Nm-Nano: Predicting 2´-O-Methylation (Nm) Sites in Nanopore RNA Sequencing Data

Doaa Hassan[1,4], Aditya Ariyur [5], Swapna Vidhur Daulatabad[1], Quoseena Mir[1], Sarath Chandra Janga[1,2,3]

1. Department of BioHealth Informatics, School of Informatics and Computing, Indiana University Purdue University, 535 West Michigan Street, Indianapolis, Indiana 46202
2. Department of Medical and Molecular Genetics, Indiana University School of Medicine, Medical Research and Library Building, 975 West Walnut Street, Indianapolis, Indiana, 46202
3. Centre for Computational Biology and Bioinformatics, Indiana University School of Medicine, 5021 Health Information and Translational Sciences (HITS), 410 West 10th Street, Indianapolis, Indiana, 46202
4. Computers and Systems Department, National Telecommunication Institute, Cairo, Egypt.
5. Carmel High School, Carmel, IN

Keywords: RNA modifications, Nm (2´-O-methylation), Nanopore

**Correspondence should be addressed to:*

Sarath Chandra Janga (scjanga@iupui.edu)
Informatics and Communications Technology Complex, IT475H
535 West Michigan Street
Indianapolis, IN 46202
317 278 4147

Abstract

Nm (2´-O-methylation) is one of the most abundant modifications of mRNAs and non-coding RNAs occurring when a methyl group (–CH3) is added to the 2´ hydroxyl (–OH) of the ribose moiety. This modification can appear on any nucleotide (base) regardless of the type of nitrogenous base, because each ribose sugar has a hydroxyl group and so 2´-O-methyl ribose can occur on any base. Nm modification has a great contribution in many biological processes such as the normal functioning of tRNA, the protection of mRNA against degradation by DXO, and the biogenesis and specificity of rRNA. Recently, the single-molecule sequencing techniques for long reads of RNA sequences data offered by Oxford Nanopore technologies have enabled the direct detection of RNA modifications on the molecule that is being sequenced, but to our knowledge there was only one research attempt that applied this technology to predict the stoichiometry of Nm-modified sites in RNA sequence of yeast cells. To this end, in this paper, we extend this research direction by proposing a bio-computational framework, Nm-Nano for predicting Nm sites in Nanopore direct RNA sequencing reads of human cell lines. Nm-Nano framework integrates two supervised machine learning models for predicting Nm sites in Nanopore sequencing data, namely Xgboost and Random Forest (RF). Each model is trained with set of features that are extracted from the raw signal generated by the Oxford Nanopore MinION device, as well as the corresponding basecalled k-mer resulting from inferring the RNA sequence reads from the generated Nanopore signals. The results on two benchmark data sets generated from RNA Nanopore sequencing data of Hela and Hek293 cell lines show a great performance of Nm-Nano. In independent validation testing, Nm-Nano has been able to identify Nm sites with a high accuracy of 93% and 88% using Xgboost and RF models respectively by training each model with Hela benchmark dataset and testing it for identifying Nm sites on Hek293 benchmark dataset. Thus, Nm-Nano outperforms the Nm sites predictors existing in the literature (not relying on Nanopore technology) that were only limited to predict Nm sites on short reads of RNA sequences and unable to predict Nm sites on long RNA sequence reads. By deploying Nm-Nano to predict Nm sites in Hela cell line, it was revealed that a total of 196 genes was identified to have the most abundance of Nm modification among all other genes that have been modified by Nm in this cell line. Similarly, deploying Nm-Nano to predict Nm sites in Hek393 cell line revealed that a total of 196 genes line was identified to have the most abundance of Nm modification among all other genes that have been modified by Nm in this cell line. According to this, a significant enrichment of a wide range of functional processes like high confidences (adjusted p-val < 0.05) enriched ontologies that were more representative of Nm modification role in immune response and cellular homeostasis were revealed in Hela cell line, and "MHC class 1 protein complex", "mitotic spindle assembly", "response to glucocorticoid", and "nucleocytoplasmic transport" were revealed in Hek293 cell line. The source code of Nm-Nano can be freely accessed at https://github.com/Janga-Lab/Nm-Nano .

1. Introduction

2′-O-methylation (or Nm, where N denotes any nucleotide) is a co- or post-transcriptional modification of RNA, occurring when a methyl group (–CH3) is added to the 2′ hydroxyls (–OH) of the ribose moiety. This modification can appear on any nucleotide regardless of the type of nitrogenous base (Base), where 2′-O-methyl ribose can occur at any base. Nm is an abundant modification that occurs frequently in mRNAs and at multiple locations in non-coding RNAs such as transfer RNA (tRNA), ribosomal RNA (rRNA), and small nuclear RNA (snRNA) [1-3]. This is due to the role that internal 2′- O-methylation 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 [4].

Nm modification has a great contribution in many biological processes such as the normal functioning of tRNA [5], protecting mRNA from degradation by DXO [6], and the biogenesis and specificity of rRNA [7,8]. It has been also found that Nm modification has been associated with many human diseases known to date and has potential indirect links to some other biological defects [9].

Detecting Nm modifications in RNAs has been a great challenge for many years and various experimental methods for identifying such modification have been presented in the literature [9]. However, each of these methods has exhibited significant limitations. For example, RiboMethseq was introduced as a high throughput method in which Nm modifications could be mapped based on their protection given against alkaline hydrolysis, resulting in Nm nucleotides being depleted from the start of sequencing reads [11]. However, RiboMethseq couldn't be applied to short RNAs. To address this limitation, two other chemical methods: Nm-seq and RibOxi-seq were presented for detecting Nm mofications in RNAs [12,13]. Using these methods, Nm sites could be mapped after ligation of linkers to the Nm-modified nucleotide at the 3′-end. However these methods were only able to identify significantly fewer Nm modification sites relative to those reported by LC-MS/MS methods, a biochemical method to detect and quantify the relative abundance of RNA modification [14,15]. Despite LC-MS/MS providing industry standard results, it is time and labor consuming, as well as requiring large amounts of input RNA and is limited for low-abundance nucleotides [16].

On the other hand, there have been few computational biology methods presented in the literature as complementary to the experimental methods to address their limitations [17-19]. What computational methods that have been reported mainly rely on developing machine/deep learning classification algorithms to identify Nm sites in RNA sequences based only on short read data. Additionally, the accuracy of predicting Nm sites in some of these methods has not been explicitly presented [18]. For instance, a support vector machine-based method was presented in [17] to identify Nm sites in RNA short reads 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 [18] proposed a deep learning-based method for identifying Nm sites in short reads RNA sequences. In this approach, dna2vec- a biological sequence embedding method originally inspired by the word2vec model of text analysis was adopted to yield embedded representations of RNA sequences that may or may not contain Nm sites. Those embedded representations were fed as features to a Convolutional Neural Network (CNN) for classification of RNA sequences into modified or not modified with Nm sites. The method was trained using the data collected from Nm-seq experimental method. Another prediction model based on Random Forest for identifying Nm sites in short read RNA sequences was presented in [19]. This model was trained with features extracted by multi-encoding scheme combination that combines the one-hot encoding, together with position-specific dinucleotide sequence profile and K-nucleotide frequency encoding.

Recently, the third-generation sequencing technologies such as the platforms provided by Oxford Nanopore Technologies (ONT) has been proposed as a new mean to detect RNA modifications on long RNA sequence data. However this technology has been only used once to predict the stoichiometry of Nm-modified sites in yeast mitochondrial rRNA using a KNN algorithm trained to classify the reads into two classes: modified or unmodified [20]. To this end, our work aims to extend this direction by combining machine learning and Nanopore Technology to identify Nm sites in long RNA sequence reads of human cell lines. 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 of Hela and Hek293 cell lines, namely the Extreme Gradient Boosting (Xgboost) and Random forest (RF) models (Figure 1).

XGboost is trained with a set of features extracted from the raw signal generated by Oxford MinION Nanopore sequencing device when sequencing RNA reads and the corresponding basecalled k-mers resulting from basecalling the generated signals back to the original RNA sequence. The features extracted from the Nanopore signals include: the mean and stanndard deviation of the signal, the mean and standard deviation of the simulated signal that is generated by evenalign module of Nanoplish (a free software for Nanpore signal extraction and analysis [21-23]), and the difference between the mean of the signal and the mean of the simulated one. The features extracted from the basecalled k-mers include a feature that is obtained by checking the matching between the reference k-mer and the model k-mer. The former refers to the basedcalled k-mer resulting from aligning events/signals to a reference genome using eventalign Nanoplish module. The later is a simulated basecalled k-mer that is generated by eventalign module of Nanpolish software. In addition, the genomic location/ position of the Nm modification is used among the extracted features used to train XGBoost model. The genomic location is also obtained as an output when aligning nanopore events/signals to a reference genome using Nanopolish eventalign module.

Similarly RF is trained with the same set of features used to train the XGBoost. In addition to embedding features that are generated by applying the word2vec [24] technique to each of reference k-mers in the extracted Nanopore signals. Using this technique, each reference k-mer resulting from basecalling the corresponding signal is represented by 1-dimensional vector. All of 1-dimentional vectors corresponding to reference k-mers are combined with the aforementioned extracted features from Nanopore signals and used to train the RF model, which in turn will able to predict whether the signal is modified by the presence of Nm sites in the testing phase.

The developed predictors integrated in Nm-Nano framework for identifying Nm sites have been trained and tested upon a set of ‘modified’ sequences containing Nm sites at known positions and ‘unmodified’ ones.

2. Results and discussions

We have used two validation methods when evaluating the performance of Nm-Nano predictors: namely the random-test splitting and the test with independent cell line. In the former, the benchmark dataset 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 ML model and 20% of the dataset is kept for testing. In the latter, two benchmark datasets for two different cell lines were used, one for training and another for testing. For the performance evaluation results of Nm-Nano ML models, Hela benchmark dataset was used for training the models, while Hek293 benchmark dataset was used for testing them.

2.1 Performance evaluation with random-test splitting

Table. 1 shows the performance of Xgboost and RF with embedding ML models implemented in Nm-Nano that are available on its GitHub page when applied to the benchmark dataset of Hela cell line. Xgboost is trained with the 7 extracted features introduced early in Section 1 and later in Subsection 5.3, while RF is trained with those 7 features combined with the features generated with word2vec embedding introduced early in Section 1 and later in Subsection 5.4. As the table shows, Xgboost model outperforms the RF with embedding in terms of accuracy, precision, recall and AUC.

Classifier	Accuracy (%)	Precision	Recall	AUC
XGboost	96.46	0.97	0.96	0.965
RF	91.5	0.93	0.9	0.915

Table 1: The performance of Nm predictors on Hela benchmark dataset with random-test splitting.

The learning (Figure 2. panels A, and D), and loss (Figure 2. panels B, and E) curves of Xgboost, and RF with k-mer embedding show the performance of Xgboost in terms of accuracy score and misclassification error outperforms the performance of RF with embedding. Also, the receiver operating characteristic (ROC) curves (Figure 2. panels C, and F) of Xgboost and RF with k-mer embedding show that the percentage of true positive rate to the false positive rate in case of Xgboost model is more than the one for RF with k-mer embedding model.

2.1.1 Performance results using single type of feature

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 and generated features with word2vec embedding technique introduced early in Section 1 and later in Subsection 5.4. Clearly the features generated by word2vec embedding technique achieve the best contribution to the performance of RF among all other features. However, the contribution of those features to the performance of XGboost model was not presented as they were not considered for training this learning model. This is because the performance of XGboost was high after tunning its parameters with grid search algorithm which takes too much time for obtaining the best parameters of Xgboost. Therefore, generating more features with word2vec embedding techniques for training grid-search Xgboost model will add extra processing overhead due adding the time that is taken by word2vec technique for generating extra features to the time that is taken by the grid search algorithm for hyper parameter tuning of Xgboost ML model.

For the extracted features, the position feature contributes more to the classifiers' accuracy than other extracted features used for training either the Xgboost or RF with embedding ML models. It is followed by the model mean, then model standard deviation features. It was also observed that the k-mer match feature achieves the lowest contribution to the performances of Xgboost and RF with embedding ML models.

Classifier	position	event_level _mean	event_stdv	Model_mean
XGboost	94.71%	58.97%	54.66%	84.21%
RF	76.63%	59.03%	54.82%	74.71%

(a)

Classifier	Mode_stdv	K-mer_match	Mean_diff	embedding
XGboost	66.3%	53.12%	53.57%	-
RF	66.03%	53.12%	53.55%	86.27%

(b)

Table 2: The performance of Nm's predictors on Hela benchmark dataset in terms of accuracy with random test-splitting using single type of feature.

2.2 Performance against independent cell line

Table 3 shows the performance of ML models against independent cell line (i.e., with independent test dataset, where Hela cell line benchmark dataset is used for training Nm-Nano's predictors and Hek293 cell line benchmark dataset is used for testing them) using the seven extracted features. As the results show, Xgboost model outperforms RF with k-mer embedding model. The learning (Figure 3. panels A, and D), and loss (Figure 3. panels B, and E) curves of Xgboost, and RF with k-mer embedding show the performance of Xgboost in terms of accuracy score and misclassification error outperforms the performance of RF with k-mer embedding. The receiver operating characteristic (ROC) curves (Figure 3. panels C, and E) of Xgboost and RF with embedding show that the percentage of true positive rate to the false positive rate in case of Xgboost model is more than the one for RF with embedding model. A supplementary Figure 1 shows the learning, loss and ROC curves of Xgboost (Panel A, B, and C) against the learning, loss and ROC curves of RF with embedding when reversing and training both models with Hek293 RNA sequence reads and testing with Hela RNA sequence reads.

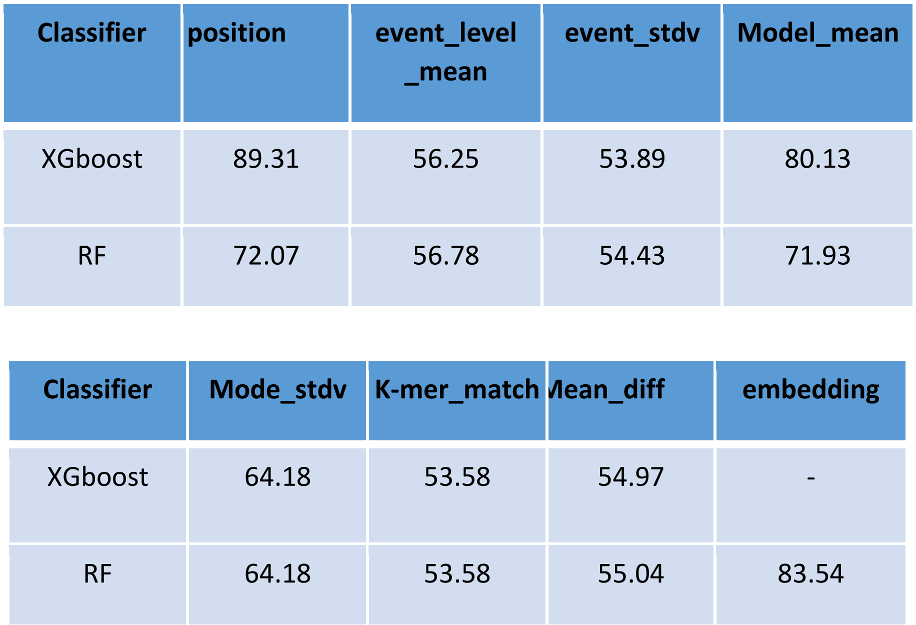
Classifier	accuracy (%)	precision	recall	AUC
XGboost	92.8%	0.96	0.89	0.928
RF	88.8%	0.92	0.85	0.89

Table 3: The performance of Nm's predictors against independent cell line.

2.2.1 Performance results using single type of feature

Table 4 shows the performance of ML models against independent cell line in terms of accuracy with single type of feature among the seven extracted features in combination with features generated with word2vec embedding technique that will be described in subsection 5.5.2. Clearly the features generated with word2vec embedding technique contributes more to the RF classifier accuracy than other features, but 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 more features by wor2vec technique and the time taken by grid search algorithm for obtaining the best parameters of Xgboost as we early mentioned in subsection 2.1.1.

As for the contribution of each of the seven extracted features, it was observed that the position feature achieves the best among all extracted features followed by model mean feature, then the model standard deviation feature. Also, it was observed that k-mer match has the lowest contribution to the performance of either Xgboost or RF with k-mer embedding models.



Classifier	position	event_level_mean	event_stdv	Model_mean
XGboost	89.31	56.25	53.89	80.13
RF	72.07	56.78	54.43	71.93

Classifier	Mode_stdv	K-mer_match	Mean_diff	embedding
XGboost	64.18	53.58	54.97	-
RF	64.18	53.58	55.04	83.54

Table 4: The performance of Nm's predictors against independent cell line in terms of accuracy using single type of feature.

2.3 Abundance of Nm sites

In order to identify the abundance of Nm sites in the RNA sequence of either Hela or Hek293 cell lines, first we run the best machine learning model in Nm-Nano framework (i.e., XGBoost) on the complete RNA sequence reads of Hela and Hek293 cell lines. Then, we identify all samples with predicted Nm sites in those reads, then we identify the number of Nm unique genomic locations as well as their frequencies in the two complete cell lines. We found that there are 27,068,157 Nanopore signal samples predicted as samples with Nm sites from a total of 920,643,074 Nanopore signal samples that represent the complete Hela cell line with 4,064,938 unique genomic locations of Nm (Supplementary excel file 1). Similarly, we found that there are 10,541,009 Nanopore signal samples predicted as samples with Nm sites from a total of 275,056,669 samples that present the complete RNA sequence of Hek293 cell line with 2,952,972 unique genomic locations of Nm modification (Supplementary excel file 2). As for overlapping between unique genomic locations of Nm in both cell lines, we found that there are 1,191,677 genomic locations common between Hela and Hek293 cell lines (Figure 4.A). Also, we found that there is an overlapping of 76 genes between the top 1% frequent modified Nm genes of both complete Hela and Hek293 cell lines (Figures 4.B). Clearly, we notice that the extent of Nm modification (the number of Nanopore signal samples predicted as samples with Nm sites to the total number of Nanopore signal samples in the complete RNA sequence of the cell line) in RNA sequences of Hela cell line is slightly less than its counterpart for Hek293 cell line (2.94 % for Hela versus 3.83% for Hek293). Therefore the distribution of Nm across normalized gene length for Hela cell line is slightly less than its equivalent in Hek293 cell line (Figures 4.C).

Since Nm modifications can occur at any RNA base, we have also reported about the percentage of unique Nm locations occurring per each RNA base in the two complete cell lines of Hela and Hek293 (Table 5.)

Cell line	A base	C base	G base	U base
Hela	27.86%	22.29%	22.41%	27.44%
Hek293	27.63%	22.49%	22.63%	27.24%

Table 5: The percentage of unique Nm locations occurring per each base of RNA sequence in Hela and Hek293 cell lines.

2.4 Functional enrichment analysis

A total of 176 genes from Hek293 and 196 genes from Hela cell lines were identified to have the most abundance of Nm modification. The short-listed genes from both cell lines were plugged into Cytsoscape ClueGo [25] application to obtain enriched ontologies and pathways at high confidence (p<0.05). Enrichment observations from this analysis are visualized in Figure 5 A, and B for Hek293 and Hela cell lines respectively.

From the functional enrichment analysis of the gene set form Hek293 cell line (Figure 5A), we observed a wide range of functional processes like" "MHC class 1 protein complex", "mitotic spindle assembly", "response to glucocorticoid", and "nucleocytoplasmic transport" being significantly enriched. Essentially highlighting the diverse regulatory role of an Nm modification, from its involvement in cell immune signaling to cellular processing.

In Hela cell line, we observed several high confidences (adjusted p-val < 0.05) enriched ontologies that were more representative of Nm modification role in immune response and cellular homeostasis (Figure 5B) like: "Regulation involved in apoptotic pathway", "antigen processing and presenting", and "ER to Golgi transport mechanism".

To observe which cellular pathways were associated with the Nm modifications, we ranked the complete human gene lists from both Hela and Hek293 cell lines based on occurrence of Nm modification locations and performed GSEA gene set enrichment analysis [26]. Across both cell lines we observed that genes associated with immune pathways were enriched in these ranked lists, reinforcing the association between Nm Modification and immune response which was previously observed in literature [27, 28, 29]. Both cell lines had pathways like: "KEGG_NATURAL_KILLER_CELL_MEDIATED_CYTOTOXICITY", "KEGG_ANTIGEN_PROCESSING_AND_PRESENTATION", and

"KEGG_AUTOIMMUNE_THYROID_DISEASE" with high enrichment scores (NES>1.5) as seen in supplementary Figure 2. Apart from immune associated pathways being enriched, we also observed some tissue specific pathways were enriched in case of Hela, like cardiovascular pathways such as "KEGG_HYPERTROPHIC_CARDIOMYOPATHY_HCM", and "KEGG_CARDIAC_MUSCLE_CONTRACTION". Those pathways were enriched with high normalized enrichment scores (NES>1.3).

3. Implementation and usage of Nm-Nano

The main file of Nm-Nano framework is implemented in python 3.x and the file has to be run on Linux environment by running the following command from Nm-Nano main directory on the user's local machine after cloning the code from Nm-Nano GitHub repository:

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

Where the following two inputs files are needed to run the main file:

- The absolute path to the reference Genome file (ref.fa)
- The absolute path to fastq reads file (reads.fastq)

Once the user runs the Nm-Nano framework by executing the main python file, then the framework pipeline that accepts the two inputs mentioned above will start execution and the user will be asked to enter the bed file name with the absolute path and extension that is needed to generate the coordinate file that is needed for labeling the Nanopore signals samples as Nm modified and unmodified ones. Next, the framework will extract the raw Nanopore signals from the input fast5 file(s) as well as extracting some of its corresponding features that are used later to train Xgboost and Rf with embedding ML models integrated in Nm-Nano framework for predicting Nm sites in direct Nanopore RNA sequence (Figure 1). It should be also mentioned that Nm-nano framework can also be extended by integrating other machine learning and deep learning models for predicting Nm sites.

4. Discussion and Conclusions

In this paper, we have proposed a new framework called Nm-Nano that integrates two machine learning models: the Xgboost and RF with k-mer embedding. It has been shown that the proposed framework was efficient in detecting Nm sites in RNA long reads which addresses the limitations of most existing Nm predictors presented in the literature that were able only to detect Nm sites in short reads of RNA sequences. It was also observed that deploying Nm-Nano on the total direct RNA Nanopore sequence of Hela and Hek293 lead to obtaining some biological results from preforming functional enrichment analysis for the total number of discovered frequently modified Nm genes in both cell lines. These results can be observed by a wide range of functional processes in Hela and Hek293 cell lines. In Hela, we observed several high confidences (adjusted p-val < 0.05) enriched ontologies that were more representative of Nm modification role in immune response and cellular homeostasis, while In Hek293 we observed a wide range of functional processes that highlight the diverse regulatory role of Nm modification, from its involvement in cell immune signaling to cellular processing. For this reasons, Nm-Nano would be a useful tool for accurate identification of Nm sites in RNA read sequence.

5. Materials and Methods

5.1 Basic approach pipeline

The complete pipeline of Nm-Nano framework for identifying Nm modifications in RNA sequence consists of several stages. The first part of the pipeline starts by culturing the cell line by extracting it from an animal and let it grows in an artificial environment. Next, the RNA is extracted from this cell during library preparation and put through the MinION device and start generating Nanopore signal data. After that, the fast5 file that stores the raw electrical signal levels that are output by the Nanopore sequencers is produced by ONT and basecalledd via Guppy basecaller [30] to determine which base is being passed through the ONT and then aligned to a reference genome to produce the SAM file using minimap2 tool [31]. From the SAM file, a BAM and sorted BAM file are generated using samtools[32], where the BAM file is a compressed version of the SAM file. Next, a coordinate file with ids of fast5 files that have the target modification is created using the produced SAM file and a provided BED file that highlights the target modified locations on the whole genome [33]. This coordinate file is needed for labeling the signal samples produced by eventalign module as modifed and unmodified when training any of Nm-Nano predictors. Next, eventalign module of the Nanopolish software that performs signal extraction is launched, which produces a dataset of Nanopore signal samples. Therefore, the structure of Nm-Nano's pipeline emphasizises that it has some common parts with the pipeline of Penguin [34], our early developed tool for detecting Pseudouridine sites in long reads RNA sequence. However, our Nm-Nano's pipeline is different from Penguin's pipeline in three phases (Figure 1): the benchmark dataset generation, the feature extraction and ML models construction phases. The benchmark dataset generaion phase in Nm-Nano's pipeline is different from its equivalent in Penguin's pipeline because Nm modifications can occur at any RNA base, and so all the samples that are generated from signal extraction are used to identify Nm sites. However, some of those samples are modified with Nm sites, while the remaining are control samples that are not modified with Nm modification and using the information in the coordinate file some of those samples will be labeled as modified, while the remaining will be labeled as unmodified. Simililarly the feature exraction phase in Nm-Nano's pipeline is different from its equivalent in Penguin's pipleine because the features extracted from the modified and unmodified signal samples to train the constructed ML models for predicting Nm sites are different from those that were extracted to train Penguin's predictors. Finally, the ML models construction phase in Nm-Nano's pipeline is different from its equivalent in Penguin's pipeline because it deploys machine learning models (the XGBoost and Random Forest wih k-mer embedding) for predicting Nm sites in long RNA sequence reads that are different from the set of developed ML predictors integrated in the Penguin tool. In the next subsection we will highlight those differences by introducing more details about the benchmark dataset generation, feature extraction and ML model constructions.

5.2 Benchmark datasets generation

Two different benchmark datasets were generated for Hek293 and Hela cell lines (Supplementary csv files Nm_hek.csv and Nm_Hela.csv). Both datasets were generated by considering all the samples output by Nanopolish eventalign module that was run on the basecalled RNA sequence reads of each cell line. In order to label each sample, all the samples generated from signal extraction were used as the the target samples for identifying Nm modification since Nm modifications can occur at any RNA base. Next, the intersection between their position column on the reference genome and the position in the coordinate file (generated from Nm BED file and SAM file for each cell line) is determined. This intersection will represent the positive samples, while the remaining samples will be the negative samples. In the end we have 56,320 samples: 28,160 are positive and 28,160 are negative ones (after sampling the negative samples which are very huge in comparison with negative ones) for Hek293. Similarly, we get 192,082 samples: 96,041 are positive samples and 96,041 are negative samples for Hela cell line.

5.3 Feature extraction

Each generated benchmark dataset has seven columns that represent seven features that were used for training the machine learning models that we developed and integrated in Nm-Nano framework. Those features are: position, event_level_mean, event_stdv, model_mean, model_stdv, mean_diff, and reference & model k-mer match. The first five features were directly extracted by picking their columns from the eventalign's output (Supplementary text2) (namely: position, event_level_mean, event_stdv, model_mean , and model_stdv columns). 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 by eventalign module (model_mean). The seventh feature is generated by checking if the reference_k-mer and model_k-mer coulmns in the eventalign's output match each other, where the former refers the basecalled k-mers resulting from inferring the RNA sequence reads from extracted Nanopore signals by evenalign in the basecalling phase, while the latter refers to bacalled k-mers resulting from inferring RNA sequence reads from simulated signals by eventalign. The value of reference & model k-mer match is 1 if reference and model k-mers match each other and 0 otherwise.

5.4 Features generation with word embedding

In addition to the extracted features, embedding features have been generated using the word2vec technique that are combined with other extracted features that have been previously mentioned for training the RF classifer model that has been developed for predicting Nm sites in long RNA sequence reads. Those embedding features are obtained by applying the word2vec technique to the reference k-mer, where each reference k-mer is represented by 1-dimensional vector of fixed size (the vector size is set optionally as a parameter when building word2vec embedding model). In summary, the combination of all extracted features and embedding features are used to train the RF model, which in turn will able to predict whether the signal is modified by the presence of Nm sites in the testing phase.

5.5 ML Models construction

We have developed two machine learning models for predicting Nm sites in RNA sequence reads including the XGBoost [35] and RF [36] with k-mer embedding. The XGBoost model parameters were tuned using the Grid-search hyperparmerter tuining algorithm [37]. For RF, the seed number parmeter was set to 1234 and the number of trees paramter was set to 30 for obtaining the best performance of RF. The optimized distributed gradient boosting python library has been used for implementing the XGBoost model [38] and the scikit-learn toolkit [39], the free machine learning python library has been used for implementing the RF model.

5.5.1 XGBoost with grid search for hyper parameter tuning

The Extreme Gradient Boosted trees (XGBoost) is a special implementation of Gradient Boosting [40]. Gradient boosting is a machine learning technique that produces a prediction model based on an ensemble of weak prediction models, which are decision trees in the case of XGBoost. This model is highly flexible and versatile and can be applied for classification-based problems, which is the main goal of this study. The advantage that XGBoost has over other tree-based models is that it has a faster training time along with its regularized boosting, which helps to prevent overfitting: this is when the machine learning model learns and becomes too accustomed to the training data and is not able to generalize and accurately predict the testing data. XGBoost does not also require feature scaling due to being a tree-based model, which is a major advantage. Feature Scaling is required for many non-tree-based models such as Support Vector Classifiers (SVM) and Logistic Regression (LR). Although feature Scaling is beneficial for these models, it causes certain feature importance and interpretability to be reduced, which may lead to lower accuracies. XGBoost can also cross-validate each iteration (round) of its training process, which can lead to higher results than models that cannot do the latter process. The use of decision trees and gradient boosting also provided the advantages of both random forest and other gradient boosting models, causing XGBoost to typically have a prediction error many times lower than regular gradient boosting or random forest.

The XGBoost machine learning model was created after the data was preprocessed by removing null values and performing feature extraction, The model has several parameters that can be adjusted and tuned to get the best performance of XGBoost. Hyper-parameter tuning using the grid search algorithm has been used 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_depths, min_child_weights, and scale_pos_weight. The optimized values for these parameters obtained using grid search algorithm were 0.01, 0.1, 15, 3, and 1 respectively. The parameter eta, representing the learning rate of the XGBoost model. Gamma parameter represents how conservative the model is. The parameter max_depth represents how deep a decision tree can be built and min_child_weight represents the minimum value needed to activate the respective node in the decision tree. The scale_pos_weight parameter controls the balance of positive and negative weights; this parameter is associated with the min_child weight. After the values for these best parameters were obtained by fitting the grid search XGBoost model to the training data, they were applied to the model to obtain its prediction results in the testing phase.

5.5.2 RF with k-mer embedding

We have developed a Random Forest (RF) ML model that has been trained with the extracted features and the features generated by applying Word2vec embedding technique to the reference k-mer, one feature column in the benchmark Nm modification dataset of Hela and Hek293 cell lines. RF algorithm has been extensively used in the literature to address several problems in bioinformatics research [41]. It has been observed that the performance of RF model is improved when it is trained by combination of the extracted features and the generated k-mer embedding features and outperforms its perfromance when it is trained with the extracted features only as we mentioned early in subsections 2.1.1 and 2.2.1.

The idea of applying Word2vec to reference k-mer has been inspired by the work in [42] in which word2vec has been 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 approach to vector encoding of k-mer using one-hot technique that is subject to the curse of dimnensionality problem..

The RF machine learning model was created after the data was preprocessed by removing null values then performing feature extraction and combing them with generated k-mer embedding features. The k-mer embedding feature was generated using genism [43], a free python library that implements word2vec algorithm using highly optimized C routines, data streaming, and pythonic interfaces. The word2vec algorithm has various parameters including: the vector size, the window size, and and the word count. The vector size is the dimensionaility of the vector that repesents 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-mers. The word count refers to the minimum count of words to consider when training the model, where words with occurrence less than this count will be ignored. The k-mer embedding features that lead to best performance of RF have been generated by setting the vector size to 20, the minimum word count to 1, and the window size to 3.

5.6 Performance evaluation metrics

The accuracy (Acc), precision (P), recall (R), and the area under ROC curve (AUC) [44] have been used as metrics for evaluating performance of Nm-Nano predictors. The mathematical notions for the first three metrics are identified 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 classifed Nm sites.
- FP denotes false positive and refers to 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 classifed non-Nm sites.

As for AUC metric, it measures the entire two-dimensional area under the ROC curve [45] which measures how accurately the model can distinguish between two things (e.g. determine if a base of RNA sequence is Nm site or not).

5.7 Environmental settings

Nm-Nano has been developed as tool for detecting Nm modification in Nanopore RNA sequence data by integrating two machine learning models: the Xgboost and RF with k-mer embedding to identify this type of RNA modification. XGboost parameters were tuned to get the best performance using the Grid search algorithm which takes around 6 hours 52 min to fit on the training dataset of Hek293 and 9 hours 12 min to fit on the training dataset of Hela cell line for obtaining the best parameters that were applied to XGboost model in the testing phase. The experiment was executed on windows machine with (8 cores) processor of Ryzen 5900HS CPU, and 16 GB RAM. It should be observed that the time taken by grid search algorithm for obtaining the best parameters of XGboost is added to the total time that is needed to apply XGboost on the benchmark dataset of a given cell line for detecting Nm modification. However, using grid search algorithm for hyper parameter tuning of XGboost causes a significant improvement in the performance of XGboost. Similarly, when developing RF with k-mer embedding ML model to identify Nm modification in Nanopore sequence data of a specific cell line, the word2vec embedding algorithm should be applied first to reference k-mers in the generated benchmark dataset before applying RF algorithm to that dataset for generating embedding features that would be added to the extracted features to train RF Model. This will add extra time to the execution time of RF algorithm. However, applying embedding with word2vec for generating embedding features added to the extracted ones achieves a significant improvement in the performance of RF. Meanwhile, we thought about improving the performance of XGboost by applying grid search algorithm for hyper parameter tuning in addition to applying k-mer embedding with word2vec for generating embedding features that would be added for the extracted features used for training XGboost. However, we found that this will make XGboost slow when applying it to the benchmark dataset of a given cell line with a slight improvement in its performance that would not be proportional to the huge increase in the processing time of Xgboost.

It should be also observed that Xgboost outperforms RF with embedding when applied on the whole dataset of Hek293 or Hela either in test-split or in the independent test. However, the performance of RF with k-mer embedding might outperform grid search XGboost model if the ML models are applied to a part of the benchmark dataset of the cell line or when using other types of cell lines different from Hek293 and Hela.

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 machine learning predictors namely XGBoost and RF with embedding respectively. DH extracted the benchmark datasets. AA evaluated the performance of XGBoost Nm predictor with the random test split and against independent cell line. DH evaluated the performance of RF with embedding Nm predictor with the random test split and against independent cell line. DH and SVD performed results and functional enrichment analysis. QM performed the cell culturing, RNA library preparation and Nanopore RNA sequence for Hela and Hek293.

Conflict of interest

The authors report no financial or other conflict of interest relevant to the subject of this article.

Acknowledgement

We thank Alexander Krohannon and Alexandre Plastow at IUPUI for giving valuable comments on this work. 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).

Figure Legends

Figure1. Nm-Nano framework consists of three main phases for predicting Nm sites when testing with an independent cell line showing: the benchmark dataset generation, the feature exraction, and ML models construction and testing phases.

Figure 2. The learning, loss and ROC curves of learning models with random split testing for XGBoost model and RF with k-mer embedding ML models.

Figure 3. The learning, loss and ROC curves of learning models against independent cell line for **XGBoost** model and RF with k-mer embedding ML models

Figure 4. Showing (a) The overlap between Nm unique locations in complete Hek293 and Hela cell lines (b) the overlapping between top frequent 1 % modified Nm genes in complete Hek293 and Hela cell lines (c) The density plots that represents Nm modifications across normalized gene length for Hek293 and Hela cell lines.

Figure 5. Functional enrichement analysis of most frequenlty Nm modified genes across a cell line in terms of functional grouping of the GO-terms based on GO hierarchy using Cytoscape ClueGO application, and a pie chart. (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 representative of the significance of association with respect to genes per GO-term).

References:

1. Darzacq, X.; Jády, B.E.; Verheggen, C.; Kiss, A.M.; Bertrand, E.; Kiss, T. Cajal body-specific small nuclear RNAs: A novel class of 20-O-methylation and pseudouridylation guide RNAs. EMBOJ. 2002, 21, 2746–2756. [CrossRef] [PubMed]
2. Rebane, A.; Roomere, H.; Metspalu, A. Locations of several novel 2´-O-methylated nucleotides in human 28S rRNA. BMC Mol. Biol. 2002, 3, 1. [CrossRef]
3. Somme,J.;VanLaer,B.;Roovers,M.;Steyaert,J.;Versées,W.;Droogmans,L.Characterization oftwohomologous methyltransferases showing different specificities for their tRNA substrates.RNA2014, 20,1257–1271. [CrossRef] [PubMed]

4. Elliott BA, Ho HT, Ranganathan SV, Vangaveti S, Ilkayeva O, Abou Assi H, Choi AK, Agris PF, Holley CL. 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.
5. 7. Guy MP, Shaw M, Weiner CL, Hobson L, Stark Z, Rose K, Kalscheuer VM, Gecz J, Phizicky EM. 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):1176–87.
6. Picard-Jean F, Brand C, Tremblay-Letourneau M, Allaire A, Beaudoin MC, Boudreault S, Duval C, Rainville-Sirois J, Robert F, Pelletier J, et al. 2′-Omethylation of the mRNA cap protects RNAs from decapping and degradation by DXO. PLoS One. 2018;13(3):e0193804.
7. Hengesbach M, Schwalbe H. Structural basis for regulation of ribosomal RNA 2′-o-methylation. Angew Chem Int Ed Engl. 2014;53(7):1742–4.
8. Erales J, Marchand V, Panthu B, Gillot S, Belin S, Ghayad SE, Garcia M, Laforets F, Marcel V, Baudin-Baillieu A, et al. Evidence for rRNA 2′-Omethylation plasticity: control of intrinsic translational capabilities of human ribosomes. Proc Natl Acad Sci U S A. 2017;114(49):12934–9.
9. Dilyana G. Dimitrova, Laure Teysset and Clément Carré.RNA 2´-O-Methylation (Nm) Modification in Human Diseases. Genes 2019, 10, 117; doi:10.3390/genes10020117.
10. Detection and Analysis of RNA Ribose 20-O-Methylations: Challenges and Solutions.
11. Marchand, V. et al. (2016) Illumina-based RiboMethSeq approach for mapping of 2'-O-Me residues in RNA. Nucleic Acids Res. 44, e135
12. Dai, Q. et al. (2017) Nm-seq maps 2'-O-methylation sites in human mRNA with base precision. Nat. Methods 14, 695–698.
13. Zhu, Y. et al. (2017) High-throughput and site-specific identification of 2'-O-methylation sites using ribose oxidation sequencing (RibOxi-seq). RNA 23, 1303–1314.
14. Yuan, B.-F. (2017) Liquid chromatography–mass spectrometry for analysis of RNA adenosine methylation. In RNA Methylation: Methods and Protocols (Lusser, A., ed.), pp. 33–42, Springer New York.
15. Jora, M. et al. (2019) Detection of ribonucleoside modifications by liquid chromatography coupled with mass spectrometry. Biochim. Biophys. Acta Gene Regul. Mech. 1862, 280–290.
16. Anreiter I, Mir Q, Simpson JT, Janga SC, Soller M. New Twists in Detecting mRNA Modification Dynamics. Trends Biotechnol. 2020 Jul 1;S0167-7799(20)30166-9.doi: 10.1016/j.tibtech.2020.06.002.
17. Chen, W., et al., Identifying 2′-O-methylationation sites by integrating nucleotide chemical properties and nucleotide compositions. Genomics, 2016. 107(6): p. 255-258
18. Milad Mostavi, Sirajul Salekin and Yufei Huang. 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, July 2018.
19. 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 Bioinformatics 20, 690 (2019).

20. Begik O, Lucas MC, Pryszcz LP, Ramirez JM, Medina R, Milenkovic I, Cruciani S, Liu H, Vieira HGS, Sas-Chen A, Mattick JS, Schwartz S, Novoa EM. 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.
21. https://github.com/jts/nanopolish
22. Quickstart - how to align events to a reference genome. Available at https://nanopolish.readthedocs.io/en/latest/quickstart_eventalign.html
23. https://nanopolish.readthedocs.io/en/latest/manual.html
24. T. Mikolov, K. Chen, G. Corrado, and J. Dean . **Efficient estimation of word representations in vector space**. (2013a). arXiv preprint arXiv:1301.3781.
25. Bindea G, Mlecnik B, Hackl H, Charoentong P, Tosolini M, Kirilovsky A, Fridman WH, Pagès F, Trajanoski Z, Galon J. ClueGO: a Cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. Bioinformatics. 2009 Apr 15;25(8):1091-3. doi: 10.1093/bioinformatics/btp101.
26. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A. 2005 Oct 25;102(43):15545-50. doi: 10.1073/pnas.0506580102
27. Paramasivam A. RNA 2'-O-methylation modification and its implication in COVID-19 immunity. Cell Death Discov. 2020 Nov 8;6(1):118. doi: 10.1038/s41420-020-00358-z.
28. Dimitrova DG, Teysset L, Carré C. RNA 2'-O-Methylation (Nm) Modification in Human Diseases. Genes (Basel). 2019 Feb 5;10(2):117. doi: 10.3390/genes10020117.
29. Freund I, Eigenbrod T, Helm M, Dalpke AH. RNA Modifications Modulate Activation of Innate Toll-Like Receptors. Genes (Basel). 2019 Jan 29;10(2):92. doi: 10.3390/genes10020092.
30. https://github.com/metagenomics/denbi-nanopore-training/blob/master/docs/basecalling/basecalling.rst
31. H Li. **Minimap2: pairwise alignment for nucleotide sequences**. *Bioinformatics*, **34**:3094-3100, 2018. doi:10.1093/bioinformatics/bty191
32. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, Whitwham A, Keane T, McCarthy SA, Davies RM, Li H, **Twelve years of SAMtools and BCFtools**, *GigaScience* (2021) 10(2) giab008 [33590861]
33. http://genome.ucsc.edu/FAQ/FAQformat#format1
34. Doaa Hassan, Daniel Acevedo, Swapna Vidhur Daulatabad, Quoseena Mir, Sarath Chandra Janga. “Penguin: A Tool for Predicting Pseudouridine Sites in Direct RNA Nanopore Sequencing Data”. bioRxiv; doi: https://doi.org/10.1101/2021.03.31.437901, 2021.
35. Tianqi Chen and Carlos Guestrin. **XGBoost: A Scalable Tree Boosting System.** In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16), August 13-17, 2016, San Francisco, CA, USA.
36. Breiman L. **Random forests**. *Machine learning*. 45:5-32, 2001.
37. B. H Shekar and Guesh Dagnew. **Grid Search-Based Hyperparameter Tuning and Classification of Microarray Cancer Data**. In Proceedings of Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), 2019.

38. Aarshay Jain. **Complete Guide to Parameter Tuning in XGBoost with codes in Python**. March 2016. Avialble at: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/
39. https://scikit-learn.org/stable/
40. https://en.wikipedia.org/wiki/Gradient_boosting
41. Qi Y (2012). **Random Forest for Bioinformatics**. *In Ensemble Machine Learning*, pp. 307-323, Springer, 2012.
42. Patrick Ng. **dna2vec- Consistent vector representations of variable-length k-mers**, Published in Biology Journal on 2017, available at:https://arxiv.org/pdf/1701.06279.pdf
43. https://radimrehurek.com/gensim/models/word2vec.html
44. Andrew E Bradley. The Use of the Area under the Roc Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, Vol. 30, No. 7, pp. 1145-1159, 1997.
45. https://en.wikipedia.org/wiki/Receiver_operating_characteristic

HIGHLIGHTS

- Nm-Nano integrates two ML learning models (i.e., predictors) namely Xgboost and RF with embedding to identify Nm sites in Nanopore direct RNA sequencing reads.
- The pipeline of Nm-Nano framework automates the data preprocessing including Nanopore direct RNA reads alignment using Minimap2, and Nanopore signal extraction using Nanopolish, feature extraction from raw Nanopore signal for training Xgboost and RF with embedding Models implemented in this platform, features generation with word2vec technique needed for training RF with embedding model, and the prediction of Nm sites using any of both models.
- Nm-Nano can predict Nm sites with a high performance on long RNA reads and it outperforms the performance of the state-of-the-art research methods existing in the literature that predict Nm sites only on short RNA reads.
- There are 10541009 Nanopore signal samples predicted by Nm-Nano best ML model (Xgboost) as Nm sites from a total of 275056669 Nanopore signal samples that represent complete RNA sequence of Hek293 cell line with 2952972 unique genomic location of Nm sites.
- There are 27068157 Nanopore signal samples predicted by Nm-Nano best ML model (Xgboost) as Nm sites from a total of 920643074 Nanopore signal samples that represent complete RNA sequence of Hela cell line with 4064938 unique genomic location of Nm sites.
- There is a small fraction of 20% (1191677 unique genomic locations) of Nm sites that are common (overlapped) between both Hek293 and Hela cell lines.
- The extend of Nm modification (the number of Nanopore signal samples predicted as Nm signal samples to the total number of Nanopore signal samples generated from the complete RNA sequence of the cell line) in RNA sequence of Hela cell line is slightly less than its counterpart for HeK293 cell line (2.94 % for Hela cell line versus 3.83% for Hek293)

Figure 1

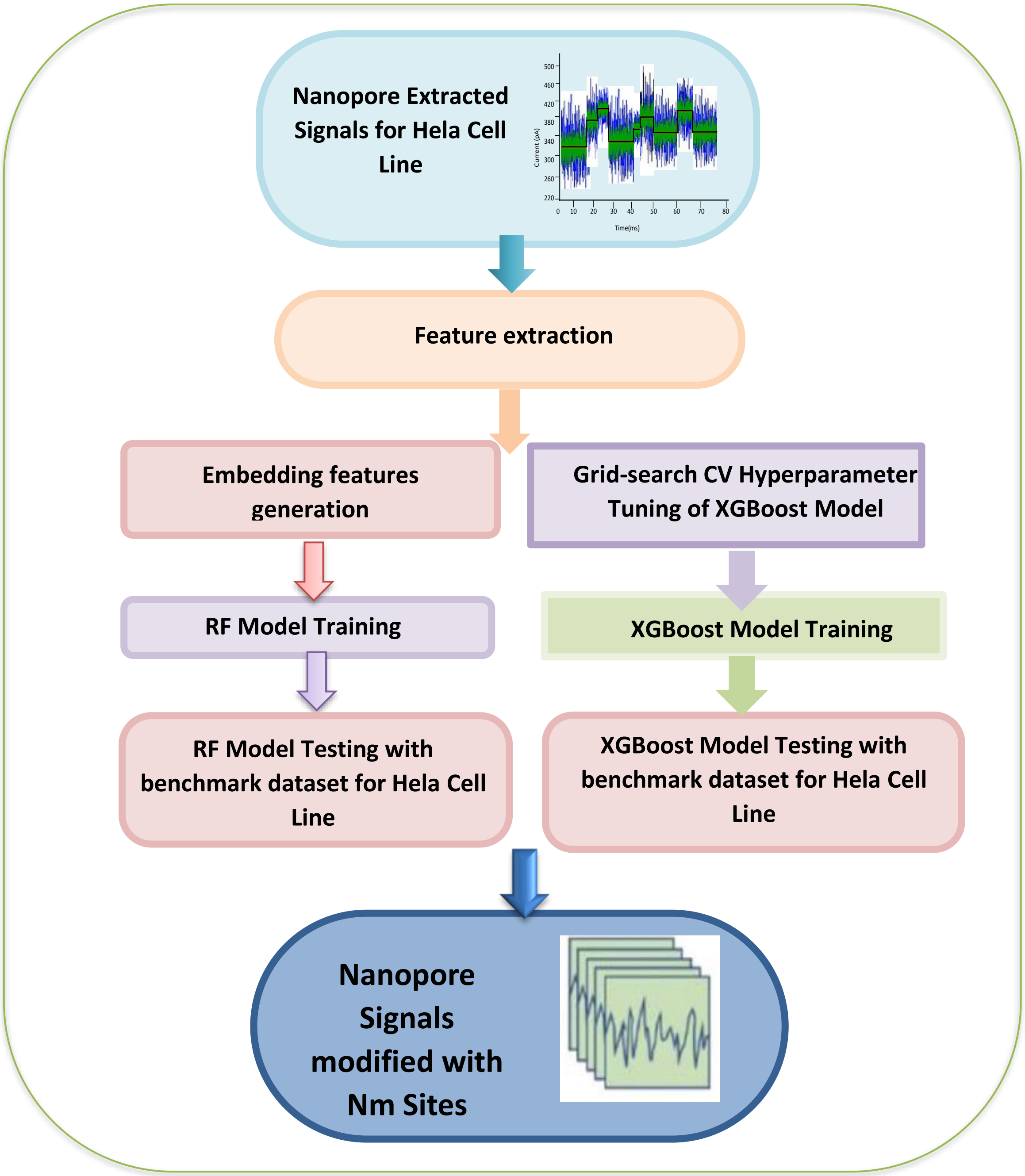

Nanopore Extracted Signals for Hela Cell Line
500
460
420
380
340
300
260
220
Current (pA)
0
10
20
30
40
50
60
70
80
Time(ms)
Feature extraction
Embedding features generation
Grid-search CV Hyperparameter Tuning of XGBoost Model
RF Model Training
XGBoost Model Training
RF Model Testing with benchmark dataset for Hela Cell Line
XGBoost Model Testing with benchmark dataset for Hela Cell Line
Nanopore Signals modified with Nm Sites

Figure 2

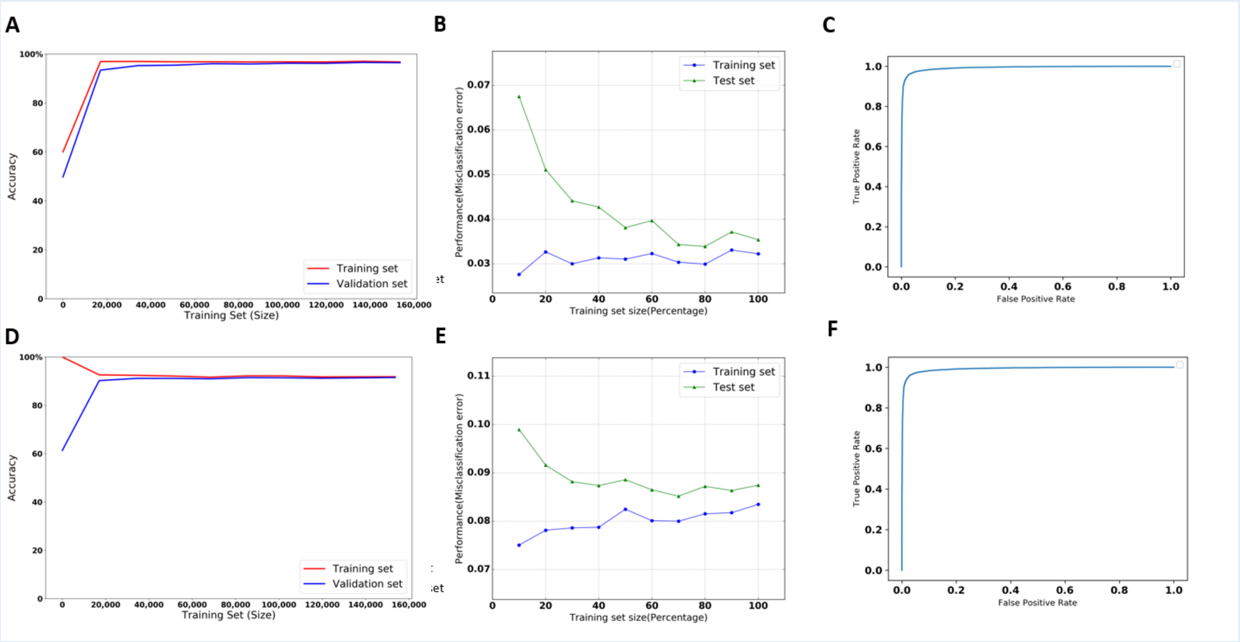

A
Accuracy
100%
80
60
40
20
0
0
20,000
40,000
60,000
80,000
100,000
120,000
140,000
160,000
Training Set (Size)
Training set
Validation set
et
B
Performance(Misclassification error)
0.07
0.06
0.05
0.04
0.03
0
20
40
60
80
100
Training set size(Percentage)
Training set
Test set
C
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
False Positive Rate
D
Accuracy
100%
80
60
40
20
0
0
20,000
40,000
60,000
80,000
100,000
120,000
140,000
160,000
Training Set (Size)
Training set
Validation set
set
E
Performance(Misclassification error)
0.11
0.10
0.09
0.08
0.07
0
20
40
60
80
100
Training set size(Percentage)
Training set
Test set
F
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
False Positive Rate

Figure 3

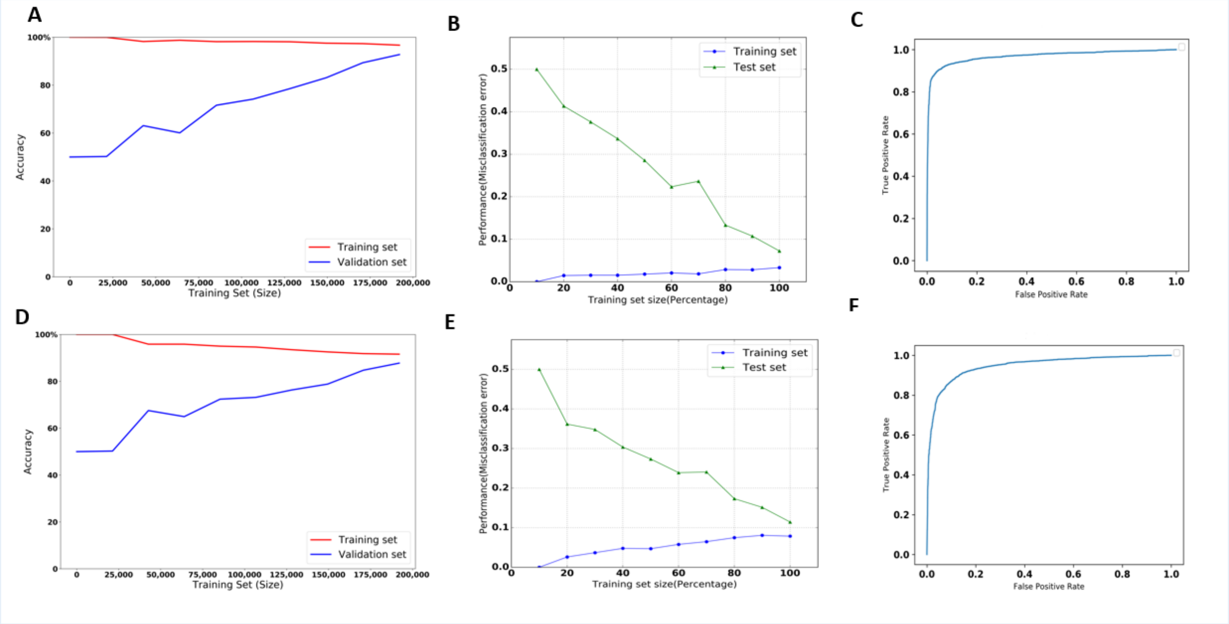

A
100%
80
60
40
20
0
Accuracy
0
25,000
50,000
75,000
100,000
125,000
150,000
175,000
200,000
Training Set (Size)
Training set
Validation set
B
Performance(Misclassification error)
0.0
0.1
0.2
0.3
0.4
0.5
0
20
40
60
80
100
Training set size(Percentage)
Training set
Test set
C
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
D
100%
80
60
40
20
0
Accuracy
0
25,000
50,000
75,000
100,000
125,000
150,000
175,000
200,000
Training Set (Size)
Training set
Validation set
E
Performance(Misclassification error)
0.0
0.1
0.2
0.3
0.4
0.5
0
20
40
60
80
100
Training set size(Percentage)
Training set
Test set
F
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate

Figure 4

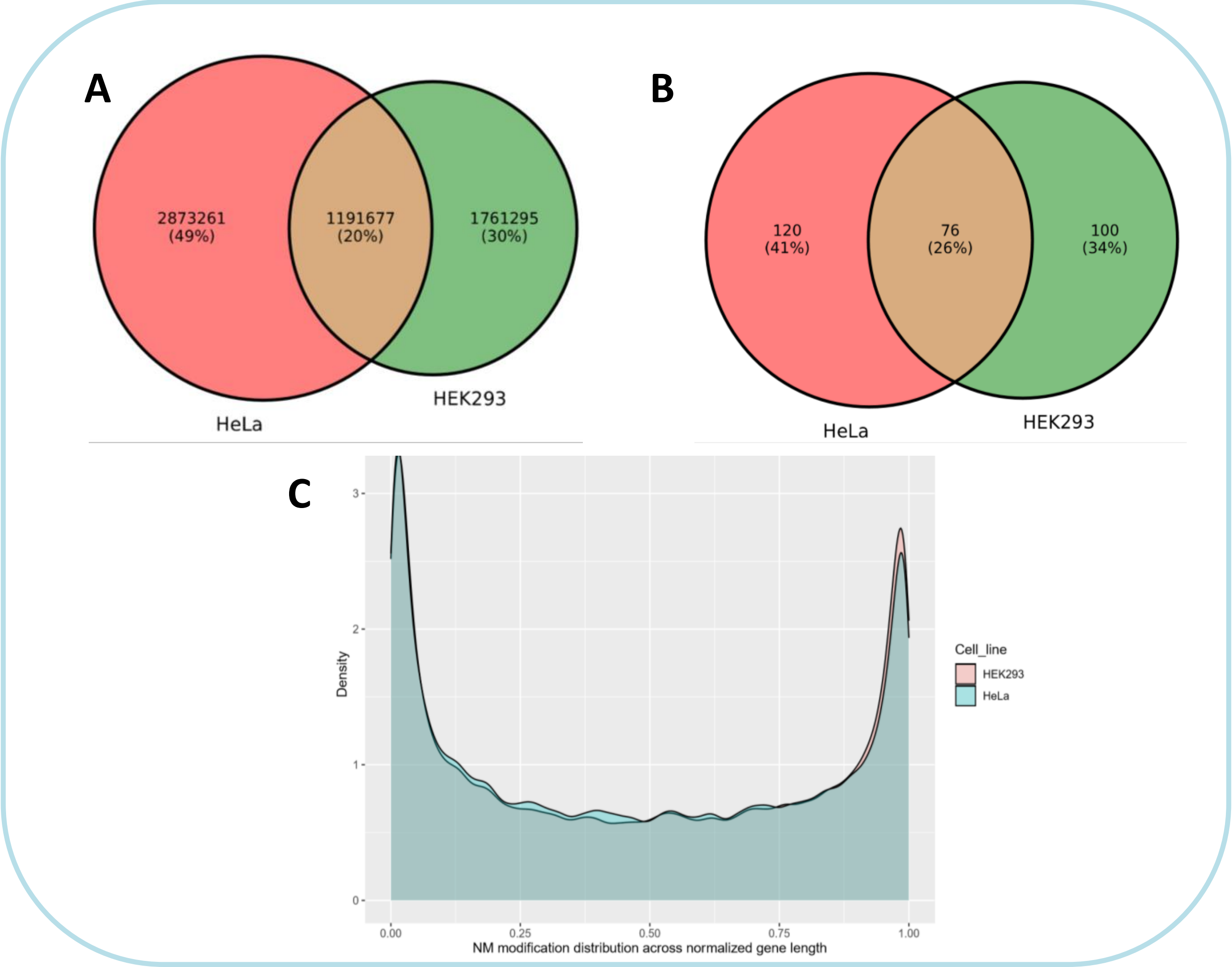
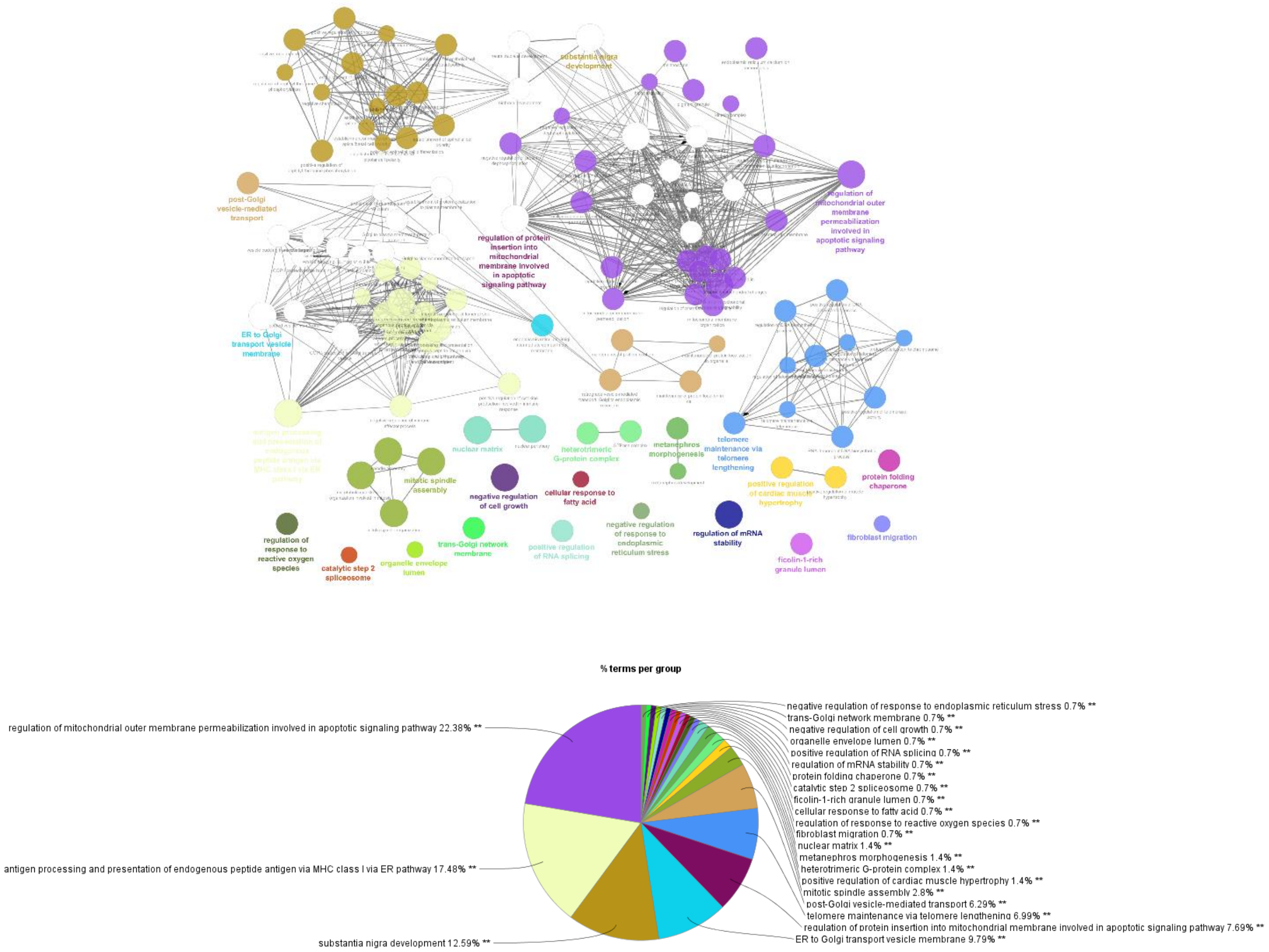
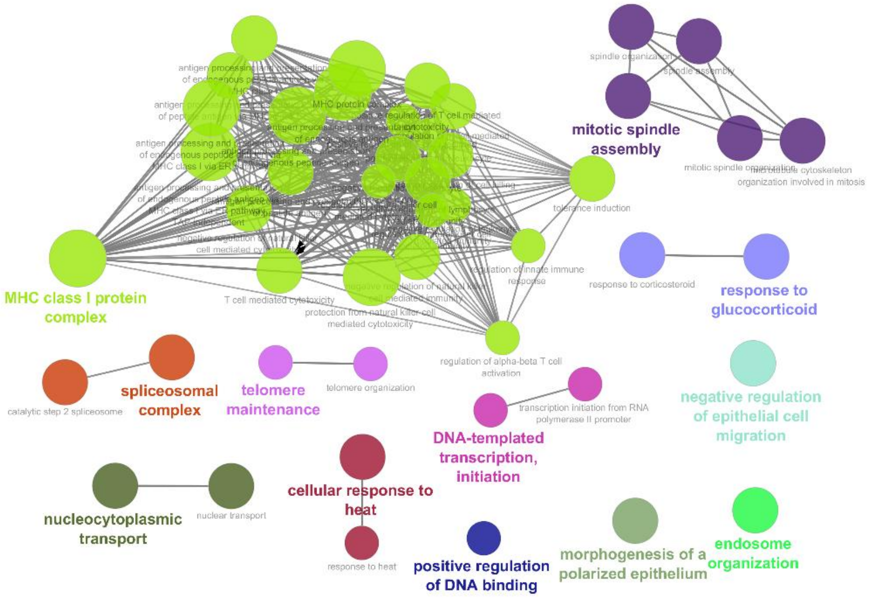

A
2873261
(49%)
1191677
(20%)
1761295
(30%)
HeLa
HEK293
B
120
(41%)
76
(26%)
100
(34%)
HeLa
HEK293
C
Density
3
2
1
0
0.00
0.25
0.50
0.75
1.00
NM modification distribution across normalized gene length
Cell_line
HEK293
HeLa

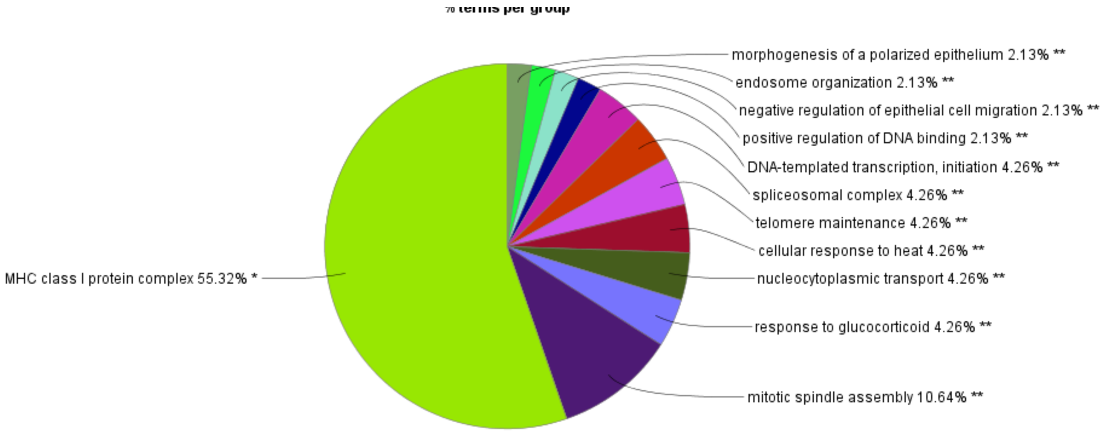
Figure 5

A


substantia nigra development
post-Golgi vesicle-mediated transport
regulation of protein insertion into mitochondrial membrane involved in apoptotic signaling pathway
regulation of mitochondrial outer membrane permeabilization involved in apoptotic signaling pathway
ER to Golgi transport vesicle membrane
antigen processing and presentation of endogenous peptide antigen via MHC class I via ER pathway
nuclear matrix
heterotrimeric G-protein complex
metanephros morphogenesis
telomere maintenance via telomere lengthening
mitotic spindle assembly
negative regulation of cell growth
cellular response to fatty acid
positive regulation of cardiac muscle hypertrophy
protein folding chaperone
regulation of response to reactive oxygen species
catalytic step 2 spliceosome
organelle envelope lumen
trans-Golgi network membrane
positive regulation of RNA splicing
negative regulation of response to endoplasmic reticulum stress
regulation of mRNA stability
ficolin-1-rich granule lumen
fibroblast migration
% terms per group
regulation of mitochondrial outer membrane permeabilization involved in apoptotic signaling pathway 22.38% **
antigen processing and presentation of endogenous peptide antigen via MHC class I via ER pathway 17.48% **
substantia nigra development 12.59% **
negative regulation of response to endoplasmic reticulum stress 0.7% **
trans-Golgi network membrane 0.7% **
negative regulation of cell growth 0.7% **
organelle envelope lumen 0.7% **
positive regulation of RNA splicing 0.7% **
regulation of mRNA stability 0.7% **
protein folding chaperone 0.7% **
catalytic step 2 spliceosome 0.7% **
ficolin-1-rich granule lumen 0.7% **
cellular response to fatty acid 0.7% **
regulation of response to reactive oxygen species 0.7% **
fibroblast migration 0.7% **
nuclear matrix 1.4% **
metanephros morphogenesis 1.4% **
heterotrimeric G-protein complex 1.4% **
positive regulation of cardiac muscle hypertrophy 1.4% **
mitotic spindle assembly 2.8% **
post-Golgi vesicle-mediated transport 6.29% **
telomere maintenance via telomere lengthening 6.99% **
regulation of protein insertion into mitochondrial membrane involved in apoptotic signaling pathway 7.69% **
ER to Golgi transport vesicle membrane 9.79% **

B


MHC class I protein complex
T cell mediated cytotoxicity
protection from natural killer cell mediated cytotoxicity
regulation of innate immune response
tolerance induction
regulation of alpha-beta T cell activation
MHC protein complex
antigen processing and presentation of endogenous peptide antigen via MHC class I via ER pathway
negative regulation of T cell mediated cytotoxicity
spindle organization
spindle assembly
mitotic spindle assembly
mitotic spindle organization
microtubule cytoskeleton organization involved in mitosis
response to corticosteroid
response to glucocorticoid
catalytic step 2 spliceosome
spliceosomal complex
telomere maintenance
telomere organization
transcription initiation from RNA polymerase II promoter
DNA-templated transcription, initiation
negative regulation of epithelial cell migration
nucleocytoplasmic transport
nuclear transport
cellular response to heat
response to heat
positive regulation of DNA binding
morphogenesis of a polarized epithelium
endosome organization


% terms per group
morphogenesis of a polarized epithelium 2.13% **
endosome organization 2.13% **
negative regulation of epithelial cell migration 2.13% **
positive regulation of DNA binding 2.13% **
DNA-templated transcription, initiation 4.26% **
spliceosomal complex 4.26% **
telomere maintenance 4.26% **
cellular response to heat 4.26% **
nucleocytoplasmic transport 4.26% **
response to glucocorticoid 4.26% **
mitotic spindle assembly 10.64% **
MHC class I protein complex 55.32% *