RNA BIOLOGY
2024, VOL. 21, NO. 1, 1–15
https://doi.org/10.1080/15476286.2024.2352192

RESEARCH PAPER

OPEN ACCESS Check for updates

# Nm-Nano: a machine learning framework for transcriptome-wide single-molecule mapping of 2′-O-methylation (Nm) sites in nanopore direct RNA sequencing datasets

Doaa Hassan[a,b], Aditya Ariyur[a], Swapna Vidhur Daulatabad[a], Quoseena Mir[a], and Sarath Chandra Janga[a,c]

[a]Department of Biohealth Informatics, Luddy School of Informatics, Computing, and Engineering, Indiana University Indianapolis (IUI), Indianapolis, Indiana, USA; [b]Computers and Systems Department, National Telecommunication Institute, Cairo, Egypt; [c]Centre for Computational Biology and Bioinformatics, Indiana University School of Medicine, Indianapolis, Indiana

**ABSTRACT**

2′-O-methylation (Nm) is one of the most abundant modifications found in both mRNAs and noncoding RNAs. It contributes to many biological processes, such as the normal functioning of tRNA, the protection of mRNA against degradation by the decapping and exoribonuclease (DXO) protein, and the biogenesis and specificity of rRNA. Recent advancements in single-molecule sequencing techniques for long read RNA sequencing data offered by Oxford Nanopore technologies have enabled the direct detection of RNA modifications from sequencing data. In this study, we propose a bio-computational framework, Nm-Nano, for predicting the presence of Nm sites in direct RNA sequencing data generated from two human cell lines. The Nm-Nano framework integrates two supervised machine learning (ML) models for predicting Nm sites: Extreme Gradient Boosting (XGBoost) and Random Forest (RF) with K-mer embedding. Evaluation on benchmark datasets from direct RNA sequecing of HeLa and HEK293 cell lines, demonstrates high accuracy (99% with XGBoost and 92% with RF) in identifying Nm sites. Deploying Nm-Nano on HeLa and HEK293 cell lines reveals genes that are frequently modified with Nm. In HeLa cell lines, 125 genes are identified as frequently Nm-modified, showing enrichment in 30 ontologies related to immune response and cellular processes. In HEK293 cell lines, 61 genes are identified as frequently Nm-modified, with enrichment in processes like glycolysis and protein localization. These findings underscore the diverse regulatory roles of Nm modifications in metabolic pathways, protein degradation, and cellular processes. The source code of Nm-Nano can be freely accessed at https://github.com/Janga-Lab/Nm-Nano.

**ARTICLE HISTORY**
Revised 25 March 2024
Accepted 1 May 2024

**KEYWORDS**
RNA modification detection; Nm (2′-O-methylation); Single molecule direct RNA sequencing; Oxford Nanopore Technology; Machine Learning; Transcriptomics

## 1. Introduction

2′-O-methylation (or Nm, where N denotes any nucleotide) is a or post-transcriptional modification of RNA, occurring when a methyl group ($-CH_3$) is added to the 2′ hydroxyl (–OH) of the ribose moiety. This modification can occur on any nucleotide, regardless of the type of nitrogenous base. Nm is an abundant modification found frequently in mRNAs and at multiple locations in non-coding RNAs, such as transfer RNA (tRNA), ribosomal RNA (rRNA), small nuclear RNA (snRNA) and piwi-interacting RNA (piRNA) [1–4]. This abundance is due to the role that internal Nm modification of mRNA plays as a new mechanism of genetic regulatory control, with the ability to influence mRNA abundance and protein levels both in vitro and in vivo [5].

The Nm modification has a great contribution in many biological processes, such as the normal functioning of tRNA [6], protecting mRNA from degradation by the decapping and exoribonuclease (DXO) protein [7], and the biogenesis and specificity of rRNA [8,9]. Additionally, it has been found that Nm modification has been associated with many human diseases (e.g. cancer and autoimmune diseases) and has potential indirect links to some other biological defects [10].

Detecting Nm modifications in RNAs has been a great challenge for many years, with various experimental methods presented in the literature [10]. However, each of these methods has exhibited significant limitations. For example, RiboMethseq [11,12] was introduced as a sequencing-based method for mapping and quantifying Nm modifications based on a simple chemical principle – the considerable difference in nucleophilicity between a 2′-OH and a 2′-O-Me. This method uses a proprietary ligation protocol for direct ligation to 5′-OH and 3′-P ends, followed by alkaline fragmentation to prepare RNA for sequencing. The read-ends of library fragments are used for mapping with nucleotide resolution and calculation of the fraction of molecules methylated at the Nm sites. However, the relative inefficiency of the ligation protocol imposes substantial amounts of input RNA (>1 µg), which requires increasing the sequencing depth. Thus, to address this limitation, another chemical method called RibOxi-

**CONTACT** Sarath Chandra Janga scjanga@iu.edu Department of Biohealth Informatics, Luddy School of Informatics, Computing and Engineering, Indiana University Indianapolis (IUI), 535 West Michigan St, Indianapolis, Indiana 46202, USA

Supplemental data for this article can be accessed online at https://doi.org/10.1080/15476286.2024.2352192

seq was presented for detecting Nm modifications in RNAs [13]. Using this method, Nm sites can be mapped after the ligation of linkers to the Nm-modified nucleotide at the 3′-end. However, this method identified significantly fewer Nm modification sites compared to those reported by LC-MS/MS , a method to detect and quantify the relative abundance of RNA modifications [14,15]. Despite LC-MS/MS providing industry standard results, it is time-consuming and labour-intensive, requireing large amount of input RNA, although can detect low-abundant nucleotides [16]. Recently, Dai et al. introduced a sensitive high-throughput experimental method called Nm-seq, which can detect Nm sites at low stoichiometry, especially in mRNAs with single-base resolution, achieving outstanding detection of Nm modifications [17].

However, in general, the experimental methods are naturally costly due to the high labour effort involved. Therefore, there have been relatively few computational biology methods proposed in the literature to overcome the limitations of experimental methods for detecting RNA Nm modifications [18,19]. These computational methods mainly rely on developing machine/deep learning classification algorithms to identify Nm sites in RNA sequences based only on short read data and have not yet been applied to long reads. Long reads, which can sequence over 10 kb on average in a single read, offer an advantage by requiring fewer reads to cover the same gene. For instance, a support vector machine (SVM)-based method was presented in [18] to identify Nm sites in RNA short read sequences of the human genome by encoding RNA sequences using nucleotide chemical properties and nucleotide compositions. This model was validated by identifying Nm sites in Mus musculus and *Saccharomyces cerevisiae* genomes. Another research work presented in [19] proposed a deep learning-based method for identifying Nm sites in RNA short read sequences. In this approach, dna2vec – a biological sequence embedding method originally inspired by the word2-vec model of text analysis – was adopted to yield embedded representations of RNA sequences that may or may not contain Nm sites. These embedded representations were fed as features for a Convolutional Neural Network (CNN) to classify RNA sequences into those modified with Nm sites or those not modified. The method was trained using the data collected from Nm-seq experimental method. Another prediction model using RF to identify Nm sites in short read RNA sequences was presented in [20]. This model was trained with features extracted by multi-encoding scheme combination that combines the one-hot encoding with position-specific dinucleotide sequence profile and K-nucleotide frequency encoding.

Recently, third-generation sequencing technologies, such as the platforms provided by Oxford Nanopore Technologies (ONT), have been proposed as a new means to detect RNA modifications on long RNA sequence data [21]. However, to our knowledge, this technology has only been applied in two studies [22,23] for detecting Nm modifications. In [22], the main goal was to predict the stoichiometry of Nm-modified sites in yeast mitochondrial rRNA using 2-class (Nm-modified or unmodified) classification algorithms deployed in a tool called nanoRMS [22]. This tool used the characteristic base-calling 'error' signatures in the Nanopore data as features for training supervised or unsupervised learning models to identify the stoichiometry of Nm sites, using a threshold for base mismatch frequency in different types of RNAs in yeast. However, nanoRMS was not applied to predict Nm sites in the RNA sequences of human cell lines, which are larger and more complex than yeast. Additionally, the single read features used to train the predictors of nanoRMS were averaged before Nm prediction, making it infeasible to obtain the contribution of each feature in predicting Nm sites. Moreover, relying on base-calling errors for detecting RNA modifications, as in the nanoRMS implementation, might decrease with the advances of developing high-accuracy Nanopore base-calling algorithms. In [23], a dual-path framework called HybridNm was proposed to predict Nm subtypes in one human cell line (HEK293) based on features extracted from RNA short reads sequenced with Illumina and RNA long reads sequenced with ONT to improve the prediction of Nm sites. Therefore, this framework did not purely rely on ONT technology for predicting Nm sites in RNA sequences. Moreover, the base-calling errors were used as features to distinguish Nm from unmodified sites, which again might decrease the performance of accurately predicting Nm sites with the advances of developing high-accuracy Nanopore base-calling algorithms. To this end, our work aims to extend this research direction and address nanoRMS and HybridNm limitations by combining ML and ONT to identify Nm sites in long RNA sequence reads of human cell lines based on features extracted from raw Nanopore signals. We have developed a framework called Nm-Nano that integrates two different supervised ML models (predictors) to identify Nm sites in Nanopore direct RNA sequencing reads from HeLa and HEK293 cell lines, namely the XGBoost and RF with K-mer embedding models (Figure 1a,b). The developed predictors integrated in the Nm-Nano framework for identifying Nm sites have been trained and tested using a dataset of both Nm-modified and unmodified Nanopore signals. These signals were generated by passing both 'modified' RNA sequences containing Nm sites at known positions (identified using the standard Nm-seq experimental method [17]) and 'unmodified' sequences through the ONT MinION device.

By deploying Nm-Nano to predict Nm sites in Nanopore direct RNA sequencing reads from HeLa and HEK293 cell lines, we performed various types of biological analysis (Figure 1c), such as identifying unique Nm genomic locations/genes, identifying the most frequently modified RNA bases with Nm sites, and performing functional, and gene set enrichment analysis of identified Nm-genes in both cell lines.

## 2. Results

When evaluating the performance of Nm-Nano predictors, we used two validation methods: random-test splitting and integrated validation testing. In the former, the benchmark dataset of the HeLa cell line (see Subsection 4.2 in Methods section) is randomly divided into two folds: one for training and another for testing. The test size parameter for this method was set to 0.2, which means 80% of the benchmark dataset is used for training the Nm-Nano ML model while 20% of the dataset is reserved for testing. In the latter, a combination of two benchmark datasets for two different cell lines, HeLa and HEK293, was used, where 50% of this

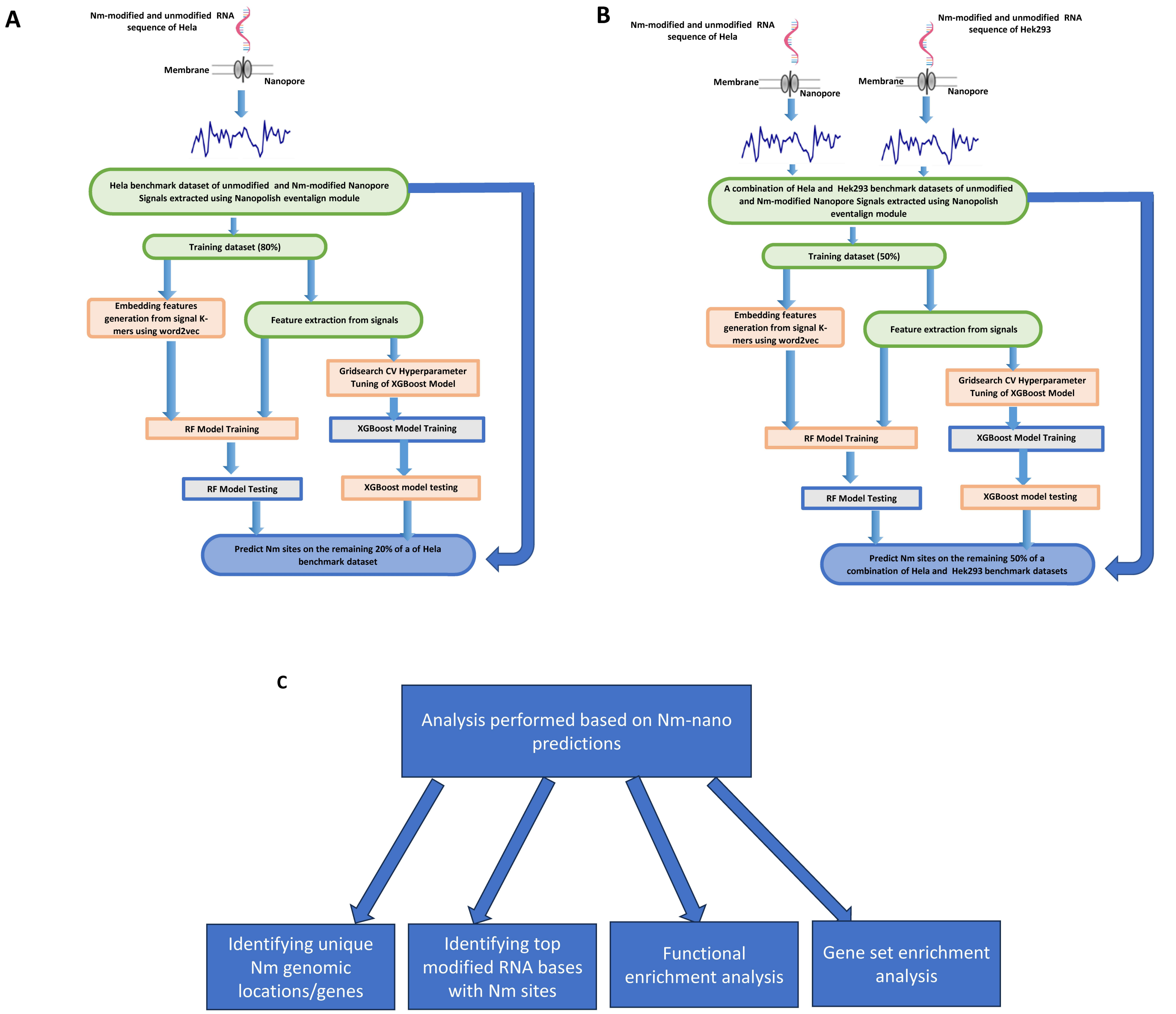
A
Nm-modified and unmodified RNA sequence of Hela
Membrane
Nanopore
Hela benchmark dataset of unmodified and Nm-modified Nanopore Signals extracted using Nanopolish eventalign module
Training dataset (80%)
Embedding features generation from signal K-mers using word2vec
Feature extraction from signals
Gridsearch CV Hyperparameter Tuning of XGBoost Model
RF Model Training
XGBoost Model Training
RF Model Testing
XGBoost model testing
Predict Nm sites on the remaining 20% of a of Hela benchmark dataset
B
Nm-modified and unmodified RNA sequence of Hela
Nm-modified and unmodified RNA sequence of Hek293
Membrane
Nanopore
Membrane
Nanopore
A combination of Hela and Hek293 benchmark datasets of unmodified and Nm-modified Nanopore Signals extracted using Nanopolish eventalign module
Training dataset (50%)
Embedding features generation from signal K-mers using word2vec
Feature extraction from signals
Gridsearch CV Hyperparameter Tuning of XGBoost Model
RF Model Training
XGBoost Model Training
RF Model Testing
XGBoost model testing
Predict Nm sites on the remaining 50% of a combination of Hela and Hek293 benchmark datasets
C
Analysis performed based on Nm-nano predictions
Identifying unique Nm genomic locations/genes
Identifying top modified RNA bases with Nm sites
Functional enrichment analysis
Gene set enrichment analysis

**Figure 1.** The Nm-Nano framework for predicting Nm sites on (a) HeLa cell line using random 80/20 train/test split (b) 50% of the combination of HeLa and HEK293 benchmark dataset using integrated validation testing with random 50/50 train/test split on this combination (c) analysis performed based on Nm-Nano predictions.

**Table 1.** The performance of Nm-Nano predictors on HeLa benchmark dataset with random-test splitting.

| Classifier | Accuracy% | Precision | Recall | AUC |
|---|---|---|---|---|
| XGBoost | 99 | 0.99 | 0.99 | 0.99 |
| RF | 92.39 | 0.9 | 0.96 | 0.92 |

combined dataset is used for training the Nm-Nano ML model and the remaining 50% is reserved for testing.

### 2.1. Performance evaluation with random-test splitting

Table 1 shows the performance of XGBoost and RF with K-mer embedding ML models implemented in the Nm-Nano framework, when applied to the HeLa benchmark dataset. As the table shows, both models perform very well in detecting Nm sites. However, the XGBoost model outperforms the RF with K-mer embedding model in terms of accuracy, precision, recall and Area Under the Curve (AUC).

The learning (Figure 2a,d) and loss (Figure 2b,e) curves of XGBoost and RF with K-mer embedding, respectively, show that the performance of XGBoost, in terms of accuracy score and misclassification error, outperforms that of RF with

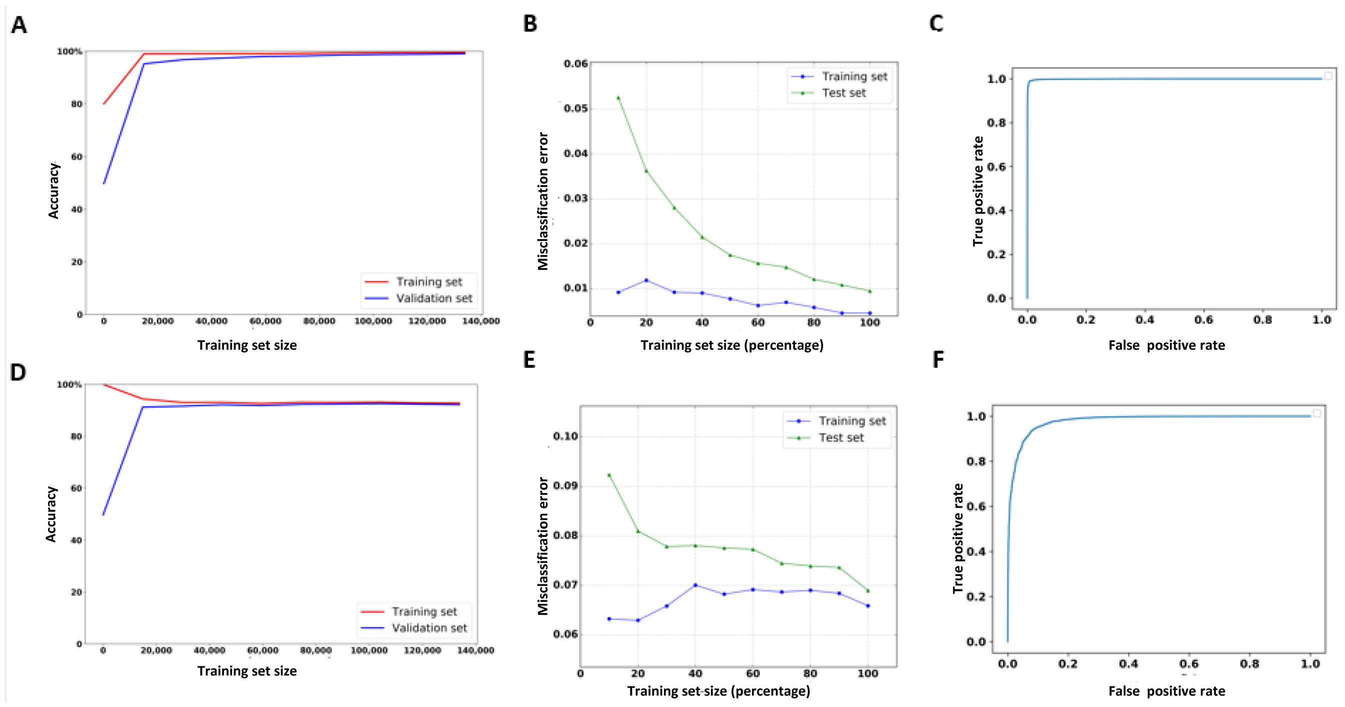

**Figure 2.** The learning, loss and ROC curves of Nm-nano predictors validated on HeLa benchmark dataset with random split testing, where 80% of data is used for training and the remaining 20% is kept for testing. (a, b, and c) XGBoost model and (d, e, and f) RF with K-mer embedding model.

K-mer embedding. Additionally, the receiver operating characteristic (ROC) curves of XGBoost and RF with K-mer embedding (Figure 2c,f, respectively) show that the percentage of true positive rate to false-positive rate in the case of XGBoost model exceeds that of the RF with K-mer embedding model. Supplementary file 1_test_split_HEK293_results.docx and Figure 1_test_split_hek show the performance of XGBoost and RF with K-mer embedding ML models with random test split on HEK293 benchmark dataset.

Table 2 shows the performance of Nm-Nano ML models with random test-splitting on HeLa benchmark dataset in terms of accuracy with each of the extracted features (introduced later in Subsection 4.3) as well as the embedding features generated using word2vec embedding technique (introduced later in Subsection 4.4). Clearly, the position feature contributes more to the classifiers' accuracy than the other extracted features used for training either the XGBoost or RF with embedding ML models. It is followed by the model mean, then K-mer match features in the case of XGboost and K-mer match then model mean features in case of RF with K-mer embedding. It was also observed that the event/signal standard deviation (Event_stdv) feature achieves the lowest contribution to the performances of XGBoost and RF with embedding models. Furthermore, Table 2 shows that the embedding features generated by the word2vec technique strongly contribute to the performance of RF, as these features follow the most contributing feature (i.e. position). Despite the success of these features in improving the performance of RF, they were not used to train the XGBoost model. This is because this model achieved high detection accuracy of Nm sites of 99% by tuning its parameters with grid search algorithm [24], which takes considerable time to obtain the best values for the parameters. Therefore, generating more features with word2vec embedding techniques and combining them with the other features used for training the grid-search XGBoost model will introduce additional processing overhead. This is due to the time required for the word2vec technique to generate embedding features, in addition to the time consumed by the grid search algorithm for hyperparameter tuning of XGBoost model. This would significantly slow down the performance of XGBoost when applied to the benchmark dataset of a given cell line. In other words, the slight improvement in XGBoost's performance would not be proportional to the substantial increase in the processing time. Table 3 shows the performance of XGBoost (when the model is trained solely with the extracted features) compared to the performance of XGBoost with K-mer embedding (when the

**Table 2.** The performance of Nm-Nano predictors on HeLa benchmark dataset in terms of accuracy (%) with random test-splitting using single type of feature.

| Classifier | Posi-tion | Event_mean | Event_stdv | Model_mean | Model_stdv | K-mer_ match | Mean_diff | K-mer embed-ding |
|---|---|---|---|---|---|---|---|---|
| XGBoost | 93.88 | 54.26 | 50.65 | 83.36 | 64.14 | 75.27 | 51.58 | - |
| RF | 89.83 | 54.7 | 51.44 | 72.65 | 64.14 | 75.27 | 51.58 | 84.87 |

**Table 3.** The performance of XGBoost versus the performance of the XGBoost with K-mer embedding model applied to HeLa benchmark dataset with random test-splitting.

| Classifier | Accuracy | Precision | Recall | AUC | Execution time (Secs) |
|---|---|---|---|---|---|
| XGBoost | 99 | 0.99 | 0.99 | 0.99 | 43.81 |
| XGBoost with K-mer embedding | 99 | 0.99 | 1 | 0.994 | 608.2 |

model is trained using a combination of the extracted features and the embedding reference K-mers features), along with corresponding execution time in seconds for each case.

## 2.2. Performance evaluation with integrated validation testing

Table 4 shows the performance of ML models with integrated validation testing, wherein Nm-Nano's predictors are applied to 50% of the combined HeLa and HEK293 benchmark datasets during the training phase and tested on the remaining 50% of this combination in the testing phase. As the results indicate, both models perform very well in predicting Nm sites, although the XGBoost model outperforms the RF with K-mer embedding model. The learning (Figure 3A,D) and loss (Figure 3B,E) curves of XGBoost and RF with K-mer embedding, respectively, show that the performance of XGBoost, in terms of accuracy score and misclassification error, outperforms the performance of RF with K-mer embedding. Additionally, the receiver operating characteristic (ROC) curves of XGBoost and RF with embedding (Figure 3C,E, respectively) show that the percentage of true positive rate to false-positive rate in the case of XGBoost model exceeds that of the RF with embedding model.

**Table 4.** The performance of Nm-Nano predictors on a combination of HeLa & HEK293 benchmark datasets with 0.5 random-test splitting.

| Classifier | Accuracy% | Precision | Recall | AUC |
|---|---|---|---|---|
| XGBoost | 98.58 | 0.99 | 0.99 | 0.99 |
| RF | 91.63 | 0.89 | 0.96 | 0.92 |

Table 5 shows the performance of ML models with integrated validation testing in terms of accuracy with a single type of feature. This was achieved by testing the performance of Nm-Nano predictors with each of the extracted features, as well as the embedded features generated using word2vec embedding technique. Clearly, the features generated using the word2vec embedding technique strongly contribute to the RF classifier accuracy, as these features follow the most contributing feature (i.e. position). However, they were not considered for training the grid search XGBoost model. Again, this is due to the extra processing overhead resulting from combining the time taken for generating embedded features by word2vec technique and the time taken by grid search algorithm to obtain the best parameter values of XGBoost, as mentioned in subsection 2.1. Regarding the contribution of each of the seven extracted features, it was observed that the position feature achieved the best contribution to the performance of XGBoost or RF among all extracted features, followed by model mean feature, then the K-mer match feature in the case of XGBoost, and K-mer match feature, then model mean feature in the case of RF with K-mer embedding. Also, it was observed that the event/signal standard deviation (event

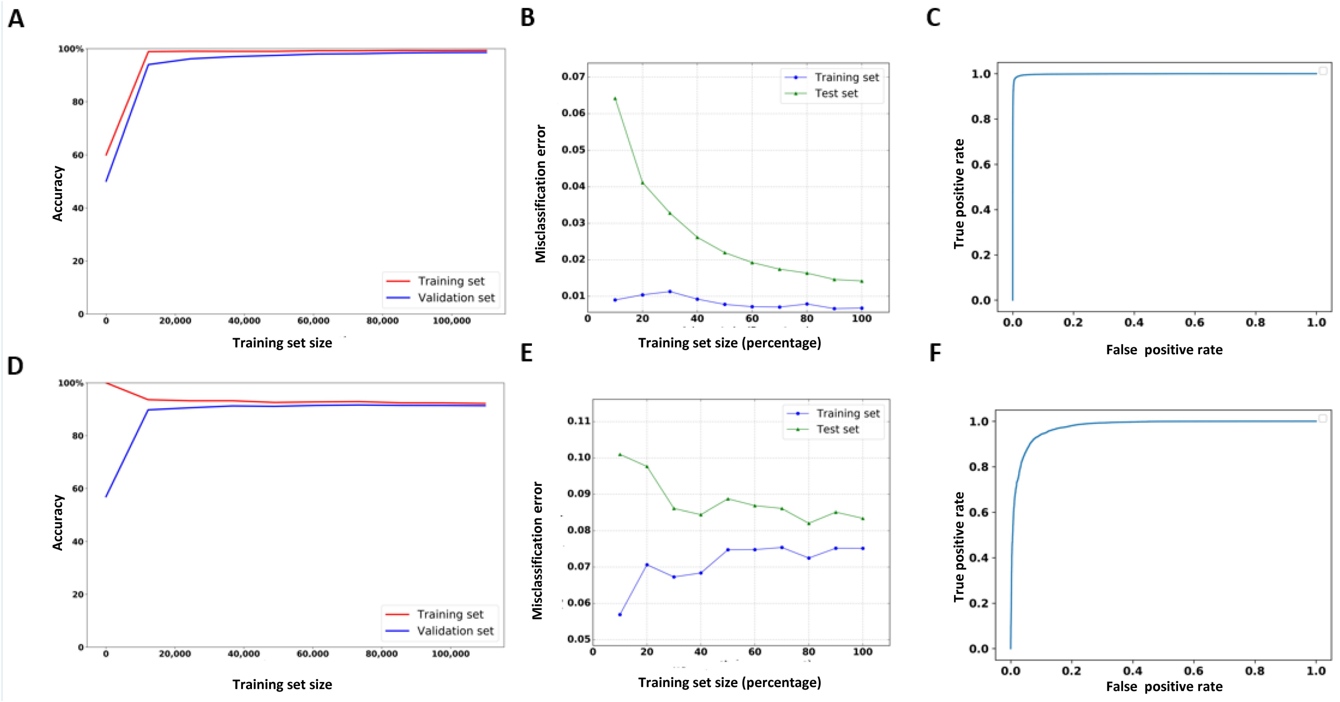

**Figure 3.** The learning, loss and ROC curves of Nm-Nano predictors in integrated validation testing, where 50% of the combination of HeLa and HEK293 benchmark datasets was used for training and the remaining 50% was used for testing. (a,b, and c) XGBoost model and (d, e, and f) RF with K-mer embedding model.

**Table 5.** The performance of nm-nano predictors with integrated validation testing in terms of accuracy (%) using a single type of feature.

| Classifier | Position | Event_mean | Event_stdv | Model_mean | Model_stdv | K-mer_ match | Mean_diff | K-mer embedding |
|---|---|---|---|---|---|---|---|---|
| XGBoost | 94.62 | 53.41 | 51.31 | 81.22 | 62.24 | 75.14 | 51.69 | - |
| RF | 85.45 | 54.19 | 51.71 | 72.54 | 62.24 | 75.14 | 51.7 | 82.92 |

_stdv) and Mean_diff features have the lowest contribution to the performance of either XGBoost or RF with K-mer embedding models.

## 2.3. Abundance of Nm sites

To determine the abundance of Nm sites in the RNA sequences of either HeLa or HEK293 cell lines, we first run the XGBoost model (since it outperforms the RF with K-mer embedding model) on the complete RNA sequence reads of both cell lines. Next, we identify all samples with predicted Nm sites in those reads, followed by identifying the number of unique genomic locations of Nm corresponding to those Nm predictions, as well as their frequencies in both cell lines. We found that there are 11,651,518 Nanopore signal samples predicted as samples with Nm sites from a total of 920,643,073 Nanopore signal samples that represent the complete HeLa cell line with 1,674,369 unique genomic locations of Nm (Supplementary Table 1_Nm_unique_genomic_locations _HeLa). Similarly, we found that there are 1,712,344 Nanopore signal samples predicted as samples with Nm sites from a total of 275,056,668 samples that present the complete RNA sequence of HEK293 cell line with 291,382 unique genomic locations of Nm modification (Supplementary Table 2_Nm_unique_genomic_locations _hek). The reference K-mers corresponding to modified Nanopore signals with Nm predictions in HeLa and HEK293 cell lines can be identified as strong K-mers compared to the reference K-mers corresponding to the unmodified/control Nanopore signals, which can be considered as weak contributors to the Nm prediction. The frequency of these strong reference K-mers provides an overview of their abundance and shows their contribution to Nm predictions in HeLa and HEK293 cell lines, available in Supplementary Tables 3_Nm_unique_reference_kmer_freq_HeLa and 4_Nm_unique_reference_kmer_freq_hek, respectively. Also, SFigures 2_top_10-modified_bases_HeLa and 3_top_10-modified_bases_hek provide the sequence logo for the top ten modified bases corresponding to Nm prediction in HeLa and HEK293 cell lines, respectively.

We found that there are 105,678 modified genomic locations shared between HeLa and HEK293 cell lines (Figure 4A). Additionally, we observed that there were 10 genes shared across the top 1% of Nm-modified(Figure 4B). Clearly, we notice that the extent of Nm modifications (the number of Nanopore signal samples predicted as containing Nm sites over the total number of Nanopore signal samples) in RNA sequences from the HeLa cell line is higher than its counterpart in the HEK293 cell line (1.27% for HeLa versus 0.62% for HEK293). Therefore, the distribution of Nm across normalized gene length for HeLa cell line is higher than its equivalent in HEK293 cell line (Figure 4C). Additionally, and as a primary observation of Figure 4C, we found that Nm modifications are likely to be more prevalent in the 3' region compared to the 5' region when observed at a transcriptomic level. This distribution reinforces our previous observation that RNA modifications, such as pseudouridine, tend to favour the 3' region over the 5' region [25].

Since Nm modifications can occur at any RNA base, we have also reported the percentage of unique Nm locations occurring for each of the four RNA bases in the two complete cell lines of HeLa and HEK293 (Table 6).

## 2.4. Functional enrichment analysis

A total of 61 genes from the HEK293 cell line and 125 genes from the HeLa cell line were identified as the top 1% of frequently modified Nm genes with the highest abundance of Nm modification. These short-listed genes from both cell lines were then plugged into the Cytoscape ClueGo [26] application to obtain enriched ontologies and pathways with high confidence ($p < 0.05$). Enrichment observations from this analysis are visualized in Figure 5A,B for HEK293 and HeLa cell lines, respectively.

From the functional enrichment analysis of the top 1% gene set from the HEK293 cell line (Figure 5A, and SFigure 4), we observed a wide range of functional processes, such as 'glycolysis/gluconeogenesis', 'regulation of protein localization to cell surface', and 'aggrephagy' being significantly enriched. This highlights the diverse regulatory role of Nm modifications, from their involvement in metabolic pathways to protein degradation and localization.

In the HeLa cell line, we observed several enriched ontologies with high confidence (adjusted p-val <0.05) that were more representative of the Nm modification's role in immune response and cellular processes (Figure 5B and SFigure 5), such as 'C3HC4-type RING finger domain binding', 'antigen processing and presentation (class I MHC)', and 'cytoplasmic translational initiation'.

To investigate which cellular pathways were associated with Nm modifications, we ranked the complete human gene lists from both the HeLa and HEK293 cell lines based on the occurrence of Nm modification locations and performed gene set enrichment analysis (GSEA) [27] using WebGestalt [28]. Across both cell lines, we observed that genes associated with metabolic processes, protein binding, and biological regulation were enriched in these ranked lists, reinforcing the association between Nm modification and RNA–protein interaction, as previously observed in literature [29,30]. The Nm-modified gene sets from both cell lines exhibited enrichment (NES > 1.2) for pathways associated with autoimmune, signalling, and diabetes as highlighted in SFigure 6. Additionally, we observed enrichment of tissue-

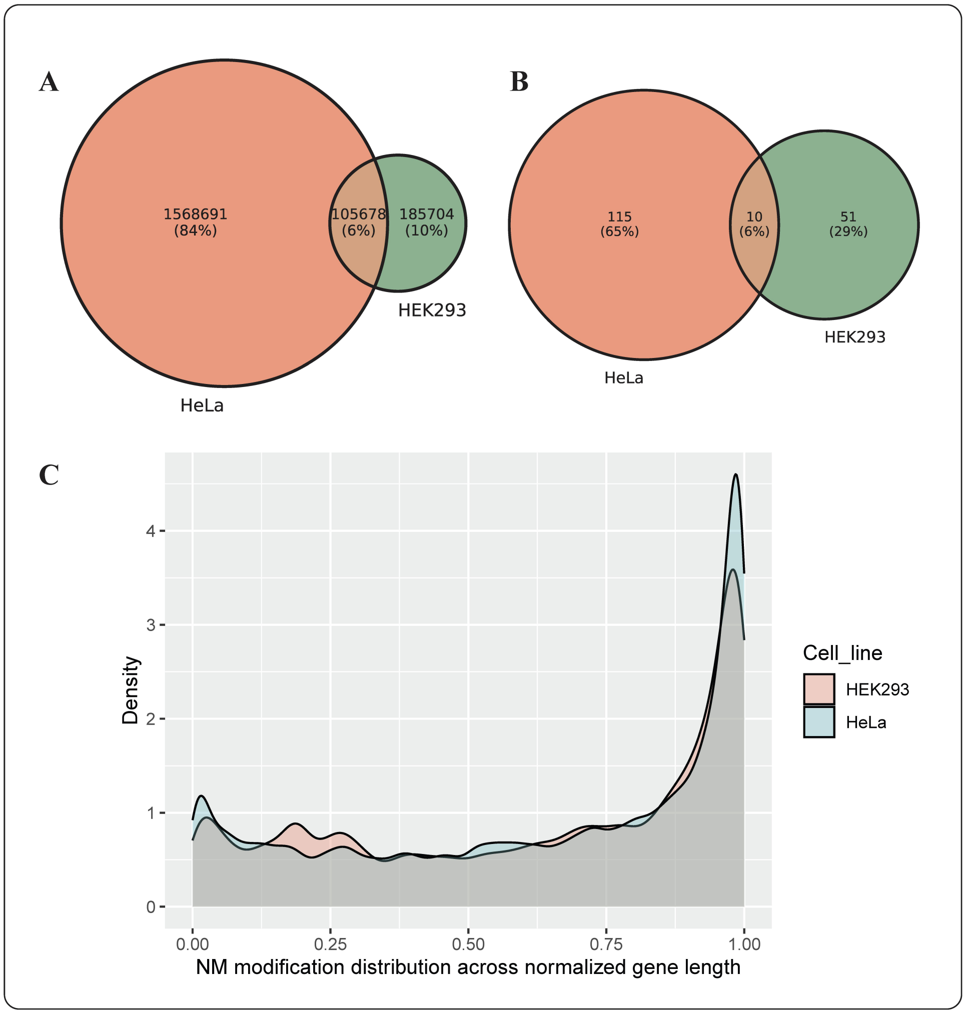

**Figure 4.** (a) The overlap between unique Nm locations in the complete dataset of HEK293 and HeLa cell lines (b) The overlap between most frequent (top 1%) modified Nm genes in HEK293 and HeLa cell lines (c) A density plot illustrating Nm modifications' distribution across normalized gene length for HEK293 and HeLa cell lines.

**Table 6.** The percentage of unique nm locations occurring for each of the four RNA bases in the HeLa and HEK293 cell lines.

| Cell line | A base | C base | G base | U base |
|---|---|---|---|---|
| HeLa | 29.75% | 21.54% | 22.91% | 25.81% |
| HEK293 | 26.18% | 25.14% | 26.12% | 22.56% |

specific/disease pathways, such as prion disease in HeLa cells and inflammatory bowel disease in HEK293 cells. These pathways were enriched with high normalized enrichment scores (NES >1.2).

## 3. Discussion

We observed that Nm-Nano, compared to existing non-Nanopore tools for Nm site prediction in the literature [18,19], achieves higher accuracy. Specifically, the accuracies were 50.1% for [17], 81.91% for [18], 84.8% for [19], and 99% for XGBoost, the best Nm-nano ML model, using a 1:1 ratio of positive and negative test samples. However, we found that comparing Nm-Nano with these tools may not be meaningful in terms of implementation but is significant in terms of accuracy. This is because these tools were only applied to predict Nm sites in short reads of RNA sequences, whereas Nm-Nano can predict Nm sites in long reads of RNA sequences. Moreover, these tools were trained and tested on the same dataset, and they were not tested for predicting Nm sites in a combination of two benchmark datasets of RNA sequence data from two different cell lines.

When comparing the performance of Nm-Nano with nanoRMS [22], an existing Nanopore-based tool for predicting Nm modifications in direct RNA sequencing data, we found that nanoRMS was only tested on predicting Nm in direct RNA sequences of yeast; it was not tested for predicting the Nm sites in direct RNA sequencing data of human cell lines, which are more complex than lower eukaryotes like yeast. Hence, directly comparing the accuracy of nanoRMS on human cell-line data is not feasible. In addition, nanoRMS predicts Nm sites on individual single reads of direct RNA sequence data, where the single read features are used to train the predictors of nanoRMS. These features are averaged before Nm prediction, making it impossible to assess the contribution of each feature in predicting Nm sites. Moreover, nanoRMS only reports accuracy values for

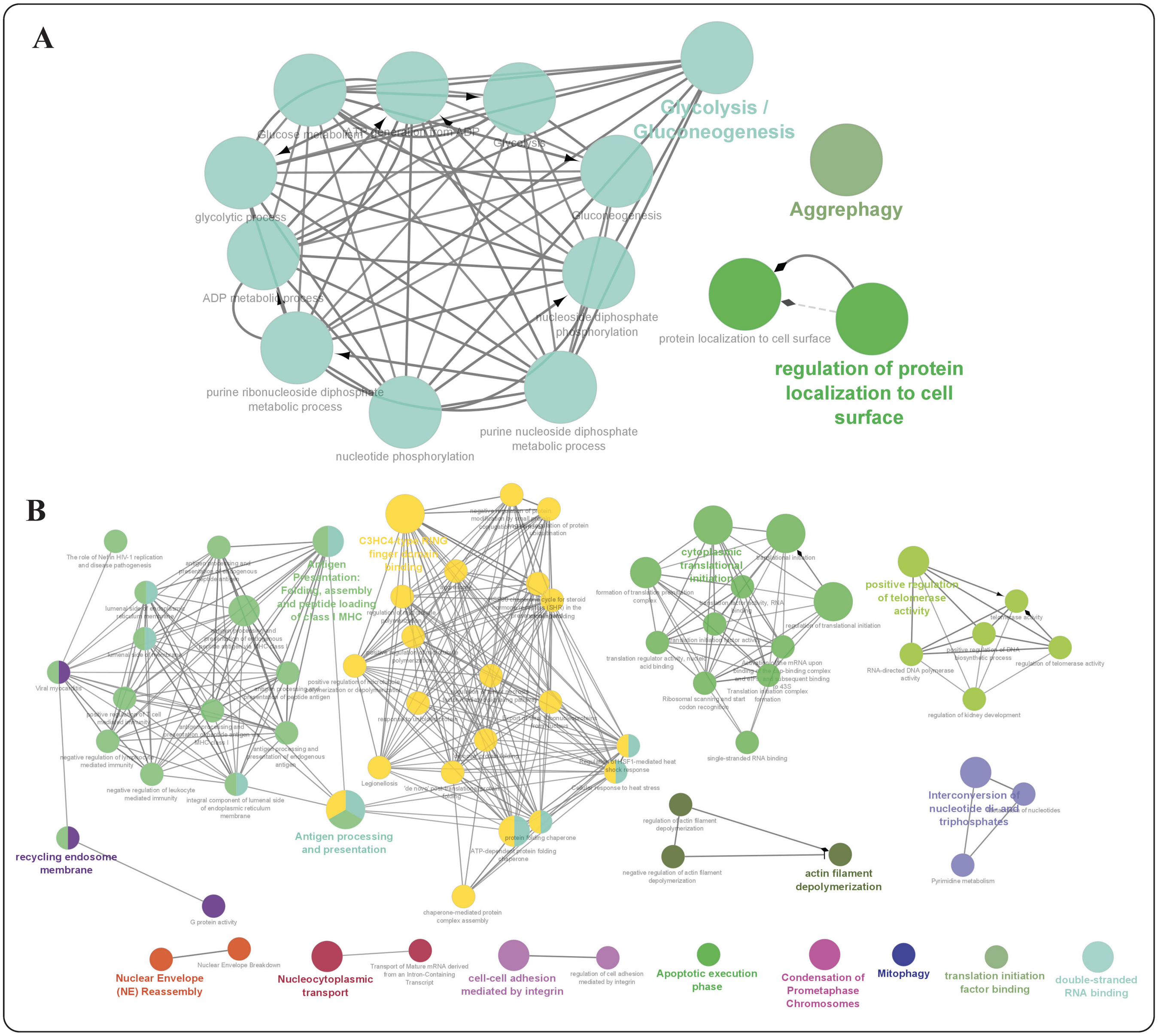
A
Glycolysis / Gluconeogenesis
Glucose metabolism
ATP generation from ADP
Glycolysis
glycolytic process
Gluconeogenesis
ADP metabolic process
nucleoside diphosphate phosphorylation
purine ribonucleoside diphosphate metabolic process
nucleotide phosphorylation
purine nucleoside diphosphate metabolic process
Aggrephagy
protein localization to cell surface
regulation of protein localization to cell surface
B
The role of Nef in HIV-1 replication and disease pathogenesis
antigen processing and presentation of endogenous peptide antigen
Antigen Presentation: Folding, assembly and peptide loading of class I MHC
lumenal side of endoplasmic reticulum membrane
antigen processing and presentation of endogenous peptide antigen via MHC class I
lumenal side of membrane
Viral myocarditis
antigen processing and presentation of peptide antigen
positive regulation of T cell mediated immunity
antigen processing and presentation of peptide antigen via MHC class I
antigen processing and presentation of endogenous antigen
negative regulation of lymphocyte mediated immunity
negative regulation of leukocyte mediated immunity
integral component of lumenal side of endoplasmic reticulum membrane
Antigen processing and presentation
recycling endosome membrane
G protein activity
C3HC4-type RING finger domain binding
negative regulation of protein modification by small protein conjugation or removal
negative regulation of protein ubiquitination
aggresome
HSP90 chaperone cycle for steroid hormone receptors (SHR) in the presence of ligand
protein refolding
regulation of microtubule polymerization
positive regulation of microtubule polymerization
positive regulation of microtubule polymerization or depolymerization
regulation of tumor necrosis factor-mediated signaling pathway
response to unfolded protein
Export of Viral Ribonucleoproteins from Nucleus
'de novo' protein folding
Legionellosis
'de novo' post-translational protein folding
Regulation of HSF1-mediated heat shock response
Cellular response to heat stress
protein folding chaperone
ATP-dependent protein folding chaperone
chaperone-mediated protein complex assembly
cytoplasmic translational initiation
translational initiation
formation of translation preinitiation complex
translation factor activity, RNA binding
regulation of translational initiation
translation initiation factor activity
translation regulator activity, nucleic acid binding
Activation of the mRNA upon binding of the cap-binding complex and eIFs, and subsequent binding to 43S
Translation initiation complex formation
Ribosomal scanning and start codon recognition
single-stranded RNA binding
positive regulation of telomerase activity
telomerase activity
positive regulation of DNA biosynthetic process
regulation of telomerase activity
RNA-directed DNA polymerase activity
regulation of kidney development
Interconversion of nucleotide di- and triphosphates
Metabolism of nucleotides
Pyrimidine metabolism
regulation of actin filament depolymerization
negative regulation of actin filament depolymerization
actin filament depolymerization
Nuclear Envelope Breakdown
Nuclear Envelope (NE) Reassembly
Nucleocytoplasmic transport
Transport of Mature mRNA derived from an Intron-Containing Transcript
cell-cell adhesion mediated by integrin
regulation of cell adhesion mediated by integrin
Apoptotic execution phase
Condensation of Prometaphase Chromosomes
Mitophagy
translation initiation factor binding
double-stranded RNA binding

**Figure 5.** Functional enrichment analysis of genes with the highest frequency of Nm modifications within a specific cell line, grouped based on the functional hierarchy of Gene Ontology (GO) terms using the Cytoscape ClueGO application. (a) HEK293 cell line and (b) HeLa cell line (visualizing high confidence (p-val <0.05) ontologies and pathways potentially associated with nm RNA modification. The size of the nodes is representative of the significance of association with respect to genes per GO-term.

predicting the stoichiometry of Nm for each read, lacking other performance metrics like precision and recall. Therefore, we can only compare the average accuracy values of the k-nearest neighbour (KNN) predictor, the best supervised classifier employed in nanoRMS, to the accuracy of the predictors integrated in Nm-Nano. Based on this comparison, each of the two ML models employed in Nm-Nano significantly outperforms the average accuracy of KNN predictor (66.17%). Regarding the implementation comparison between Nm-Nano and nano-RMS, we found that nano-RMS relies on base-calling 'error' signatures in the Nanopore data as features for detecting the Nm-modification. However, those base-calling errors might not be the same for each type of modified K-mer context resulting from base-calling as not all modified bases can be detected as base-calling errors. So, the generated benchmark dataset used in nanoRMS might be biased. Moreover, as Nanopore technology advances, base-callers are expected to become more accurate, potentially resulting in lower base-calling errors and smaller training datasets for tools like nanoRMS. This could lead to decreased performance of the ML models deployed in nanoRMS for predicting Nm sites. In other words, relying on the erroneous base-calling of Nanopore RNA sequencing for generating a training data on which ML models applied for detecting Nm modification is a challenge. This is either because not all the modified bases in K-mers contexts resulting from base-calling process would generate base-calling errors which might result in biased training or validation dataset, or because of the high accuracy of base-calling might lead to generating limited-size of training data. In summary, relying solely on base-calling errors for detecting RNA modifications may become obsolete as base-callers approach 100% accuracy.

Finally, we compared the performance of Nm-Nano predictors with that of HybirdNm, proposed in [23]. However, we found that the accuracies of HybirdNm for predicting Nm sites were not explicitly mentioned. Instead, we found that AUC was a common metric used to evaluate the performance of both Nm-Nano and HybirdNm, so we used this metric for comparison. The best AUC achieved by HybirdNm was 0.962 for predicting the Um subtype, with an average AUC of 0.917 for predicting all four subtypes. This is less than the AUC values obtained by either XGBoost (AUC of 0.991) or RF with K-mer embedding (AUC of 0.957) within the Nm-Nano framework when applied to the HEK293 benchmark dataset with random test splitting (supplementary file 1_test_split_HEK293_results. docx). Additionally, the HybirdNm framework uses base-calling errors in the Nanopore data as a feature for predicting Nm subtypes, which may not be ideal once Nanopore base-callers reach optimal performance. Moreover, HybirdNm was trained and tested on the same benchmark dataset of the HEK293 cell line and was not tested for predicting Nm sites on a combination of two benchmark datasets of RNA sequence data for two different cell lines.

Thus, Nm-Nano offers significant advantages over existing computational tools for detecting Nm-sites, summarized as follows:

(1) Nm-Nano is designed to predict the presence of Nm sites in Nanopore direct RNA sequencing reads of human cell lines. This addresses the limitations of previous Nm predictors that only detected these sites in short reads of RNA sequencing data of cell lines from different species or long read sequencing data from non-human cell lines like yeast.

(2) Nm-Nano has advantages over other ONT frameworks for predicting Nm sites, namely nanoRMS and HybridNm. The former was applied to predict Nm sites in RNA sequences from yeast, neglecting human cell lines which are larger and more complex. The latter was proposed to only predict Nm **subtypes** in a single human cell line (HEK293) and was not tested on multiple cell lines.

(3) Nm-Nano was developed to rely solely on ONT-based single molecule direct RNA sequencing data to predict Nm-sites at individual read-level resolution. This gives the tool a significant advantage over non-pure ONT frameworks like HybridNm, which was proposed as a dual-path framework to predict Nm subtypes in HEK293 human cell line based on features from RNA short reads sequenced with Illumina and RNA long read sequenced with ONT, to improve the prediction of Nm sites in single molecules of RNA transcripts.

(4) Nm-Nano investigates the contribution of each feature, unlike nanoRMS in which single read features used to train its predictors are averaged before Nm prediction, making it not feasible to assess the contribution of each feature in predicting Nm sites.

(5) Nm-Nano does not rely on base-calling errors for Nm-site detection in RNA sequences. Therefore, its implementation will not be affected negatively from advancements in high-accuracy Nanopore base-calling algorithms, which would affect the implementation of nanoRMS and HybridNm for detecting Nm sites in RNA sequences.

Supplementary file (Nm-nano_advantages.docx) tabulates the advantages of Nm-Nano over existing ONT methods for predicting Nm modifications.

We also found that the position and sequence features used to train Nm-Nano models have been explored in other research by Zhang et al. [31] and Haung et al. [32] to enhance the performance of prediction of DNA sequencing depth and m6A RNA modifications, respectively. In the work of Zhang et al., they used local features, such as sequence and base probability, to predict DNA sequencing depth, which differs from our objective of predicting RNA modification sites. Nonetheless, similar to their approach, we have incorporated sequencing features in the form of K-mers context among the features used for predicting Nm-sites. However, we found that base probability is not a relevant feature for our main objective, which focuses on detecting Nm-modified sites, but it was a relevant feature in Zhang et al. ‘s work to predict sequence depth by quantifying the frequency of a particular nucleotide being read during the sequencing process.

In Haung et al'.s work, the geographic representation of transcript as vectors (Geo2vec) scheme has exhibited strong interpretability and was applicable to m6A and N1-methyladenosine (m1A) but was not applied or tested on Nm modifications. Based on this, we found that Geo2vec and K-mer-embedding, used by the RF model in our approach, share similarities in terms of vector representation. However, while Geo2vec is a vector representation of the geographic presence on a transcript, K-mer embedding is vector representation of K-mer sequences. We also found that Geo2vec explored different strategies for encoding sub-molecular geographic information of ribonucleotides, capturing the position of the target ribonucleotide (or site) relative to transcript landmarks, like our approach, which also used the position as a feature for predicting Nm sites. Thus, our approach leads to a similar observation as Huang et al'.s work, where either the position feature or the encoding that captures it, contributes greatly to RNA modification detection.

It has also been shown that Nm-nano predictors exhibit high accuracy in both the random test-split applied on individual HeLa or HEK293 benchmark datasets and the integrated validation test applied on combined HeLa and HEK293 benchmark datasets. However, a significant decrease in the performance of Nm-Nano predictors was observed during validation with an independent cell line, when one cell line is used for training the Nm-Nano predictor and another for testing it. For instance, RF and XGBoost achieved accuracies of 66% and 59%, respectively, in detecting Nm-sites on the HEK293 benchmark dataset after training on the HeLa benchmark dataset and achieved Nm detection accuracies of 57.26% and 56%, respectively in the inverse cross validation. This clear decrease in cross validation and inverse cross-validation accuracies in detecting Nm sites is likely due to the small dataset size of the Nm-seq data [17] used to generate the benchmark training dataset for HeLa and HEK293. This small dataset size leads to increased specific differences between both cell lines, resulting in decreased Nm prediction accuracy when tested on an independent cell line. In addition, it is possible that not all cell line-specific features were captured when trained on individual cell-line datasets, further lowering cross-validation accuracies. This discrepancy was not observed in the integrated validation testing of Nm-Nano predictors, which achieved high accuracy in detecting Nm sites by training on 50% of the combined benchmark datasets from HeLa and HEK293 and testing them on the remaining 50%.

It was also observed that employing Nm-Nano on direct RNA sequencing data of HeLa and HEK293 cell lines leads to identifying top frequently modified Nm genes associated with various biological processes. However, it may be unclear how the enrichment of specific functional families for only two considered human cell lines (HeLa and HEK293) would strengthen the confidence in the Nm-Nano's predictions. To address this, we looked at publicly available direct RNA sequencing data for human cell lines on SRA and found data for multiple cell lines. However, this data is only available in fastq files, not fast5 files, which our algorithm needs for signal-level analysis. Thus, it is not possible at this point to run our algorithm on additional cell lines available in the public domain. However, studying the extent of Nm sites across multiple cell lines to understand common and unique sites is an exciting question, which we believe can be addressed as more cell line-based direct RNA sequencing datasets in the form of fast5 files with signal data are publicly available in the future. Finally, it is worth noting that the current study is limited to detecting the Nm modification in mRNA due to that the current protocol of Nanopore direct RNA sequencing, which is restricted to sequencing mRNA with polyA [33]. However, for other small RNAs to be captured by Nanopore sequencing, it is possible to attach polyA with a modified protocol of Nanopore RNA sequencing. Hence, mapping modifications on such small RNAs are beyond the scope of the current study, which focuses solely on mapping Nm modifications on mRNA. Nonetheless, we believe applying Nm-Nano predictors for detecting Nm sites in other small RNAs would be feasible with a modified and rigorously validated version of Nanopore RNA sequencing protocol capable of attaching other types of small RNAs to polyA.

## 4. Materials and methods

### 4.1. Computational pipeline

The complete pipeline of Nm-Nano framework for identifying Nm modifications in RNA sequence consists of several stages. The first stage begins with culturing the cell line by extracting it from an animal and letting it grow in an artificial environment. Next, the RNA is extracted from this cell during library preparation and is put through the ONT device to generate Nanopore signal data. Specifically, the MinION Mk1B device with a FLO-MIN106 flowcell was used for direct RNA sequencing of HeLa and HEK293 cell lines. The raw electrical signals output by the ONT device for each cell line are stored in the fast5 files, which are then base-called via Guppy [34] to produce fastq files containing the base-called RNA sequence reads. These reads are subsequently aligned to a reference genome using the minimap2 tool [35] to produce the SAM file, which are further processed to generate BAM and sorted BAM files using SAMtools [36], where the BAM file is a compressed version of the SAM file. Using the SAM file and a provided BED file [37], a coordinate file is generated. The BED file contains the Nm-modified locations across the whole genome that have been experimentally verified based on the research presented in [17]. This coordinate file is essential for labelling the Nanopore signal samples produced by the eventalign module as either modified or unmodified when training the two Nm-Nano predictors (the Supplementary Files 2_HeLa.txt and 3_HEK293.txt show the coordinate files generated for HeLa and HEK293, respectively). Next, the eventalign module from Nanopolish, a free software for Nanopore signal extraction and analysis [38–40], is utilized to extract Nanopore signals, which produces a dataset of Nanopore signal samples. While the structure of Nm-Nano's pipeline is similar to that of other RNA modification prediction tools [25], it differs in three phases (Figure 1A,B): benchmark dataset generation, feature extraction, and ML model construction. The benchmark dataset generation phase in Nm-Nano's pipeline is different because Nm modifications can occur at any RNA base. Therefore, all samples generated from the signal extraction process are used to identify the Nm sites using information from the coordinate file, where some of the samples are labelled as modified with Nm sites, while the remaining are control samples that are labelled as unmodified. Similarly, the feature extraction phase in Nm-Nano's pipeline is different because it uses different features (e.g. position, signal/event_mean, signal/event_stdv, model_mean, model_stdv, kmer_match, mean_diff, and word2vec embedding features of K-mers) extracted from the modified and unmodified signal samples to train the constructed ML models for predicting Nm sites. Finally, the ML model construction phase in Nm-Nano's pipeline is unique because it employs two different ML models, XGBoost with tuned parameters and RF with K-mer embedding, for predicting Nm sites in long RNA sequence reads. Further details about the differences in benchmark dataset generation, feature extraction and ML model construction will be discussed in the next subsections.

### 4.2. Benchmark dataset generation

Two different benchmark datasets were generated for the HeLa and HEK293 cell lines (Supplementary Tables 5_training_HeLa and 6_training_hek). Both datasets encompassed all Nanopore signals samples generated by passing the long RNA sequences of either HeLa or HEK293 through the ONT device, with signals extracted using the Nanopolish eventalign module. Initially, each dataset was labelled with Nm sites using a BED file containing Nm-modified locations on the whole genome that have been experimentally verified in literature based on the Nm-seq protocol. Nm-seq reported Nm sites in two different cell lines, totalling 699 Nm sites in HeLa and 2102 Nm sites in HEK293. To label each sample as Nm-modified or not, all samples generated from signal extraction were used as the target samples for identifying Nm modification, since Nm modifications can occur at any RNA base. Next, the intersection between the position column in the reference genome and the coordinate file (generated from the Nm BED file and SAM file for each cell line) determined the positive samples, with the remaining samples designated as negative. In total, the HEK293 cell line yielded 52,582 samples: 26,291 positive and 26291 negative (after sampling the negative samples which are very huge in comparison with the positive ones). Similarly, the HeLa cell line yielded 167,374 samples: 83,687 positive and 83687 negative. Analysis revealed a total of 507 and 1024 different reference K-mer combinations captured in the modified and unmodified signal datasets, respectively, for the HeLa training data (Supplementary Tables 7_training_modified_kmer_freq_HeLa and 8_training_unmodified_kmer_freq_HeLa), and a total of 238 and 1022 different reference K-mer combinations captured in the modified and unmodified signal datasets, respectively, for the HEK293 training data (Supplementary Tables 9_training_modified_kmer_freq_hek and 10_training_unmodified_kmer_freq_hek). Supplementary Figures 7_top_10-modified_bases_training_HeLa and 8_top_10-modified_bases_training_hek provide the sequence logo for the top ten modified bases corresponding to Nm prediction in the benchmark training datasets of HeLa and HEK293 cell lines, respectively.

### 4.3. Feature extraction

Each generated benchmark dataset has seven columns representing the features used to train the ML models integrated into the Nm-Nano framework. Those features are position, event_level_mean, event_stdv, model_mean, model_stdv, mean_diff, and K-mer_match. The first five features were directly extracted by selecting their corresponding columns from the eventalign's output (Supplementary File 4.txt). The sixth feature is generated by calculating the difference between the mean of the signal (event_level_mean) and the mean of the simulated signal generated by the eventalign module (model_mean). The seventh feature is generated by comparing the reference_K-mer and model_K-mer columns in the eventalign's output to determine if they match; the former represents the base-called K-mers inferred from the RNA sequence reads extracted from the Nanopore signals in the base-calling process, while the latter represents the base-called K-mers inferred from the RNA sequence reads from the simulated signals by eventalign. The value of reference and model K-mer match is 1 if they match and 0 otherwise. Additionally, it is important to note that the position feature simply refers to the genomic location of Nm modification and does not include information about the nature of nucleotide or neighbouring sequence. Therefore, training Nm-Nano predictors with this feature will not cause predictions to be highly biased towards the same conserved sequence in other RNA.

### 4.4. Feature generation with word embedding

In addition to the extracted features, embedding features have been generated by applying the word2vec technique [41] to the corpus of reference K-mers obtained from aligning Nanopore signals to a reference genome using the eventalign module of the Nanopolish software. This technique outputs a set of one-dimensional vectors of fixed size that represent the embedding features of those reference K-mers. The vector size can be optionally set as a parameter when building the word2vec embedding model.

The idea of applying word2vec to reference K-mers was inspired by the research work in [42], where word2vec was applied to DNA K-mers to generate embedding features represented by vectors of real numbers as representations of those K-mers. This approach was introduced as an alternative to the hot-one technique for vector encoding of K-mers, which is subject to the curse of dimensionality problem. With one-hot encoding, as the length of RNA sequence increases, the binary feature representation grows exponentially, resulting in an excessing number of features being added to the dataset [43].

The embedding features generated by word2vec are combined with the other extracted features introduced in the previous section for training the RF classifier model developed to predict Nm sites in long RNA sequence reads. In other words, the combination of all extracted features and embedding features is used to train the RF model, enabling it to predict whether the signal is modified by the presence of Nm sites in the testing phase.

### 4.5. ML models construction

We have developed two machine learning models for predicting Nm sites in RNA sequence reads: XGBoost [44] with tuned parameters and RF [45] with K-mer embedding. The XGBoost model parameters were tuned using the grid-search hyperparameter tuning algorithm [24]. For RF, the seed number parameter was set to 1234 and the number of trees parameter was set to 30 to obtain the best performance of RF. The XGBoost model was implemented using the optimized distributed gradient boosting Python library [46] and the RF model was implemented using scikit-learn toolkit [47], a free machine learning Python library.

#### 4.5.1. XGBoost with grid search for hyper parameter tuning

Extreme Gradient Boosted trees (XGBoost) is a special implementation of Gradient Boosting [48], a machine learning technique that produces a prediction model based on an ensemble of weak prediction models, utilizing decision trees in the case of XGBoost. This model is highly flexible and versatile, making it suitable for classification-based problems – the main goal of this study. The advantage that XGBoost has over other tree-based models is its faster training time and regularized boosting, which helps prevent overfitting – a scenario where the machine learning model becomes too accustomed to the training data, compromising its ability to generalize and predict the testing data accurately. Additionally, XGBoost, a tree-based model, does not require feature scaling, ensuring that feature scaling does not affect the split point value or the structure of the tree model. XGBoost can also cross-validate each iteration (round) of its training process, which can lead to higher results compared to models lacking this capability. The combination of decision trees and gradient boosting provides advantages over both random forest and other gradient boosting models, causing XGBoost to typically have a much lower prediction error than regular gradient boosting or random forest.

The XGBoost machine learning model was created after data preprocessing, which involved removing null values and performing feature extraction. The model has several adjustable parameters aimed at optimizing the performance. Hyperparameter tuning using the grid search algorithm was employed since it allows for the best and most accurate combination of parameters to be obtained. The parameters that were optimized for the XGBoost model were eta, gamma, max_depth, min_child_weight, and scale_pos_weight. Specifically, the optimized values obtained through grid search were 0.01, 0.1, 15, 3, and 1, respectively. Eta represents the learning rate of the XGBoost model, gamma represents how conservative the model is, max_depth represents how deep a decision tree can be built, and min_child_weight represents the minimum value needed to activate a node in the decision tree. Additionally, scale_pos_weight controls the balance between positive and negative weights and is associated with the min_child_weight. Once these optimal parameters were obtained by fitting the grid search XGBoost model to the training data, they were applied to obtain its prediction results in the testing phase.

XGBoost is trained with the features set mentioned in section 4.3. These features are extracted from the raw signal obtained through direct RNA Nanopore sequencing, along with the corresponding base-called K-mers resulted from inferring the underlying RNA sequence during base-calling.

#### 4.5.2. RF with K-mer embedding

We have developed a Random Forest (RF) ML model that has been trained using the same feature set as the XGBoost model, along with additional embedding features generated by applying word2vec embedding technique to the reference K-mers from the extracted Nanopore signals. RF algorithm has been extensively used in the literature to address several problems in bioinformatics research [49]. It has been observed that the features generated by applying the word2vec embedding technique to the reference K-mers greatly enhance the performance of the RF model, as mentioned in the results subsections 2.1 and 2.2.

The RF ML model was created after data processing, which involved removing null values, performing feature extraction, and integrating them with the generated K-mer embedding features. The K-mer embedding features were generated using genism [50], a free Python library that implements the word2vec algorithm using highly optimized C routines, data streaming, and Pythonic interfaces. The word2vec algorithm has various parameters, including vector size, window size, and word count. The vector size is the dimensionality of the vector that represents each K-mer. The window size refers to the maximum distance between a target word/K-mer and words/K-mers around the target word/K-mer. The word count refers to the minimum count of words to consider when training the model, with words occurring less than this count being ignored. The K-mer embedding features that lead to best performance of RF were generated by setting the vector size to 20, the minimum word count to 1, and the window size to 3.

### 4.6. Performance evaluation metrics

The accuracy (Acc), precision (P), recall (R), and the area under ROC curve (AUC) [51] have been used as metrics to evaluate the performance of Nm-Nano predictors. The mathematical notions for the first three metrics are as follows:

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

Where:

- TP denotes true positive and refers to the number of correctly classified Nm sites.
- FP denotes false positive and refers to the number of non-Nm sites misclassified as Nm sites.
- FN denotes false negative and refers to the number of Nm sites misclassified as non-Nm sites.
- TN denotes true negative and refers to the number of correctly classified non-Nm sites.

As for the AUC metric, it measures the entire two-dimensional area under the ROC curve [52], which measures how accurately the model can distinguish between two conditions (e.g. determining if a base in the RNA sequence is an Nm site or not).

### 4.7. Environmental settings

Nm-Nano has been developed as tool for detecting Nm modifications in Nanopore RNA sequence data by integrating two ML models: XGBoost with tuned parameters and RF with K-mer embedding to predict this type of RNA modification. XGBoost parameters were tuned to get the best performance using the grid search algorithm, which took around 6 h and 52 min to fit on the HEK293 training dataset and 9 h and 12 min to fit on the HeLa training dataset to obtain the best parameters that were applied to XGBoost model in the testing phase. The experiment was executed on a Windows 10 machine with an 8-core Ryzen 5900HS CPU and 16 GB RAM. It should be mentioned that although the grid search algorithm took considerable processing time to tune the XGBoost parameters, it significantly improved the model's performance. The performance results of XGBoost versus XGBoost tuned with the grid search algorithm are shown in Supplementary file 5_xgboost_versus_grid_search_xgboost_results.docx. Similarly, while using word2vec for generating embedding features when developing RF with K-mer embedding model added extra processing time to the RF algorithm's execution time, it also greatly improved RF performance. This is because combining the embedding features generated by word2vec with the extracted features from Nanopore signals positively affects the performance of RF as presented in result subsections 2.1 and 2.2. The performance results of RF versus RF with K-mer embedding are shown in Supplementary file 6_RF_versus_RF_with_kmer_embedding_results.docx. Meanwhile, we considered improving the performance of grid search XGBoost model by applying K-mer embedding with word2vec to generate embedding features and combine them with the extracted features used for training this model. However, we found that this would make XGBoost very slow when applied to the benchmark dataset of a given cell line, with only a slight improvement in its performance that would not justify the considerable increase in the processing time of XGBoost, as presented in result subsection 2.1.

### 4.8. Implementation and usage of Nm-Nano

The ML models within the Nm-Nano framework are implemented in Python 3.x. To run the XGBoost model, the user must type the following command from the Nm-Nano main directory on their local machine after cloning the code from the Nm-Nano GitHub repository:

```
python test_xgboost.py
```

Similarly, to run the RF with K-mer embedding model, the user should use the following command from Nm-Nano main directory:

```
python RF_embedding.py
```

To allow the user to practice with Nm-Nano predictors, we have provided a small benchmark dataset sample for HeLa cell line in the Nm-Nano GitHub repository (Nm_benchmark_HeLa_sample.csv). However, the user is encouraged to generate a benchmark dataset for other cell lines by following the instructions outlined in the README file within the generate_benchmark folder of the Nm-Nano GitHub repository.

To generate a benchmark dataset for a specific cell line, the following command should be run in the command line of a Linux environment from the generate_benchmark folder in the Nm-Nano main directory:

```
python main.py -r ref.fa -f reads.fastq
```

Where main.py is a Python script file included in the generate_benchmark folder, implemented in Python 3.x, ref.fa is the reference genome file and reads.fastq is the fastq reads file. Both ref.fa and reads.fastq files should be placed in the same path as the main.py file.

Before running the main script, the user must ensure that the folder containing the fast5 files (fast5_files) from which the reads.fastq file was generated, is in the same directory as the main.py file. Once the user runs the main.py script, it will initiate the execution of several command lines for generating the eventalign output. These command lines are documented in the generate_eventalign_output.txt file located in the generate_benchmark folder of the Nm-Nano GitHub repository. Additionally, the main.py script will call two other Python files. The first file, gen_coors_Nm.py, asks the user to enter the name of the BED file containing the Nm-modified genomic locations with the absolute path and extension to generate the coordinate file. The second file, extract_nm.py, takes as input the coordinate file and the eventalign output to extract features and generate the benchmark dataset.

To allow the user to practice with the Nm-Nano pipeline for benchmark dataset generation, we include the following in generate_benchmark folder on the Nm-Nano GitHub repository:

(1) A link to download a sample fast5 file for the HEK293 cell line, which should be placed in the fast5_files folder located in the same path as the main.py file.
(2) A sample of fastq files (reads.fastq) for HEK293 corresponding to the fast5 files in step 1.
(3) A link to download a sample reference genome (ref.fa) that should also be placed in the same path as the main.py file.
(4) A sample BED file for the HEK293 cell line (hek.bed.txt)

It should also be mentioned that the Nm-Nano framework can be extended by integrating additional ML/deep learning models for predicting Nm sites. Moreover, the framework's pipeline is generic and can be used to any direct RNA sequencing output from any ONT devices, such as MinION, GridION, and PromethION.

## 5. Conclusions

In this paper, we propose a new framework called Nm-Nano, which integrates two machine learning models: XGBoost with tuned parameters using the grid search algorithm and RF with K-mer embedding. Our results demonstrate the efficiency of the proposed framework for detecting Nm sites in RNA long reads from human cell lines. This approach addresses the limitations of existing Nm predictors presented in the literature, which were only able to detect Nm sites in RNA short read sequences from cell lines of various species, or in RNA long read sequences from non-human cell lines like yeast, and a single human cell line (HEK293).

Employing Nm-Nano on direct RNA sequencing data from HeLa and HEK293 cell lines enabled the identification of the top frequently modified Nm genes associated with various biological processes. In HeLa, we observed several high confident (adjusted p-val <0.05) enriched ontologies that were more representative of the Nm modification's role in immune response and cellular processes, such as 'C3HC4-type RING finger domain binding', 'antigen processing and presentation (class I MHC)', and 'cytoplasmic translational initiation'. Similarly, in HEK293, we observed a wide range of functional processes, such as 'glycolysis/gluconeogenesis', 'regulation of protein localization to cell surface', and 'aggrephagy' being significantly enriched, highlighting the diverse regulatory role of Nm modifications across metabolic pathways, protein degradation and localization. Thus, Nm-Nano serves as a useful computational framework for accurate and interpretable predictions of Nm sites in RNA sequences from human and other species' cell lines, offering insights into various biological findings.

## Acknowledgments

We thank Alexander Krohannon, Hunter M. Gill and Alexandre Plastow at IUI for giving valuable comments on a previous version of this manuscript.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work is supported by the National Science Foundation (NSF) grant [#1940422 and #1908992] as well as the National Institute of General Medical Sciences of the National Institutes of Health under Award Number [R01GM123314] (SCJ).

## Data availability statement

Nm-nano is available at the Github repository https://github.com/Janga-Lab/Nm-Nano. The directRNA-sequencing data generated in this study for HEK293, and HeLa cell lines are publicly available on SRA, under the project accession PRJNA685783 and PRJNA604314, respectively.

## Author contributions

DH, AA, and SCJ conceived and designed the study. DH implemented the Nm-Nano Github software version. AA and DH implemented the Nm modifications ML predictors namely XGBoost and RF with K-mer embedding respectively. DH extracted the benchmark datasets. AA tuned the parameters of XGBoost using the grid-search algorithm. AA and DH evaluated the performance of XGBoost and RF with K-mer embedding models with the random test split and integrated validation testing. DH identified the unique Nm genomic locations and the top modified RNA bases with Nm sites on HeLa and HEK293 cell lines. SVD performed gene length distribution analysis, functional and gene set enrichment analysis. QM performed the cell culturing, RNA library preparation and Nanopore RNA Sequencing for HeLa and HEK293 cell lines.

## References

[1] Xavier D, Beáta EJ, ArnoldMK, et al. Cajal body-specific small nuclear RNAs: a novel class of 20-O-methylation and pseudouridylation guide RNAs. The EMBO Journal. 2002;21 (11):2746–2756. doi: 10.1093/emboj/21.11.2746

[2] Rebane ARH, Metspalu A, Metspalu A. Locations of several novel 2´-O-methylated nucleotides in human 28S rRNA. BMC Mol Biol. 2002;3(1):1. doi: 10.1186/1471-2199-3-1

[3] Somme J, Roovers M VLB, Steyaert J, Versées W, Droogmans L. Characterization of two homologous 2'-O-methyltransferases showing different specificities for their tRNA substrates. RNA. 2014 Aug;20(8):1257–12571. doi: 10.1261/rna.044503.114 Epub 2014 Jun 20. PMID: 24951554; PMCIDPMC4105751

[4] Kurth HM, Mochizuki K. 2'-O-methylation stabilizes Piwi-associated small RNAs and ensures DNA elimination in Tetrahymena. RNA. 2009 Apr;15(4):675–685. doi: 10.1261/rna.1455509 Epub 2009 Feb 24. PMID: 19240163; PMCID: PMC2661841

[5] Elliott BA, Ranganathan HH, Vangaveti SV, et al. Modification of messenger RNA by 2'-O-methylation regulates gene expression in vivo. Nat Commun. 2019 Jul 30;10(1):3401. doi: 10.1038/s41467-019-11375-7 PMID: 31363086; PMCID :PMC6667457

[6] Guy MP, Weiner SM, Hobson CL, et al. Defects in tRNA anticodon loop 2′-O-methylation are implicated in nonsyndromic X-linked intellectual disability due to mutations in FTSJ1. Hum Mutat. 2015;36(12):1176–1187. doi: 10.1002/humu.22897

[7] Picard-Jean F, Brand C, Tremblay-Letourneau M, et al. 2′-omethylation of the mRNA cap protects RNAs from decapping and degradation by DXO. PLOS ONE. 2018;*13*(*3*:e0193804.

[8] Hengesbach M, Schwalbe H. Structural basis for regulation of ribosomal RNA 2′-o-methylation. Angew Chem Int Ed Engl. 2014;53(7):1742–1744. doi: 10.1002/anie.201309604

[9] Erales J, Marchand V, Panthu B, et al. Evidence for rRNA 2′-omethylation plasticity: control of intrinsic translational capabilities of human ribosomes. Proc Natl Acad Sci USA. 2017;114 (49):12934–9. doi: 10.1073/pnas.1707674114

[10] Dimitrova DG, Teysset L, Carré C. RNA 2'-O-Methylation (nm) modification in human diseases. Genes (Basel). 2019;10(2):117. doi: 10.3390/genes10020117

[11] Krogh NB, Nielsen H. RiboMeth-seq: profiling of 20 -O-Me in RNA. Methods Mol Biol. 2017;1562:189–209.

[12] Motorin Y, Marchand V. Detection and analysis of RNA ribose 2'-O-Methylations: challenges and solutions. Genes (Basel). 2018 Dec 18;9(12):642. doi: 10.3390/genes9120642 *PMID: 30567409; PMCID: PMC6316082*

[13] Yinzhou Zhu SPPAGGC, Pirnie SP, Carmichael GG. High-throughput and site-specific identification of 2′- O -methylation sites using ribose oxidation sequencing (RibOxi-seq). RNA. 2017 May 11;23(8):1303–1314. doi: 10.1261/rna.061549.117

[14] Yuan B-F. Liquid chromatography–mass spectrometry for analysis of RNA adenosine methylation. In: Lusser, editor. RNA methylation: methods and protocols. New York: Springer; 2017. pp. 33–42.

[15] Jora M, Lobue PA, Ross RL, et al. Detection of ribonucleoside modifications by liquid chromatography coupled with mass spectrometry. Biochim Biophys Acta, Gene Regul Mech. 2019 Mar;1862(3):280–290. doi: 10.1016/j.bbagrm.2018.10.012

[16] Anreiter I, Mir Q, JT S, SC J, Soller M. New twists in detecting mRNA modification dynamics. Trends Biotechnol. 2020 Jul 1; 39 (1):72–89. doi: 10.1016/j.tibtech.2020.06.002

[17] Dai Q, Moshitch-Moshkovitz S, Han D, et al. Erratum: corrigendum: nm-seq maps 2′-O-methylation sites in human mRNA with base precision. Nat Methods. 2018;15(3):226–227. doi: 10.1038/nmeth0318-226c

[18] Chen W, Feng P, Tang H, et al. Identifying 2′-O-methylationation sites by integrating nucleotide chemical properties and nucleotide compositions. Genomics. 2016;107(6):255–258. doi: 10.1016/j.ygeno.2016.05.003

[19] Milad Mostavi SSAYH. Deep-2′-O-Me: predicting 2′-O-methylation sites by convolutional neural networks. In: proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Honolulu, HI, USA; 2018 July.

[20] Zhou Y, Cui Q, Zhou Y. NmSEER V2.0: a prediction tool for 2′-O-methylation sites based on random forest and multi-encoding combination. BMC Bioinf. 2019;20(S25):690. doi: 10.1186/s12859-019-3265-8

[21] YK W, Hendra C, PN P, et al. Beyond sequencing: machine learning algorithms extract biology hidden in Nanopore signal data. Trends Genet. 2022 Mar;38(3):246–257. doi: 10.1016/j.tig.2021.09.001

[22] Begik O, Lucas MC, LP P, JM R, Medina R, et al. Quantitative profiling of pseudouridylation dynamics in native RNAs with nanopore sequencing. Nat Biotechnol. 2021 Oct;39 (10):1278–1291. doi: 10.1038/s41587-021-00915-6 *Epub 2021 May 13. PMID: 33986546.*

[23] Pan S, Zhang Y, Wei Z, et al. Prediction and motif analysis of 2'-O-methylation using a hybrid deep learning model from RNA primary sequence and nanopore signals. Curr Bioinf. 2022;17 (9):873–882. doi: 10.2174/1574893617666220815153653

[24] Dagnew BHSAG. Grid search-based hyperparameter tuning and classification of microarray cancer data. In: Proceedings of Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India, 2019.

[25] Hassan D, Acevedo D, Daulatabad SV, et al. Penguin: a tool for predicting pseudouridine sites in direct RNA nanopore sequencing data. Methods. 2022 Jul;203:478–487: *Epub 2022 Feb 16. PMID: 35182749; PMCID: PMC9232934*

[26] Bindea G, Mlecnik B, Hackl H, et al. ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. Bioinformatics. 2009 Apr 15;25 (8):1091–1093. doi: 10.1093/bioinformatics/btp101

[27] Subramanian A, Tamayo P, Mootha VK, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci USA. 2005 Oct 25;102(43):15545–50. doi: 10.1073/pnas.0506580102

[28] Liao Y, Wang J, Jaehnig EJ, et al. WebGestalt 2019: gene set analysis toolkit with revamped UIs and APIs. Nucleic Acids Res. 2019 Jul 2;47(W1):W199–W205. doi: 10.1093/nar/gkz401 *PMID: 31114916; PMCID: PMC6602449*

[29] YM H, Zhang X, JA H, Davis DR. An important 2'-OH group for an RNA-protein interaction. Nucleic Acids Res. 2001 Feb 15;29 (4):976–85. doi: 10.1093/nar/29.4.976 *PMID: 11160931; PMCID: PMC29614*

[30] Lacoux C, Di Marino D, Pilo Boyl P, et al. BC1-FMRP interaction is modulated by 2'-O-methylation: RNA-binding activity of the tudor domain and translational regulation at synapses. Nucleic Acids Res. 2012 May;40(9):4086–96. doi: 10.1093/nar/gkr1254 *Epub 2012 Jan 11. PMID: 22238374; PMCID: PMC3351191*

[31] Zhang JX, Yordanov B, Gaunt A, et al. A deep learning model for predicting next-generation sequencing depth from DNA sequence. Nat Commun. 2021;12(1):4387. doi: 10.1038/s41467-021-24497-8

[32] Huang D, Chen K, Song B, et al. Geographic encoding of transcripts enabled high-accuracy and isoform-aware deep learning of RNA methylation. Nucleic Acids Res. 2022 Oct 14;50(18):10290–10310. doi: 10.1093/nar/gkac830 *PMID: 36155798; PMCID: PMC9561283*

[33] Garalde D, Snell E, Jachimowicz D, et al. Highly parallel direct RNA sequencing on an array of nanopores. Nat Methods. 2018;15 (3):201–206. doi: 10.1038/nmeth.4577

[34] Basecalling using Guppy. Workflows and tutorials for longread analysis with specific focus on oxford nanopore data. Available from: https://timkahlke.github.io/LongRead_tutorials/BS_G.html

[35] Heng L. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018 Sep 15;34(18):3094–3100. doi: 10.1093/bioinformatics/bty191

[36] Danecek P, Bonfield JK, Liddle J, et al. Twelve years of SAMtools and BCFtools. Gigascience. 2021;10(2). doi: 10.1093/gigascience/giab008

[37] BED file format - Genome Browser FAQ. Available from: https://genome.ucsc.edu/FAQ/FAQformat.html#format1

[38] Loman N, Quick J, Simpson J. A complete bacterial genome assembled de novo using only nanopore sequencing data. Nat Methods. 2015;12(8):733–735. doi: 10.1038/nmeth.3444

[39] Simpson J. Aligning nanopore events to a reference. 2015 Apr 8.

[40] *Nanopolish*. Available from: https://github.com/jts/nanopolish

[41] Tomás M, Kai C, Greg C, et al. Efficient estimation of word representations in vector space. ICLR (Workshop Poster). 2013. arXiv preprint arXiv:1301.3781. Available from: https://simpsonlab.github.io/2015/04/08/eventalign/

[42] Ng P. dna2vec- consistent vector representations of variable-length k-mers. doi: 10.48550/arXiv.1701.06279

[43] Milad Mostavi YH. Machine learning and deep learning challenges for building 2′o site prediction. bioRxiv 2020.05.10.087189. doi:10.1101/2020.05.10.087189

[44] Guestrin TCAC. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16); San Francisco, CA; 2016 Aug 13–17.

[45] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. doi: 10.1023/A:1010933404324

[46] Jain A. In complete guide to parameter tuning in XGBoost with codes in Python. 2016 Mar. Available from: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-XGBoost-with-codes-python/

[47] scikit-learn Machine Learning in Python. Available from: https://scikit-learn.org/stable/

[48] Grover P. Gradient boosting from scratch. 2017 Dec 8. Available from: https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d

[49] Qi Y. Random forest for bioinformatics. In ensemble machine learning. US: Springer; 2012. p. 307–323.

[50] Genism topic modelling for humans. Available from: https://radimrehurek.com/gensim/models/word2vec.html

[51] Bradley AE. The use of the area under the Roc curve in the evaluation of machine learning algorithms. Pattern Recognition. 1997;30(7):1145–1159. doi: 10.1016/S0031-3203(96)00142-2

[52] Receiver operating characteristic. Available from: https://en.wikipedia.org/wiki/Receiver_operating_characteristic