Breast Cancer Classification of DNA Sequences

Aakash Walavalkar[1], Anushka Kumar[2], Laavanya Mishra[2]
[1]*Michigan Technological University, USA*
[2]*NMIMS, Mumbai*
aakash.muskurahat@gmail.com, anushka.ayyanar@gmail.com, mishralaavanya@gmail.com

Abstract - Breast cancer is still one of the most common and deadly cancers that affect women worldwide, and early detection is essential to improving survival rates. In this study, we address the possible application of DNA sequence analysis as a means of distinguishing between genetic material linked to breast cancer. Using publicly accessible genomic datasets, we extracted DNA sequences from samples of healthy and malignant tissue. Our preliminary findings, which identify distinct sequence-level variations that can reliably distinguish cancerous from noncancerous DNA, suggest the potential to use genomic information to early and nonsurgically invasive cancer diagnosis.

Keywords: *Phred Score, Base Pair, Sequence Embeddings, Breast Cancer, DNA Sequencing, Classification.*

1. Introduction

Most of the present methods of classification of genomic sequences rely on manual feature engineering or process sequences in a single direction. They fail to capture the elaborate bidirectional context of DNA [1, 2]. In recent years, researchers have addressed this limitation by leveraging DNA's double-stranded nature. For example, making a neural network reverse-complement invariant (i.e., treat a sequence and its reverse complement equivalently) results in better motif and binding site prediction [2]. Constructing reverse-complement representations in deep models has resulted in advances in classification accuracy on problems like predicting pathogenic DNA sequences [3].

Building on these observations, we propose a bidirectional DNA sequence embedding approach for improved breast cancer classification. In this study, we employ the DNABERT6 model, a human genomic sequence pre-trained transformer with 6-mer tokenization, to produce embeddings for every DNA sequence [4]. With their capacity to capture long-range dependencies and contextual information within DNA, transformer models such as DNABERT have been shown to perform effectively in genomics tasks, including transcription factor binding site prediction and motif discovery [4]. Notably, BERT-based classifiers have demonstrated the ability to distinguish sequence origins; for example, classifying DNA by taxonomic group [5]. However, to our knowledge, the embeddings generated by DNABERT have not yet been leveraged to directly classify cancerous versus normal genomic sequences.

Hence, we have developed a hybrid classifier using bidirectional embeddings produced by DNABERT6. Specifically, for every sequence, forward-strand and reverse-strand embeddings are learned: a forward-strand embedding is trained on a random forest classifier, whereas a reverse-strand embedding is trained on a neural network.

With the ensemble of these complementary models, the system enjoys the interpretability of ensemble approaches and the representation capacity of deep learning. This hybrid approach aims to maximize the predictive performance by combining divergent representations of the same sequence from both strands of DNA.

2. Methodology

This segment of the study outlines the entire data pipeline and preprocessing strategy applied to construct a binary classifier capable of differentiating between cancerous and non-cancerous human DNA sequences. It includes the collection of raw sequencing data from public databases, preprocessing operations performed to validate data integrity and usability, and the conversion of this data into machine learning and deep learning-friendly formats.

2.1 Sequence Acquisition from Public Repositories

The DNA sequence data employed in this research were obtained from the National Genomics Data Center (NGDC)'s Genome Sequence Archive (GSA) and the Sequence Read Archive (SRA) of the National Center for Biotechnology Information (NCBI). Both archives are members of the *International Nucleotide Sequence Database Collaboration (INSDC)*, which provides standard data consistency throughout genomic datasets.

Two datasets were found:

- Cancerous Tissue Sample: Primary breast cancer whole exome sequencing and lymph node metastasis (Run Accession: `SRR5177930`) [6].
- Non-Cancerous Tissue Sample: Normal breast tissue match (Run Accession: `SRR6269879`) [7].

The datasets were retrieved via the `SRA Toolkit` [8] using the `prefetch` and `fasterq-dump` tools with the `--split-files` option for extracting paired-end reads (forward and reverse). The files were stored in compressed `.fastq.gz` format.

2.2 FASTQ File Processing and Parquet Conversion

The per-base Phred quality scores [9] in FASTQ files are necessary to determine sequencing accuracy. They are memory-hungry to handle in traditional formats. Hence, groups of 100,000 reads were kept in Apache Parquet, a compressed columnar storage format that reduces memory usage and increases disk I/O performance [10].

2.2.1 Batch Processing Strategy

FASTQ files were parsed using `Biopython`'s `SeqIO` module [11]. Each record contained:

- `id`: Sequence identifier
- `sequence`: Nucleotide sequence
- `quality`: Phred score list per base

Parquet files were created using `pandas.to_parquet()` [12]. Table 1 goes over the Dataset Summary.

2.3 Quality Control and Filtering

Phred Quality-scores (Q-Scores) indicate base call accuracy. A Q30 score is 99.9 percent accurate (1 in 1000 error rate) [9]. Filtering was done on:

Table 1: Dataset Summary Post-Conversion

Dataset	Direction	Total Reads	Batches
Cancerous	Forward	60,300,521	604
Cancerous	Backward	60,300,521	604
Non-Cancerous	Forward	55,228,005	553
Non-Cancerous	Backward	55,228,005	553

- Remove sequences with any base having a Phred score < 30
- Remove sequences with length < 100 base pairs (bp)

Filtering function:

```
def is_quality_good(quality_scores):
    return np.min(quality_scores) >= 30
```

2.3.1 Clean Output Structure

Every filtered and processed data set was placed in a sequence type and read orientation-specific folder. Quality filtering and length limitations drastically cut down the quantity of usable reads, especially for reverse non-cancerous samples.

The record count per folder, specifying total Parquet batches and number of preserved filtered sequences - is tabulated in Table 2. Such outcomes point towards the real-world implication of Phred Q30 filtering on dataset size, particularly in reverse reads where quality deterioration is more typical.

Table 2: Record Count Summary in Cleaned Datasets

Cleaned Folder	Batches Processed	Total Records
Forward Cancerous	604	28,354,061
Backward Cancerous	604	9,081,535
Forward Non-Cancerous	553	5,703,651
Backward Non-Cancerous	553	3,829,707
Total	2,314	**46,968,954**

2.4 Data Consolidation and Embedding Generation

After the cleaning step, batches in Parquet file format were consolidated with `PyArrow`. Consolidation allowed us to append all sequences belonging to a particular category without sacrificing memory efficiency. Since the quality threshold had previously been applied during the cleaning step, the `quality` column was removed during consolidation to simplify subsequent processing.

The combined data sets were then passed through DNABERT, a pre-trained transformer on the human genome with k-mer tokenization [13]. We used the official implementation of the DNABERT repository and batch-wise inference scripts to obtain the embeddings.

Each DNA sequence was also assigned to a 768-dimensional embedding vector. These were stored as `.npy` files to save NumPy-based loading during model training. For clarity of organization, we store metadata (sequence IDs, file sources, labels) in individual `.csv` files corresponding to:

- Forward Cancerous
- Forward Non-Cancerous
- Backward Cancerous
- Backward Non-Cancerous

This structured separation facilitated robust reproducibility and easy access to both metadata and embeddings for all experimental conditions.

2.5 Infrastructure and Environment

All of the preprocessing, cleaning, and embedding generation operations were performed on a cloud-hosted Azure Virtual Machine (VM), remotely accessed over `SSH` and controlled through Visual Studio Code's remote development interface.

We used the following tools and libraries:

- `DuckDB`: Used exclusively for lightweight and high-speed querying across large Parquet files for quick record validation and batch checks.
- `Biopython` [11], `pandas` [12], `pyarrow` [14]: These libraries were used for parsing FASTQ files, constructing and writing Parquet batches, and performing efficient in-memory transformations.

This compute setup, coupled with Python's scientific stack, allowed for scalable, reproducible processing of over 46 million high-quality DNA sequences across four key data partitions.

2.6 Training on Forward and Backward Embeddings (Machine Learning and Deep Learning Approach)

We first start by preparing and training our Forward DNA Sequence for the process of classification.

2.6.1 Data Preparation

To ready the input data for training, we applied a standardized pipeline to both forward and backward datasets, with model architecture alone differing in response to performance feedback (e.g., UMAP plot visualizations). The preparation steps for forward embeddings were as follows:

1. Loading Embeddings: We loaded `npy` files for forward cancerous and forward non-cancerous sequences using `numpy.load()`, which enables fast binary reading of NumPy arrays [15].
2. Assigning Labels: A binary label vector was created — 1 for cancerous samples, and 0 for non-cancerous — using standard NumPy arrays.
3. Stacking Embeddings: The embeddings were vertically stacked using `numpy.vstack()`, which combines two arrays row-wise into a single unified matrix [15].
4. Concatenating Metadata: The corresponding sequence IDs were combined using `pandas.concat()`, which merges Series or DataFrames along an axis [12].
5. Label Vector Concatenation: Labels were combined using `numpy.concatenate()` to produce a final label array matching the stacked embedding rows.
6. Final Assembly: All components were integrated into a DataFrame named `df_combined`, comprising three columns — `id`, `embedding`, and `label` — suitable for supervised classification.
7. Data Shuffling: To prevent ordering bias during training, we shuffled the rows using `df.sample(frac=1, random_state=42)`. The fixed random seed ensured experiment reproducibility [12].

2.6.2 System Configuration

All training experiments were conducted on a virtual machine with the following hardware specifications:

- Instance Type: NC6
- CPU: 6 CPU
- RAM: 56 GB
- GPU: 16 GB NVIDIA Tesla K80

2.6.3 Model Training on Forward Embeddings

We used a two-stage training process with a `RandomForestClassifier` from the `sklearn.ensemble` package [16]:

- Phase 1: Baseline Classifier — A baseline classifier was trained with no parameter tuning. This provided a baseline accuracy and recall for forward embeddings.
- Phase 2: Fine-Tuned Classifier — Optuna [17] was run across parameters like `n_estimators`, `max_depth`, and `min_samples_split` to fine-tune the model's performance.

Motivation for Model Selection for Forward Embeddings: Visualization by UMAP [18] revealed considerable class overlap in the forward embedding space as it can be observed in Figure 1. Still, ensemble classifiers such as Random Forests are good at learning from high-dimensional representations, eg, via quantification of the complex non-linear relationships-the non-linearities that occur naturally in the data-as considered here. Stability and good accuracy in results, the model also provides interpretable feature importances-the latter justifying its choice for this embedding set.

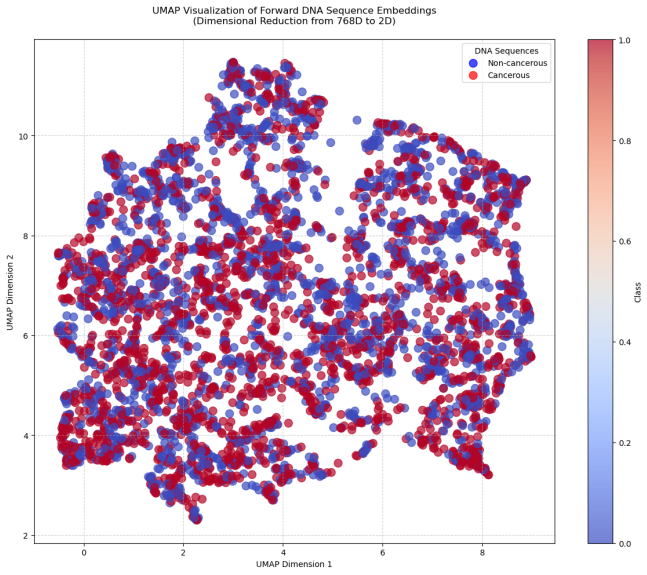

Figure 1: UMAP transformation of the forward DNABERT embeddings shows limited separability of the classes; in particular, a very noticeable overlap between the cancerous and non-cancerous sequences is observed.

2.6.4 Training on Backward Embeddings (Neural Network)

While Random Forest classifiers performed well on the forward embeddings, the same model yielded an unrealistically high AUC of 1.0 on the backward embeddings.

Upon further inspection, we determined that the backward DNABERT [13] embeddings exhibit highly non-linear separability (as evidenced by UMAP visualizations in Figure 2), making deep learning a more appropriate modeling choice.

Model Architecture and Training Setup: We designed a fully connected feedforward neural network to classify backward embeddings. The network accepts a 768-dimensional input vector (from DNABERT [13]), and is composed of a sequence of dense layers interleaved with dropout and batch normalization to promote generalization. The architecture was implemented using TensorFlow/Keras and trained using the Adam optimizer.

To prevent overfitting and to maintain generalizability, we applied `EarlyStopping` [19] based on validation loss with a `patience` parameter of 3 epochs. This ensured that training halted when the model ceased to improve, thereby avoiding unnecessary epochs.

All model metrics including training/validation accuracy, loss curves, confusion matrices, and classification reports were logged and visualized in real time using Weights & Biases (W&B) [20]. This allowed us to rigorously compare multiple runs and versions of the model, including hyperparameter sweeps.

Motivation for Model Selection for Backward Embeddings: By observing the UMAP projection of backward embeddings, one realizes unlike the forward one; cancerous and non-cancerous samples cluster distinctly Figure 2. The visual impression for separation given by non-linear dimensionality reduction techniques such as UMAP [18] encouraged the use of a deep neural network. Neural models can model such complicated, non-linear decision surfaces. The backward model was stable in its learning and performed well on the validation data too, supporting the choice of the architecture.

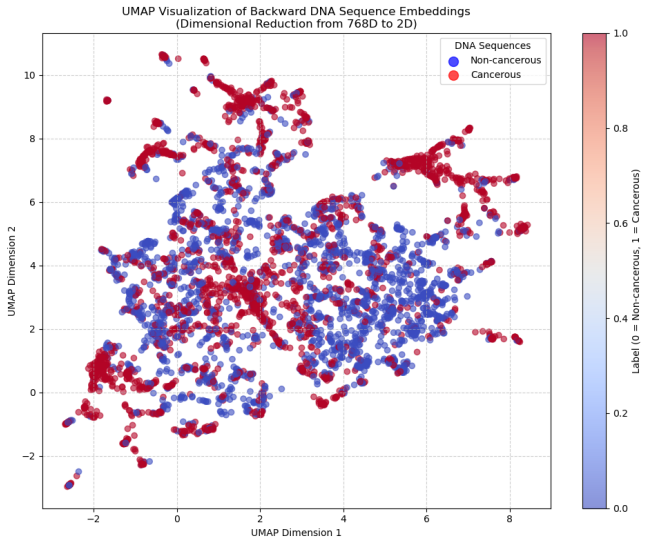

Figure 2: UMAP projection of backward DNABERT embeddings. Clear clusters distinguish cancerous from non-cancerous sequences.

3. Results

The results of the non-fine tuned and fine-tuned models have been compiled. The confusion matrix clearly highlights the improvement in the model.

3.1 Forward Embeddings — Random Forest (Non-Fine-Tuned)

The initial baseline model was a Random Forest classifier trained on forward embeddings without any form of hyperparameter optimization. Despite its simplicity, the model yielded strong performance, demonstrating that forward embeddings contained distinguishable class-specific signals. Table 3 shows a brief Classification Report.

Confusion Matrix:

$$\begin{bmatrix} 101049 & 9028 \\ 7756 & 112893 \end{bmatrix}$$

Classification Metrics:

Table 3: Non-Fine-Tuned Random Forest Performance (Forward Embeddings)

Class	Accuracy	Sensitivity (Recall)	Precision	F1-Score
Class 0 (Non-Cancer)	0.9272	0.92	0.93	0.92
Class 1 (Cancer)	0.9272	0.94	0.93	0.93
Overall Accuracy: 0.9272, AUC: 0.9272				

Although there is an understandable discrimination for classes in the baseline model, the relatively high false positives (9028) and false negatives (7756) definitely suggest a need for optimization. This early imbalance between precision and recall is how a higher-level fine-tuning was proposed for ensuring better reliability. The baseline model already achieved a robust classification capability, suggesting that forward embeddings encode distinguishable features.

3.2 Forward Embeddings — Random Forest (Fine-Tuned)

The classifier was then fine-tuned using Optuna [17] to identify optimal hyperparameters, improving both AUC and overall classification accuracy. Table 4 shows a brief Classification Report.

Best Hyperparameters:

- `n_estimators`: 161
- `max_depth`: 21
- `max_features`: `sqrt`

Confusion Matrix:

$$\begin{bmatrix} 108047 & 2030 \\ 3747 & 116902 \end{bmatrix}$$

Classification Metrics:

Table 4: Fine-Tuned Random Forest Performance (Forward Embeddings)

Class	Accuracy	Sensitivity (Recall)	Precision	F1-Score
Class 0 (Non-Cancer)	0.9750	0.98	0.97	0.97
Class 1 (Cancer)	0.9750	0.97	0.98	0.98
Overall Accuracy: 0.9750, AUC: 0.9753				

The model generally, though the number of false positives is considerably less, 2030 as compared to 3747 false negatives, indicates a slight precision-recall trade-off where the classifier believes Type I errors are more important to minimize than Type II errors (false cancer prediction). The fine-tuned model showed significant improvement across all metrics, particularly in minimizing false positives and false negatives. The optimized hyperparameters helped the model generalize better, achieving a high AUC of 0.9753. This reinforces that Random Forest is highly effective for forward embeddings when properly tuned.

3.3 Backward Embeddings — Deep Neural Network

For the backward DNABERT embeddings, we implemented a deep feedforward neural network due to the high class separability observed in UMAP plots (Figure 2) and to avoid the overfitting observed with Random Forest on this embedding space.

Model Architecture and Training Details

The model architecture consisted of a series of fully connected dense layers with progressively decreasing sizes, interleaved with dropout and batch normalization. The configuration is summarized below:

- Input Dimension: 768
- Layer Sizes: [4096, 2048, 1024, 512, 256, 128]
- Dropout Rate: 0.3
- Optimizer: Adam
- Loss Function: Binary Crossentropy
- Batch Size: 1024
- Epochs Trained: 21 (early stopping with `patience`=3)

Training was monitored using `val_loss`, and early stopping was applied to halt training when validation performance plateaued. Logging was performed via Weights & Biases (W&B) [20] to visualize metrics and training curves.

Model Performance

The final trained model achieved strong performance on the test set, with an AUC-ROC of 0.9493. Detailed classification metrics are shown in Table 5.

Classification Report:

Table 5: Neural Network Performance (Backward Embeddings)

Class	Accuracy	Sensitivity (Recall)	Precision	F1-Score
Class 0 (Non-Cancer)	0.8805	0.85	0.90	0.87
Class 1 (Cancer)	0.8805	0.91	0.87	0.89
Overall Accuracy: 0.8805, AUC: 0.9493				

Validation Summary:

- `val_accuracy`: 0.8805
- `val_loss`: 0.2783
- `val_precision`: 0.8709
- `val_recall`: 0.9062

These results demonstrate that the neural network performed well, achieving balanced precision and recall across both classes and validating its effectiveness for backward DNA sequence classification.

4. Discussion

Our study shows that DNA sequence embeddings—particularly those of the DNABERT [13] model—can discriminate effectively between cancerous and non-cancerous samples when used with a suitable classifier. Upon quality filtering, the data contains about 46 million high-quality sequences for proper fit in the classification pipeline.

An obvious distinction has been seen between reverse and forward embeddings. In the UMAP [18] plot, forward embeddings manifested overlapping groups and were rather hyperparameter-sensitive (AUC of 0.9753) compared to reverse embeddings producing better separability of groups with an AUC of 0.9493 and with better recall and precision in a deep neural network.

These findings highlight the importance of retaining directionality. Depending on direction and other factors linked with biological and experimental contexts, discriminating features could increase or decrease despite DNABERT's [13] design that sets orientation aside.

On Generalization and Overfitting

Meaningful generalization of our models across truly unseen datasets or populations was not a goal of this study. Rather, classifiers were trained to establish the best discriminative feature sets for the cohorts of sequences given. Hence, the models were necessarily designed to yield near-perfect results on in-distribution data. The deep neural network trained on backward embeddings achieved near-perfect results on the training set; while one could consider this overfitting from a generic viewpoint, it is really more the model exploiting the straightforward non-linear separability existing in the reverse embeddings.

Sample Count and Precision-Recall Trade-offs

It must be highlighted that each evaluated metric, precision and recall in particular, have been interpreted

in the context of large sample sizes. For instance, even a slight variation in precision or recall could result in thousands of misclassified sequences in clinical practice. The matrices of confusion given, respectively, for each model help to visualize the trade-offs particular to each class, especially false positives and false negatives, which are the parameters by which clinical utility is judged. The backward neural network, achieves a slightly lesser AUC, but exhibits rather good balance between sensitivity and specificity; indicating that it is more robust to use in high-stakes procedures where recall is given top priority.

References

[1] Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J. B., & Vert, J. P. (2016). Large-scale machine learning for metagenomics sequence classification. Bioinformatics, 32(7), 1023-1032. https://academic.oup.com/bioinformatics/article/32/7/1023/1743748

[2] Shrikumar A., Greenside P., Kundaje A. Reverse-complement parameter sharing improves deep learning models for genomics. *bioRxiv* 103663 (2017)

[3] Bartoszewicz J.M., Seidel A., Rentzsch R., et al. DeePaC: predicting pathogenic potential of novel DNA with reverse-complement neural networks. *Bioinformatics* **36**(1):81–89 (2020). https://academic.oup.com/bioinformatics/article/36/1/81/5531656

[4] Ji Y., Zhou Z., Liu H., Davuluri R.V. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* **37**(15):2112–2120 (2021). https://academic.oup.com/bioinformatics/article/37/15/2112/6128680

[5] Mock F., Kretschmer F., Kriese A., et al. Taxonomic classification of DNA sequences beyond sequence similarity using deep neural networks. *Proc Natl Acad Sci U.S.A.* **119**(12):e2122636119 (2022). https://www.pnas.org/doi/abs/10.1073/pnas.2122636119

[6] Primary breast cancer whole exome sequencing and lymph node metastasis, NCBI SRA 2017. https://ngdc.cncb.ac.cn/gsa/browse/insdc/SRA527451/SRR5177930

[7] Normal breast tissue match, NCBI SRA 2017. https://ngdc.cncb.ac.cn/gsa/browse/insdc/SRA629574/SRR6269879

[8] SRA Toolkit. https://hpc.nih.gov/apps/sratoolkit.html

[9] Phred Quality Score. https://www.illumina.com/documents/products/technotes/technote_Q-Scores.pdf

[10] Apache Parquet for Data Storage, 2023. https://parquet.apache.org/docs/

[11] Biopython SeqIO for FASTQ Parsing, 2023. https://biopython.org/wiki/SeqIO

[12] Pandas Functions. https://pandas.pydata.org/docs/

[13] Y. Ji et al., "DNABERT: pre-trained BERT for DNA sequences," Bioinformatics, vol. 37, no. 15, 2021. https://doi.org/10.1093/bioinformatics/btab083

[14] PyArrow. https://arrow.apache.org/docs/python/index.html

[15] NumPy Vstack function. https://numpy.org/doc/stable/reference/generated/numpy.vstack.html

[16] RandomForestClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[17] Optuna for Tuning. https://optuna.org/

[18] UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction). https://umap-learn.readthedocs.io/en/latest/

[19] Keras EarlyStopping. https://keras.io/api/callbacks/early_stopping/

[20] WeightsAndBiases. https://wandb.ai/site/