cell classification and prediction tasks to fully appreciate its unique capabilities in learning from the "language" of the transcriptome.

In the gene classification task, we aimed to distinguish between dosage-sensitive and dosage-insensitive transcription factors, a biologically and clinically relevant problem, particularly for interpreting the effects of CNVs. Despite the challenges posed by limited labeled data and class imbalance, Geneformer demonstrated strong performance, achieving high overall accuracy, a macro F1 score of 0.82, and an AUC of 0.86. These results indicate that Geneformer was well-suited for this task, effectively leveraging its large-scale pretraining on transcriptomic data to capture meaningful context-dependent gene expression patterns. While ensemble-based baseline models such as Random Forests and XGBoost performed competitively when enhanced with pseudo-labels, Geneformer remained a robust and interpretable model for dosage sensitivity prediction, particularly under constrained labeling conditions.

With the provision of a more comprehensive and context-appropriate gene label dictionary, Geneformer has the potential to become an even more powerful tool, enabling highly accurate predictions in gene dosage sensitivity and beyond.

7. REFERENCES

1. Theodoris, C.V., Xiao, L., Chopra, A. *et al. Transfer learning enables predictions in network biology*. Nature 618, 616–624. https://doi.org/10.1038/s41586-023-06139-9, 2023.
2. Li Q, Hu Z, Wang Y, Li L, Fan Y, King I, Jia G, Wang S, Song L, Li Y. *Progress and opportunities of foundation models in bioinformatics.* Brief Bioinform. *23;25(6)* doi: 10.1093/bib/bbae548, Sep 2024.
3. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin. *Attention is All you Need.* Advances in Neural Information Processing Systems 30, 2017.
4. Wang, J., *et al. scBERT: a deep learning model for parallel cell-type annotation of single-cell RNA-seq data.* Nucleic Acids Research, 50(18), e109-e109, 2022.
5. Wang, H., *et al. scGPT: towards building a foundation model for single-cell biology.* bioRxiv, .04.03.535402, *2023.*
6. *Bragin, E., Chatzimichali, E. A., Wright, C. F., Hurles, M. E., Firth, H. V., Bevan, A. P., & Swaminathan, G. J. (2017). DECIPHER: Database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation. Nature Communications, 8, 14366. https://doi.org/10.1038/ncomms14366.*
7. Riggs, E. R., Church, D. M., Hanson, K., Horner, V. L., Kaminsky, E. B., Kuhn, R. M., ... & Martin, C. L. (2017). Towards an evidence-based process for the clinical interpretation of copy number variation. *BMC Biology, 15*(1), 1–13. https://doi.org/10.1186/s12915-017-0418-y.