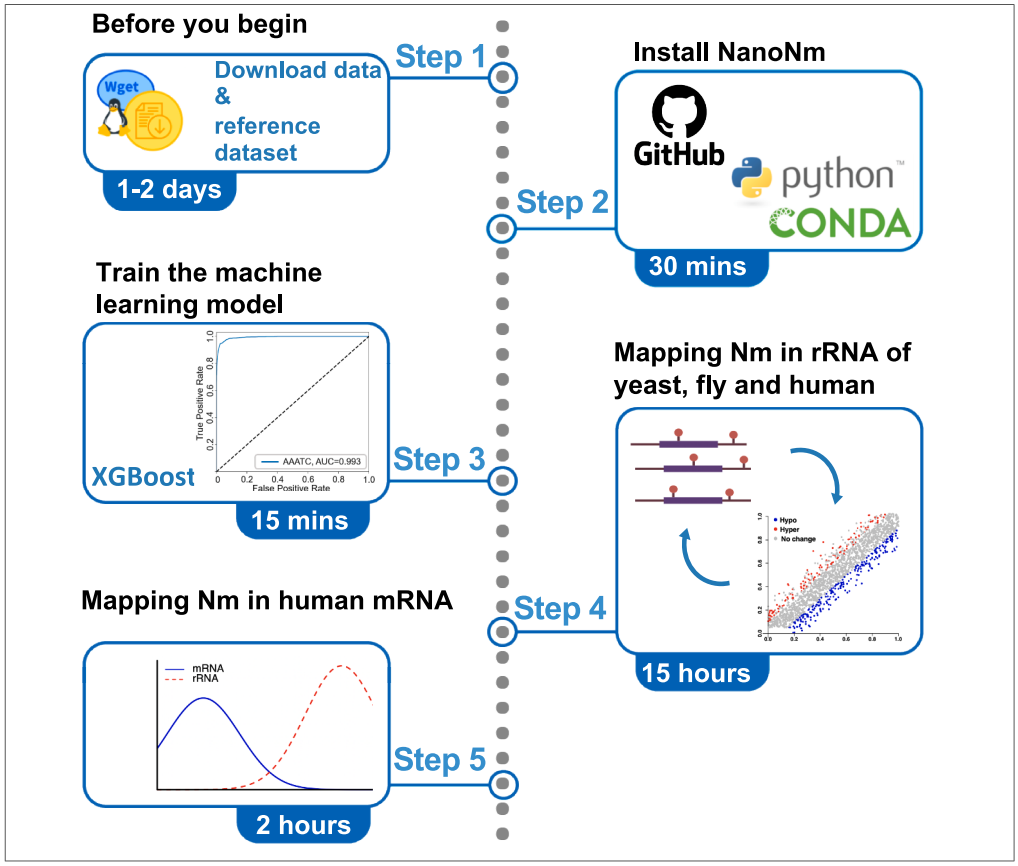
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for mapping 2′-O-methylation using nanopore direct RNA-seq data with NanoNm



2′-O-methylation (Nm) is pivotal in rRNA and the internal site of mRNA. Here, we present an integrated bioinformatics protocol for the identification of 2′-O-methylation using nanopore direct RNA sequencing (RNA-seq) data. We describe steps for software installation, data collection, and training the machine learning model. We then detail procedures for mapping 2′-O-methylation in both rRNA and mRNA in yeast and human cells.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Yanqiang Li, Jiayi Chen, Yunxia Wang, ..., Lili Zhang, Qi Cao, Kaifu Chen

liyq215@gmail.com (Y.L.)
kaifu.chen@childrens.harvard.edu (K.C.)

Highlights

Instructions for NanoNm software installation

Step-by-step guide for genome preparation and model training

Guidance for detecting 2′-O-methylation in rRNA from yeast, fly, and human samples

Procedures for 2′-O-methylation detection in human mRNA datasets

Li et al., STAR Protocols 6, 104003
September 19, 2025
https://doi.org/10.1016/j.xpro.2025.104003

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for mapping 2′-O-methylation using nanopore direct RNA-seq data with NanoNm

Yanqiang Li,[1,2,9,*] Jiayi Chen,[1,2,3] Yunxia Wang,[1,2] Dean Li,[1,2] Jiahan Li,[1,2,4] Yang Yi,[5,6] Lili Zhang,[1,2] Qi Cao,[5,6] and Kaifu Chen[1,2,7,8,10,*]

[1]Basic and Translational Research Division, Department of Cardiology, Boston Children's Hospital, Boston, MA 02115, USA
[2]Department of Pediatrics, Harvard Medical School, Boston, MA 02115, USA
[3]Department of Biomedical Informatics, Harvard Medical School, Boston, MA 02115, USA
[4]Commonwealth School, Boston, MA 02116, USA
[5]Department of Urology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611, USA
[6]Robert H. Lurie Comprehensive Cancer Center, Northwestern University Feinberg School of Medicine, Chicago, IL, USA
[7]Broad Institute of MIT and Harvard, Boston, MA 02215, USA
[8]Dana-Farber/Harvard Cancer Center, Boston, MA 02215, USA
[9]Technical contact
[10]Lead contact
*Correspondence: liyq215@gmail.com (Y.L.), kaifu.chen@childrens.harvard.edu (K.C.)
https://doi.org/10.1016/j.xpro.2025.104003

SUMMARY

2′-O-methylation (Nm) is pivotal in rRNA and the internal site of mRNA. Here, we present an integrated bioinformatics protocol for the identification of 2′-O-methylation using nanopore direct RNA sequencing (RNA-seq) data. We describe steps for software installation, data collection, and training the machine learning model. We then detail procedures for mapping 2′-O-methylation in both rRNA and mRNA in yeast and human cells.

For complete details on the use and execution of this protocol, please refer to Li et al.[1]

BEFORE YOU BEGIN

Data collection

2′-O-methylation widely occurs in rRNA, tRNA, and has recently been found in the internal site of mRNA. Detecting 2′-O-methylation using nanopore direct RNA-seq, combined with machine learning methods, offers an effective approach for identifying and measuring Nm.

1. Download the example dataset and feature dataset.

   Note: Duration of ∼2 h depending on the data and internet.

   Note: You can download the processed data from Zenodo: https://zenodo.org/records/14632831, which allows for easy re-execution of the entire pipeline to reproduce the figures in this protocol. Alternatively, you may use your own dataset to perform Nm mapping in your data.

```
>wget https://zenodo.org/records/14632831/files/rRNA.zip?download=1

>wget https://zenodo.org/records/14632831/files/example_data.zip?download=1

>wget https://zenodo.org/records/14632831/files/Yeast_Fly_feature_dataset.zip?download=1
```

2. Download the raw dataset (1-2 days depending on the data and internet) (Optional):

Note: This is the original raw dataset of FAST5 files, which is large and will take a long time.

```
mkdir -p raw_data/{human,yeast,fly,C4-2}

#Human rRNA:

>cd raw_data/human

>wget https://sra-pub-src-1.s3.amazonaws.com/SRZ190740/HS_rRNA_dRNA_fast5.tar.gz.1

#Human IVT (in vitro transcripts):

>wget https://sra-pub-src-1.s3.amazonaws.com/SRZ190744/IVT_Hs_mRNA_fast5.tar.gz.1

#Yeast rRNA of WT, snoR60, sno61 ad sno62:

>cd raw_data/yeast

>wget https://sra-pub-src-1.s3.amazonaws.com/SRZ190768/SC_rRNA_dRNA_fast5.tar.gz.1

>wget https://sra-pub-src-2.s3.amazonaws.com/ERR5317712/RNA475632.fast5.tar.gz.1

>wget https://sra-pub-src-1.s3.amazonaws.com/ERR5317713/RNA475633.fast5.tar.gz.1

>wget https://sra-pub-src-2.s3.amazonaws.com/ERR5317714/RNA475634.fast5.tar.gz.1

#Fly rRNA:

>cd raw_data/fly

>wget ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR606/ERR6065782/NT_rep1.tar.gz

>wget ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR606/ERR6066352/NT_rep2.tar.gz

>wget ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR606/ERR6066014/FBL_KD_rep1.tar.gz

>wget ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR606/ERR6066353/FBL_KD_rep2.tar.gz

#C4-2 mRNA, include rRNA:

>cd raw_data/C4-2

>wget https://sra-pub-src-1.s3.amazonaws.com/SRR20374459/nanopore_siCTRL_rep1.tar.gz.2

>wget https://sra-pub-src-1.s3.amazonaws.com/SRR20374458/nanopore_siCTRL_rep2.tar.gz.2

>wget https://sra-pub-src-1.s3.amazonaws.com/SRR20374457/nanopore_siFBL_rep1.tar.gz.2

>wget https://sra-pub-src-1.s3.amazonaws.com/SRR20374456/nanopore_siFBL_rep2.tar.gz.2
```

Our pipeline is compatible with both total RNA-seq and mRNA-seq data. To obtain reliable and robust detection of modifications using our pipeline, we recommend achieving a minimum sequencing depth of at least 20× per site. This level of coverage improves confidence and resolution. This integrated protocol to analyze RNA sequencing (RNA-seq including mRNA-seq and total RNA-seq) data from nanopore direct RNA-seq includes two steps: training models from rRNA and mapping Nm with base-resolution in rRNA and mRNA, using yeast, fly and human dataset as an example (Figure 1).

NanoNm was designed to detect Nm sites and measure the methylation levels from nanopore direct RNA-seq data in human and other species.[1] We recommend using the GRCh38 gene annotation GTF file from the GENCODE project for the human's dataset.[2]

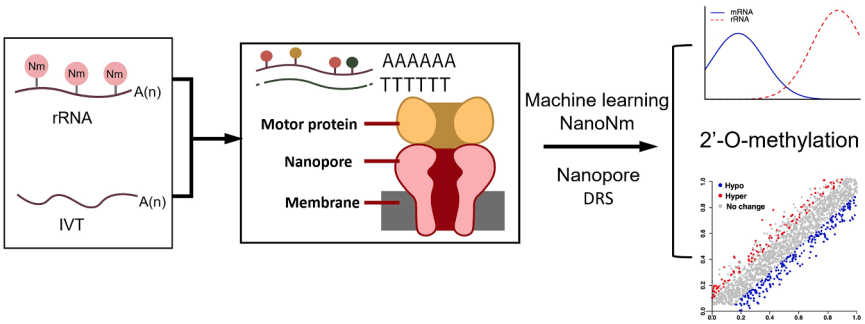


Figure 1. Flowchart showing the Nm mapping from nanopore RNA-seq with NanoNm

Step 1: RNA Preparation and Adapter Ligation. Ribosomal RNAs (rRNAs) containing native modifications were poly (A)-tailed, followed by adapter ligation and reverse transcription (RT). As a control, unmodified in vitro transcripts (IVTs) were also prepared. Step 2: Nanopore Sequencing and Signal Detection. During Nanopore sequencing, ionic current blockage events were recorded, each corresponding to the 5-mer RNA sequence passing through the pore constriction. Step 3: Machine Learning-Based Modification Detection. An *XGBoost* model was trained to distinguish between modified and unmodified 5-mers. This model was then applied to Nanopore direct RNA-seq data to quantify 2′-O-methylation levels in mRNA and rRNA.

In addition to detailing the processing of FAST5 data from nanopore RNA-seq and training the machine learning model for Nm detection, this protocol outlines the applications of NanoNm for analyzing rRNA in yeast and fly, as well as a toy example dataset of human rRNA and mRNA sourced from publicly available GEO datasets.

Note: Example data can be found at Zenodo: https://zenodo.org/records/14632831.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Raw nanopore sequence data for C4-2 cells	Li et al.[1]	NCBI GEO: GSE208837
Raw nanopore sequence data of Yeast	Begik et al.[3]	ENA database: ERA3183105
Raw nanopore sequence data of *Drosophila*	Sklias et al.[4]	NCBI GEO: PRJEB45722
Raw nanopore sequence of human rRNA	Jenjaroenpun et al.[5]	NCBI SRA: SRP166020
Custom scripts for this protocol	This paper	https://github.com/kaifuchenlab/STAR_Protocol
Big raw and processed data for this protocol	This paper	https://doi.org/10.5281/zenodo.14632831
Human mRNA sequence	Public dataset	https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_27/gencode.v27.transcripts.fa.gz
Human genome sequence	Public dataset	https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_38/GRCh38.p13.genome.fa.gz
Software and algorithms		
tombo v.1.5.1	Oxford Nanopore	https://github.com/nanoporetech/tombo
guppy v.3.6.1	Oxford Nanopore	https://nanoporetech.com/
ont_fast5_api 3.1.6	Oxford Nanopore	https://github.com/nanoporetech/ont_fast5_api
Minimap2 version 2.22-r1101	Li et al.[6]	https://github.com/lh3/minimap2
NanoNm for Nm detection	Li et al.	https://github.com/kaifuchenlab/NanoNm
BEDTools (v.2.29.2)	N/A	https://bedtools.readthedocs.io
SamTools (v.1.3.1)	Li et al.[7]	https://github.com/samtools/
minimap2 (v.2.17-r941)	Li et al.[6]	https://github.com/lh3/minimap2
python (v.3.7.3)	N/A	https://www.python.org/downloads/release/python-373/
Picard (2.26.2)	Broad Institute	https://broadinstitute.github.io/picard/

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
h5py (v.2.9.0)	N/A	https://pypi.org/project/h5py/
statsmodels (v.0.10.0)	N/A	https://www.statsmodels.org
joblib (v.0.16.0)	N/A	https://joblib.readthedocs.io
xgboost (v.0.80)	N/A	https://pypi.org/project/xgboost/
pysam (v.0.16.0.1)	N/A	https://pypi.org/project/pysam/
tqdm (v.4.39.0)	N/A	https://github.com/tqdm/tqdm
pycairo (v.1.19.1)	N/A	https://pypi.org/project/pycairo/
scikit-learn (v.0.22)	N/A	https://pypi.org/project/scikit-learn/
Other		
Computing platform: this protocol was performed on a Linux server with 80 Gb memory and 60 CPU cores	CentOS Linux release 7.6.1810	N/A

MATERIALS AND EQUIPMENT

Data (nanopore RNA-seq raw data, fast5 files– see data collection in before you begin).

Software

NanoNm and its Dependencies: NanoNm is implemented in Python 3. While different versions of the Python software and associated packages may work correctly with NanoNm, the authors use Python 3.7 and the required packages at the specified versions were listed in the key resources table.

STEP-BY-STEP METHOD DETAILS

Installing NanoNm and preparing the reference genome sequences and gene annotation files

Timing: ~30 min

Download the NanoNm software from GitHub to install NanoNm. An example of how to perform all steps of this protocol using example data is available on the project's GitHub repository: https://github.com/kaifuchenlab/NanoNm and Zenodo: https://zenodo.org/records/14632831.

1. Install NanoNm and other necessary software by running the following code:
   a. Install NanoNm by running the following code:

```
> git clone https://github.com/kaifuchenlab/NanoNm
```

   b. Install NanoNm dependencies.

```
> conda env create -f Nanopore.environment.yml
```

   c. Install ont-fast5 for multi_to_signgle_fast5.

```
> conda install -c bioconda ont-fast5-api
```

   d. Download software guppy from nanopore community and install.
   e. Install tombo.

```
>conda install -c bioconda ont-tombo
```

⚠ CRITICAL: Check if all required dependencies are downloaded and installed correctly. Alternatively, installing packages via conda will automatically check for and install the dependencies. However, errors during installation could occur while installing on computational environments (troubleshooting 1 and troubleshooting 2).

Prepare the reference genome sequences and gene annotation files

Timing: ~15 min

Based on our experience, NanoNm can map Nm sites in humans, mice, yeast, and flies. In this protocol, we used human, yeast, and fly species as examples. Use the human reference genome, transcript sequences, and gene annotation files to predict Nm in mRNA. In addition, use the rRNA sequences of human, yeast, and fly to predict Nm sites in rRNA.

2. Prepare reference genome sequences for human mRNA and rRNA.
   a. Download hg38 genome sequences from the Gencode website (https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_38/GRCh38.p13.genome.fa.gz).
   b. Unzip the downloaded genome file using unzip it into genome folder.
   c. Download the rRNA sequences for yeast and fly from the previously mentioned repository (https://github.com/kaifuchenlab/STAR_Protocol/rRNA or https://zenodo.org/records/14632831/files/rRNA.zip?download=1) and unzip the rRNA folder.

```
>cd genome
>samtools faidx GRCh38.p13.genome.fa
>picard CreateSequenceDictionary R=GRCh38.p13.genome.fa O=GRCh38.p13.genome.dict
>cd rRNA
>samtools faidx human.rRNA.fa
>picard CreateSequenceDictionary R=human.rRNA.fa O=human.rRNA.dict
>samtools faidx yeast.rRNA.fa
>picard CreateSequenceDictionary R=yeast.rRNA.fa O=yeast.rRNA.dict
>samtools faidx fly.rRNA.fa
>picard CreateSequenceDictionary R=fly.rRNA.fa O=fly.rRNA.dict
```

3. Prepare human reference transcript sequences in FASTA format.
   a. Download hg38 transcript sequences from the Gencode website into the genome folder (https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_27/gencode.v27.transcripts.fa.gz).
   b. Unzip the file in the genome folder.
4. Prepare gene and transcript annotation information in a tab-separated file as one of the input files for nanoNm software:
   a. Extract the transcript annotation information using "grep", "awk" and "cut" commands in Unix/Linux system.

```
>cd genome
>grep ">" gencode.v27.transcripts.fa| awk -F '[\t|]' '{print $6,"\t", $1}'|sed 's/>//' >v27.
gene.transcript.txt
```

```
>cut -f1,2 v27.gene.transcript.txt |awk '{FS="\t"}{a[$1] = a[$1]"\t"$2} END{for (i in a)
print i a[i]}' |sort >hg38.gene2transcripts.txt
```

b. Add the 18S, 28S, and 5.8S rRNA transcript entries manually to the top of the hg38.gene2transcripts.txt file. The manner of hg38.gene2transcripts.txt is as follows:

```
18S    18S

28S    28S

5.8S    5.8S

A1BG   ENST00000596924.1 ENST00000263100.7 ENST00000595014.1 ENST00000598345.1 ENST00000
600966.1
```

Read treatment, base calling, and mapping

Timing: 70 min

After installing all the dependencies and preparing the reference genome sequences and annotation files, it is time to run NanoNm. Currently, NanoNm is compatible only with UNIX/Linux systems. For detailed instructions on running NanoNm and updates to parameters, visit the project's GitHub repository (https://github.com/kaifuchenlab/NanoNm).

Note: Here we only provided the running time for the toy example dataset, which contains three fast5 files. The actual running time for real samples depends on the number of fast5 reads and the computational capacity of the HPC. This step is applicable for model training and Nm prediction in rRNA and mRNA.

5. Split the multiple fast5 to single fast5 files, using HS_rRNA_dRNA_0.fast5 as an example. Convert multi-read FAST5 files to single-read FAST5 files using the following script: (~1 min).

```
>multi_to_single_fast5 -i HS_rRNA_dRNA_0.fast5 -s $id --recursive -t 40
```

6. Run base calling using guppy_basecaller and filter the bad reads that have Phred quality scores of less than 7. Use the base-calling algorithm from Oxford Nanopore Technologies to convert raw nanopore signal data into FASTQ files and extract associated signal information:

```
>guppy_basecaller --input_path HS_rRNA_dRNA_0 --save_path HS_rRNA_dRNA_0_guppy --qscore_
filtering 7 --num_callers 40 --recursive --fast5_out --config rna_r9.4.1_70bps_hac.cfg --cpu_
threads_per_caller 10
```

Note: In the latest version of guppy (6.5.7, 11/15/2023), -qscore_filtering and -fast5_out, were no longer used and should be modified to -min_qscore and -post_out, respectively.

7. Resquiggle the signal of fast5 files to each transcript. Provide the re-squiggle algorithm with FAST5 input files containing raw signals and base calls. Map the base calls to a genome or transcriptome reference, then assign the raw signal to the reference using Tombo's resquiggle function.

```
>tombo resquiggle --rna --overwrite HS_rRNA_dRNA_0_guppy/workspace/ human_uniq.rRNA.fa
--processes 40 --fit-global-scale --include-event-stdev
```

Note: In this step, the output will be saved to HS_rRNA_dRNA_0_guppy/workspace/, overwriting the existing files in that directory.

8. Feature calling of each transcript, extract the 5-mer signal features from each transcript using the following script.

```
> find HS_rRNA_dRNA_0_guppy/workspace/ -name "*.fast5" >${id}_guppy.list

> python ../NanoNm/extract_raw_and_feature_fast_AUCG.py --cpu=30 --fl= HS_rRNA_dRNA_0_
guppy.list -o HS_rRNA_dRNA_0_guppy.feature --clip=5
```

Training the model with Nm sites in human rRNA

Timing: 15 min

9. Map the reads to the rRNA.

```
>minimap2 -ax map-ont -uf -k14 -x splice -t 20 human_uniq.rRNA.fa > HS_rRNA_dRNA_0_guppy.
feature.feature.fa|samtools view -@ 20 -bS - |samtools sort -@ 20 - > HS_rRNA_dRNA_0.rRNA.
sort.bam

>sam2tsv -r ../rRNA/human_uniq.rRNA.fa HS_rRNA_dRNA_0.rRNA.sort.bam > HS_rRNA_dRNA_0.
rRNA.sort.bam.tsv
```

10. Extract the 5-mer features of fast5 files of Nm sites in rRNA.

```
>python ../NanoNm/filter_get.fast5.py -i HS_rRNA_dRNA_0.rRNA.sort.bam.tsv -b rRNA.Nm1.bed
-f HS_rRNA_dRNA_0_guppy.feature.feature.tsv -o HS_rRNA_dRNA_0.fast5.rRNA.signal.
txt > HS_rRNA_dRNA_0.rRNA.feature.anno.txt

>cat *_guppy.feature.feature.tsv >> all.feature.mod.tsv

>cat kmer.txt|xargs -i -e echo "grep {} all.feature.mod.tsv >{}.mod.tsv''

>cat kmer.txt|xargs -i -e echo "grep {} all.feature.unmod.tsv >{}.unmod.tsv''
```

11. Train the 2′-O-methylation model from the nanopore direct RNA-seq data of rRNA.

```
>cat kmer.txt|xargs -i -e echo "python ../NanoNm/train_model_scale_pos_weight_Nm.py {}
>>Auc.scale1.txt & " |sh
```

The AUC and accuracy of NanoNm for two 5-mers as examples in rRNA trained are shown in Figure 2.

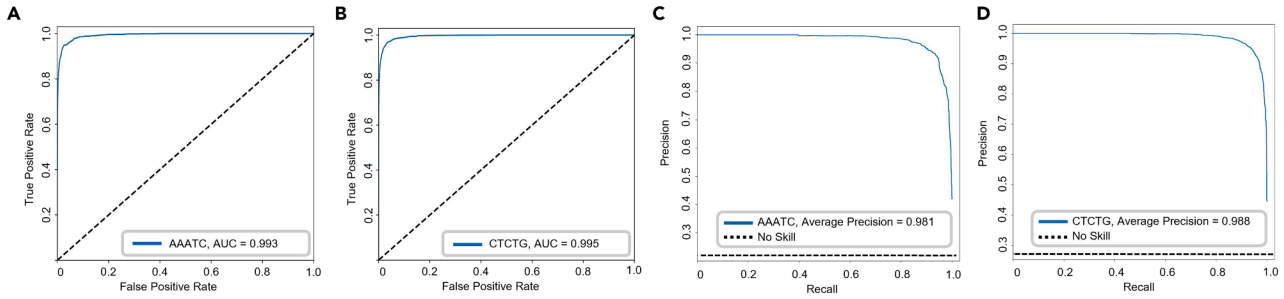


Figure 2. Performance of NanoNm on the rRNA training dataset

(A and B) Area under the curve (AUC) of NanoNm using AAAmTC and CTCmTG as representative examples.

(C and D) Accuracy of NanoNm on the same rRNA training dataset, illustrated with AAAmTC and CTCmTG examples.

Running the NanoNm in rRNA in yeast and fly

Timing: 15 h

12. Process the fast5 files of the yeast and fly rRNA dataset, and repeat step 3 for each sample.
13. Map the Nm sites in rRNA in WT yeast and in yeast strains with snoR60, snoR61, and snoR62 knock out.

```
#!/bin/bash

>cd Yeast_Fly_feature_dataset

>samples=("WT" "snoR60" "snoR61" "sno62")

>for id in "${samples[@]}"; do

  >gunzip "${id}_guppy.feature.feature.*.gz"

  > python ../NanoNm/predict_sites_Nm.final.py --model ../NanoNm/model --cpu 20 -i ${id}
_guppy.feature -o $id\_Nm_model -r ../rRNA/yeast.rRNA.fa -g ../rRNA/yeast.rRNA.fa -b
../rRNA/yeast_rRNA.list

  > python ../script/ratio2bed.py $id\_Nm_model/ratio.0.5.tsv $id\_yeast

>done
```

14. Compute the methylation ratio of each site in the rRNA of yeast, specifically at the positions 28S_187, 25S_1133, and 25S_1888, which is the target of snR60, snR61 and snR62, respectively (Table1).

Table 1. Methylation ratios of the known rRNA sites in yeast wild-type and snoRNA KO strains

	WT			snR60_KO			snR61_KO			snR62_KO		
Nm Position	Nm Ratio	Modified Reads	Total Reads	Nm Ratio	Modified Reads	Total Reads	Nm Ratio	Modified Reads	Total Reads	Nm Ratio	Modified Reads	Total Reads
25S_817	0.91	69	76	**0.03**	35	1056	0.90	942	1042	0.90	1043	1153
25S_1133	0.97	83	86	0.97	1117	1146	**0.04**	46	1095	0.97	1217	1255
25S_1888	0.93	105	113	0.98	1973	2021	0.98	1357	1389	**0.09**	214	2515

For each of the four sample groups (WT, snR60_KO, snR61_KO, and snR62_KO), the three columns indicate: the Nm ratio, the number of modified reads, and the total number of reads at the specific snoRNA-targeted Nm site.

15. Map the Nm in the rRNA of the fly from four samples.

```
#!/bin/bash

>cd Yeast_Fly_feature_dataset

>samples=("Fly_NT_rep1" "Fly_NT_rep2" "Fly_KD_rep1" "Fly_KD_rep2")

>for id in "${samples[@]}"; do

  >gunzip "${id}_guppy.feature.feature.*.gz"

  > python ../NanoNm/predict_sites_Nm.final.py --model ../NanoNm/model --cpu 20 -i ${id}
_guppy.feature -o $id\_Nm_model -r ../rRNA/fly.rRNA.fa -g ../rRNA/fly.rRNA.fa -b ../rRNA/
fly_rRNA.list

  > python ../script/ratio2bed.py $id\_Nm_model/ratio.0.5.tsv $id\_fly

>done
```

16. Plot the methylation ratio of each site in the rRNA of fly in siCTRL and siFBL with Excel software (Figure 3).

Note: This is a good time to compare the output results files with Table 1 and Figure 3 to ensure the protocol has run correctly.

Pause Point: Once we know the parameters we wish to use and have successfully run NanoNm, this is an ideal moment to pause and evaluate the results before proceeding with the optional steps.

Running the NanoNm in nanopore DRS in human cells

Timing: 2 h with 30 CPUs and 80 Gb memory

The NanoNm workflow is designed to analyze nanopore Direct RNA Sequencing (DRS) data in human cells, focusing on both rRNA and mRNA. This workflow processes nanopore DRS data by

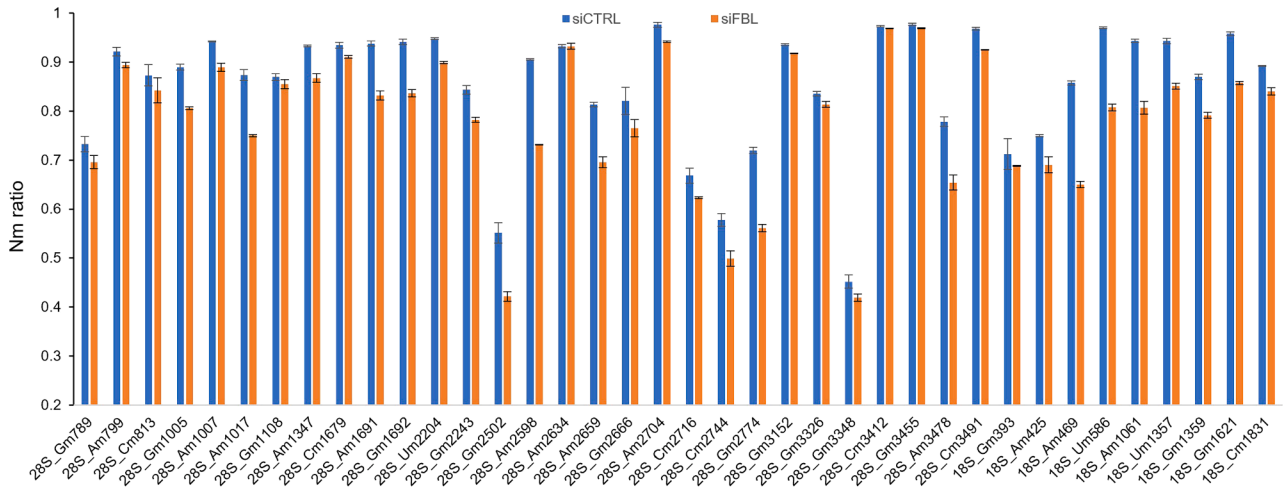


Figure 3. Bar plot depicting the methylation ratios of the known site in FBL knockdown and control flies

The plot compares the methylation levels between the FBL KO group and the controls. Data represent mean +/− s.d. of two biological replicates.

converting multi-read FAST5 files to single-read files, performing base calling with Guppy, aligning raw signals with Tombo, and generating a FAST5 file list. It extracts raw signals and AUCG features, followed by the identification and quantification of Nm modification sites using NanoNm.

Note: Here we only provided the running time for the toy example dataset, which contains four samples. The actual running time for the real sample is dependent on the number of fast5 files and the computational resource.

17. Detect Nm in rRNA of human nanopore test dataset.

```
#!/bin/bash

>cd example_data

>samples=("siCTRL_rRNA" "siFBL_rRNA")

>conda activate NanoNm

>for id in "${samples[@]}"; do

  >multi_to_single_fast5 -i "${id}.fast5" -s "$id" --recursive -t 40

  >guppy_basecaller --input_path "$id" --save_path "${id}_guppy" --num_callers 40 --qscore_filtering 7 --recursive --fast5_out --config rna_r9.4.1_70bps_hac.cfg --cpu_threads_per_caller 10

  >tombo resquiggle --rna --overwrite "${id}_guppy/workspace/" human.rRNA.fa --processes 40 --fit-global-scale --include-event-stdev

  >find "${id}_guppy/workspace/" -name "*.fast5" >"${id}_guppy.list"

  >python ../NanoNm/extract_raw_and_feature_fast_AUCG.py --cpu=30 --fl="${id}_guppy.list" -o "${id}_guppy.feature" --clip=5

  >python ../NanoNm/predict_sites_Nm.final.py --model ../NanoNm/model --cpu 10 -i ${id}_guppy.feature -o rRNA_QC7 -r ../rRNA/human.rRNA.fa -g ../rRNA/human.rRNA.fa -b ../rRNA/rRNA_gene2transcript.txt 2>rRNA.QC7.err

  >python ../scripts/ratio2bed.py rRNA_QC7/ratio.0.5.tsv "$sample"

>done
```

18. Detect Nm in mRNA of human nanopore test datasets.

```
#!/bin/bash

>cd example_data

>samples=("siCTRL_mRNA" "siFBL_mRNA")

>conda activate NanoNm

>for id in "${samples[@]}"; do

  >multi_to_single_fast5 -i "${id}.fast5" -s "$id" --recursive -t 40

  >guppy_basecaller --input_path "$id" --save_path "${id}_guppy" --num_callers 40 --recursive --fast5_out --qscore_filtering 7 --config rna_r9.4.1_70bps_hac.cfg --cpu_threads_per_caller 10

  >find "${id}_guppy/workspace/" -name "*.fast5" >"${id}_guppy.list"

  >python ../NanoNm/extract_raw_and_feature_fast_AUCG.py --cpu=30 --fl="${id}_guppy.list" -o "$id" --clip=5
```

```
  >python ../NanoNm/predict_sites_Nm.final.py --model ../NanoNm/model -i "$id" -o "${id}
_Nm_model" -r ../genome/gencode.v27.transcripts.fa -g ../genome/GRCh38.p13.genome.fa -b
../genome/hg38.gene2transcripts.txt

  >python ../scripts/ratio2bed.py "${id}_Nm_model/ratio.0.5.tsv" siCTRL_mRNA

>done
```

19. Combine the rRNA and mRNA ratio files from the two samples, siFBL and siCTRL (Figure 4).

```
>python ../scripts/merge_mRNA.py siCTRL_mRNA.filter.bed siFBL_mRNA.filter.bed Human_mRNA.
ratio.txt

>python    ../scripts/merge_human.rRNA.py    human.rRNA.Nm.bed    siCTRL_rRNA.filter.bed
siFBL_rRNA.filter.bed Human_rRNA.ratio.txt

>Rscript ../scripts/Human_siFBL_siCTRL_rRNA_mRNA.plot.R
```

EXPECTED OUTCOMES

By the end of processing the example dataset, we will have multiple output files. An example of the key result is siCTRL.mRNA.filter.bed (Figure 5), which contains the chromosome information, position, methylation read counts, and total coverage and methylation ratio. We can adjust the cutoff including methylation read (modified) counts (5 by default), coverage (total > 20 by default), and methylation levels, to choose the most confident methylation sites in mRNA. Additionally, there are parameters such as read counts (–support), and probability cutoff (–proba) available to fine-tune the cutoffs before obtaining the final results.

LIMITATIONS

The accuracy of Nm identification using NanoNm is inherently influenced by the sequencing noise characteristic of nanopore technology. This is particularly critical for Nm sites in mRNA, where sequencing errors may affect the detection precision. Although we can extract the reads with high quality, false-positive sites are present due to noise signals. For example, NanoNm shows a false-positive rate of 0.17% (12 false positives out of 6989 known negatives) and false-negative rate of 0.93% (1 false negative out of 107 known positives) for detecting Nm sites in human rRNAs.

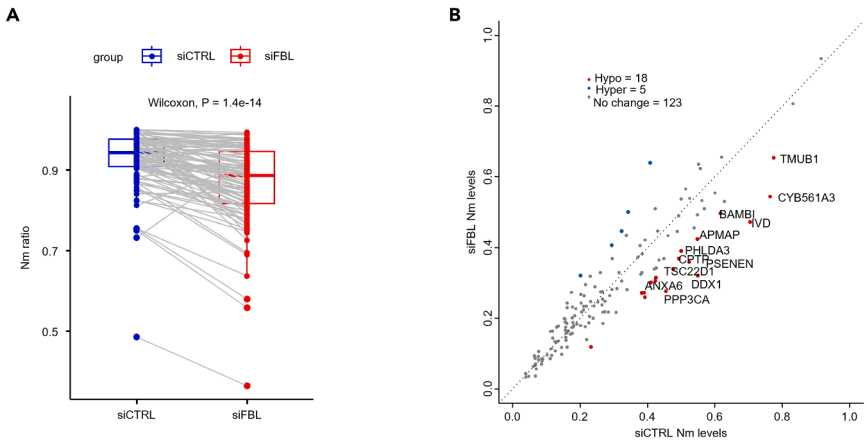


Figure 4. Detection of Nm sites in rRNA and mRNA by NanoNm in the test dataset

Methylation ratios at known Nm sites in rRNA (A) and representative Nm sites in mRNA (B) from the test dataset.

chr_postion	start	end	methylation ratio	gene	modified	total
chr1_1324925	1324925	1324926	0.3698630136986301	CPTP	27	73
chr1_1326993	1326993	1326994	0.19090909090909092	CPTP	21	110
chr1_1327251	1327251	1327252	0.2608695652173913	CPTP	30	115
chr1_1327279	1327279	1327280	0.46956521739130436	CPTP	54	115
chr1_1327913	1327913	1327914	0.23484848484848486	CPTP	31	132
chr1_1328066	1328066	1328067	0.15555555555555556	CPTP	21	135
chr1_1328563	1328563	1328564	0.1643835616438356	CPTP	24	146
chr1_1328574	1328574	1328575	0.5	CPTP	73	146
chr1_1328808	1328808	1328809	0.4482758620689655	CPTP	65	145
chr1_1328860	1328860	1328861	0.4112903225806452	CPTP	51	124

Figure 5. Screenshot of siCTRL.mRNA.filter.bed

The text on the top of image indicates the corresponding annotation for each column of the file.

In this method, we extracted training features, including the median, standard deviation, mean, and dwell time of the Nanopore signal for the 5-mer around each Nm site in the rRNA. Moreover, the current model, trained predominantly on human 107 known rRNA sites, covers only 100 out of the 1024 (4^5) possible 5-mer sequences from the native transcripts. This limited representation may reduce its generalizability when applied to all Nm site detection in mRNA or rRNA of other species, like flies or yeast. Expanding the model to include a broader range of 5-mers and incorporating data generated using new kits, such as the R004 DRS kit from nanopore, would likely improve the accuracy and robustness of Nm site identification.

TROUBLESHOOTING

Problem 1

Install the software-dependent packages (Step 1).

Potential solution

Whenever possible, use Anaconda (https://www.anaconda.com/) to install Python 3.7 and its dependent packages. To avoid potential conflicts with installed Python packages, you can create a new conda environment to install all the necessary packages using the "conda create" command.

Problem 2

Error for ont-tombo install: conda install -c bioconda ont-tombo, "reporting: ERROR: Command errored out with exit status 1".

Potential solution

Tombo software needs an old version of Cython,

```
>pip install Cython==0.29.36
```

Make sure that Python and other dependencies versions in the key resources table are appropriate.

You can use Anaconda to specify the version of the installed package, using the following commands:

```
>conda install <package>=<version>
```

Or use GitHub tag to specify the package version.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Kaifu Chen (kaifu.chen@childrens.harvard.edu).

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Yanqiang Li (liyq215@gmail.com).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- Nanopore data have been deposited at GEO and are publicly available. The accession number is listed in the key resources table.
- The example dataset and custom codes for this study are available at Zenodo: https://doi.org/10.5281/zenodo.14632831 and GitHub: https://github.com/kaifuchenlab/STAR_Protocol, respectively.

ACKNOWLEDGMENTS

This project is supported in part by NIH grants R01GM138407 (K.C.), R01GM125632 (K.C.), R01HL148338 (K.C.), R01HL133254 (K.C.), R01HL174928 (K.C.), R01CA278832 (K.C. and Q.C.), R01CA208257 (Q.C.), R01CA256741 (Q.C.), and R01HL155632 (L.Z.); U.S. Department of Defense grants W81XWH-17-1-0357 (Q.C.), W81XWH-19-1-0563 (Q.C.), W81XWH-20-1-0504 (Q.C.), and HT9425-23-1-0661 (Y.Y.); the Prostate SPORE P50CA180995 Development Research Program (Q.C.); the Polsky Urologic Cancer Institute of the Robert H. Lurie Comprehensive Cancer Center of Northwestern University at Northwestern Memorial Hospital (Q.C.); and the NIH P50CA180995 SPORE in Prostate Cancer Career Enhancement Award (Y.Y.). The funders had no role in the study design, data collection, analysis, decision to publish, or preparation of the manuscript.

AUTHOR CONTRIBUTIONS

Y.L. wrote the protocol with comments from K.C., J.C., Y.W., D.L., Y.Y., L.Z., and Q.C.; code testing and processing were carried out by Y.L. with the help of J.C., Y.W., and J.L.; all authors contributed to the article and approved the submission of this manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Li, Y., Yi, Y., Gao, X., Wang, X., Zhao, D., Wang, R., Zhang, L.-S., Gao, B., Zhang, Y., Zhang, L., et al. (2024). 2′-O-methylation at internal sites on mRNA promotes mRNA stability. Mol. Cell *84*, 2320–2336.e6. https://doi.org/10.1016/j.molcel.2024.04.011.

2. Frankish, A., Carbonell-Sala, S., Diekhans, M., Jungreis, I., Loveland, J.E., Mudge, J.M., Sisu, C., Wright, J.C., Arnan, C., Barnes, I., et al. (2023). GENCODE: reference annotation for the human and mouse genomes in 2023. Nucleic Acids Res. *51*, D942–D949. https://doi.org/10.1093/nar/gkac1071.

3. Begik, O., Lucas, M.C., Pryszcz, L.P., Ramirez, J.M., Medina, R., Milenkovic, I., Cruciani, S., Liu, H., Vieira, H.G.S., Sas-Chen, A., et al. (2021). Quantitative profiling of pseudouridylation dynamics in native RNAs with nanopore sequencing. Nat. Biotechnol. *39*, 1278–1291. https://doi.org/10.1038/s41587-021-00915-6.

4. Sklias, A., Cruciani, S., Marchand, V., Spagnuolo, M., Lavergne, G., Bourguignon, V., Brambilla, A., Dreos, R., Marygold, S.J., Novoa, E.M., et al. (2024). Comprehensive map of ribosomal 2′-O-methylation and C/D box snoRNAs in Drosophila melanogaster. Nucleic Acids Res. *52*, 2848–2864. https://doi.org/10.1093/nar/gkae139.

5. Jenjaroenpun, P., Wongsurawat, T., Wadley, T. D., Wassenaar, T.M., Liu, J., Dai, Q., Wanchai, V., Akel, N.S., Jamshidi-Parsian, A., Franco, A.T., et al. (2021). Decoding the epitranscriptional landscape from native RNA sequences. Nucleic Acids Res. *49*, e7. https://doi.org/10.1093/nar/gkaa620.

6. Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics *34*, 3094–3100. https://doi.org/10.1093/bioinformatics/bty191.

7. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R.; 1000 Genome Project Data Processing Subgroup (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics *25*, 2078–2079. https://doi.org/10.1093/bioinformatics/btp352.