EpiNano: Detection of m^{6}A RNA Modifications using Oxford Nanopore Direct RNA Sequencing

Huanle Liu[1,#], Oguzhan Begik[1,2,3,#] and Eva Maria Novoa[1,2,3,4,*]

[1]Centre for Genomic Regulation (CRG), The Barcelona Institute of Science and Technology, Dr. Aiguader 88, Barcelona 08003, Spain

[2]Department of Neuroscience, Garvan Institute of Medical Research, Darlinghurst, NSW, 2010, Australia

[3]St. Vincent's Clinical School, UNSW Sydney, Darlinghurst, NSW, 2010, Australia

[4]Universitat Pompeu Fabra (UPF), Barcelona, Spain

[#] Equal contribution

* Correspondence to: Eva Maria Novoa (eva.novoa@crg.eu)

Running title: Identifying m^{6}A modifications using nanopore sequencing

Abstract

RNA modifications play pivotal roles in the RNA life cycle and RNA fate, and are now appreciated as a major post-transcriptional regulatory layer in the cell. In the last few years, direct RNA nanopore sequencing (dRNA-seq) has emerged as a promising technology that can provide single molecule resolution maps of RNA modifications in their native RNA context. While native RNA can be successfully sequenced using this technology, the detection of RNA modifications is still challenging. Here, we provide an upgraded version of *EpiNano* (version 1.2), an algorithm to predict m^6A RNA modifications from dRNA-seq datasets. The latest version of *EpiNano* contains models for predicting m^6A RNA modifications within dRNA-seq data that has been base-called with *Guppy*. Moreover, it can now train models with features extracted from both base-called dRNA-seq FASTQ data as well as from raw FAST5 nanopore outputs. Finally, we describe how *EpiNano* can be used in standalone mode to extract base-calling 'error' features and current intensity information from dRNA-seq datasets. In this chapter, we provide step-by-step instructions on how to produce *in vitro* transcribed constructs to train *EpiNano*, as well as detailed information on how to use *EpiNano* to train, test and predict m^6A RNA modifications in dRNA-seq data.

Keywords: Oxford Nanopore Technologies, direct RNA sequencing, native RNA, RNA modification, base-calling 'errors', in vitro transcription, Support Vector Machine, N6-methyladenosine, nanopore sequencing

1 Introduction

Chemical modifications in RNA have been well-documented for over a half century. In the 1950s, pseudouridine was discovered to be the most abundant RNA modification present in cellular RNAs [1]. Later studies showed that internal modifications were also present in mRNAs and long-noncoding RNAs (lncRNAs), revealing that N6-methyladenosine (m^6A) was the most abundant mRNA modification [2–5]. Interest in functionally dissecting and mapping RNA modifications transcriptome-wide re-emerged in the past decade, largely triggered by the discovery of the biological function of m^6A demethylases FTO [6] and ALKBH5 [6, 7]. At the same time, the availability of novel methods to map m^6A RNA modifications transcriptome-wide (m^6A-seq) opened new possibilities to study m^6A modifications across a wide variety of conditions and tissues [8, 9]. Using m6A-Seq, m^6A RNA modifications were found to play pivotal roles in a wide variety of biological processes, including cellular differentiation [10–13] and sex determination [14, 15], among others.

While the field of RNA modifications owes largely to improved methods for detection using next-generation sequencing (NGS) technologies [8, 9, 16–18], these methods present several caveats: i) they lack single molecule resolution [19]; ii) they are limited to those RNA modifications for which there are commercial antibodies and chemicals that are selective towards a particular RNA modification [20]; iii) they cannot provide isoform-specific information, due to the short-read nature of Illumina-based technologies; and iv) they are limited to those regions that can be reverse transcribed and/or PCR-amplified. Direct RNA nanopore sequencing (dRNA-seq) offers an alternative to NGS-based methods to detect RNA base modifications in a transcriptome-wide fashion [21]. Indeed, it is capable of sequencing native RNA molecules, including RNA modifications, in their native RNA context, and with single molecule resolution.

In this chapter, we describe the use of *EpiNano*, an algorithm to detect RNA base

modifications from data generated using direct RNA nanopore sequencing. We exemplify the usage of *Epinano* to detect m^6A RNA modifications both in *in vitro* transcribed constructs as well as in *in vivo* datasets.

2 Materials

2.1 *In vitro* transcribed RNAs with RNA modifications

1. Plasmids containing 'curlcake' sequences, to be used as templates for the *in vitro* transcription reaction. The 'curlcakes' are a set of synthetic sequences that comprise all possible 5-mers (median occurrence of each 5-mer=10), while minimizing the RNA secondary structure, and thus can be used to systematically identify the perturbations of current intensity caused by the presence of a given RNA modification in all possible 5-mer contexts (n=1024). Plasmids can be obtained from Addgene:

pUC57-Curlcake1 (2329 bp, Addgene # 139340)

pUC57-Curlcake2 (2543 bp, Addgene # 139341)

pUC57-Curlcake3 (2678 bp, Addgene # 139342)

pUC57-Curlcake4 (2795 bp, Addgene # 139343)

2. Competent *E. coli* cells: 10-beta Competent *E. coli* High Efficiency.

3. SOC medium: 20 g/L Tryptone, 5 g/L Yeast Extract, 4.8 g/L MgSO4, 3.603 g/L dextrose, 0.5 g/L NaCl, 0.186 g/L KCl.

4. Agar Plates with Ampicillin (100 μg/mL).

5. LB Broth.

6. Ampicillin (100 mg/mL): Dissolve 1 g of sodium ampicillin in 10 mL nuclease-free water

7. Qiagen Plasmid Maxi Kit.

8. Molecular Grade Ethanol.

9. Nuclease-Free water.

10. BSA (NEB).

11. Cutsmart Buffer, BamHI-HF, and EcoRV-HF.

12. Phenol:Chloroform:Isoamyl Alcohol 25:24:1, Saturated with 10 mM Tris, pH 8.0, 1 mM EDTA.

13. Chloroform.

14. 3 M Sodium Acetate, pH 5.2 (Molecular Biology Grade).

15. Pellet Paint® Co-Precipitant or glycogen.

16. Agarose.

17. 1xTBE Buffer: Dissolve 10.8 g Tris and 5.5 g Boric acid in 900 mL distilled water. Add 4 mL 0.5 M Na2EDTA, pH 8.0. Adjust the volume to 1 L.

18. Gel Red Nucleic Acid Stain (10,000X).

19. Gel Loading Dye, Purple (6X).

20. AmpliScribe™ T7 High Yield Transcription Kit.

21. N6-Methyl-ATP (Jena Bioscience).

2.2 Clean-up of IVT RNAs

1. RNeasy Mini Kit.
2. Nuclease-Free water.
3. Qubit™ RNA HS Assay Kit.

2.3 PolyA tailing, Clean-up and Quality Check

1. SUPERase In.
2. E. coli Poly(A) Polymerase and accompanying buffer.
3. 10 mM ATP.
4. Agencourt RNA Clean XP Beads.

4. Nuclease-free water.
5. Molecular Grade Ethanol.
6. Magnetic separator, suitable for 1.5 mL Eppendorf tubes.
7. Hula Mixer.
8. Nanodrop or similar.
9. Agilent Tapestation and accompanying reagents: RNA ScreenTape Sample Buffer, RNA ScreenTape Ladder, RNA ScreenTape, Optical tube strip caps (8x Strip), Optical tube strips (8x Strip), and Loading Tips.

2.4 Direct RNA nanopore sequencing library preparation

1. Direct RNA Sequencing Kit (SQK-RNA002).
2. Flow Cell Priming Kit (EXP-FLP001).
3. 1.5 mL Eppendorf DNA LoBind tubes 0.2 mL thin-walled PCR tubes.
4. Nuclease-free water.
5. Freshly prepared 70 % ethanol in nuclease-free water.
6. SuperScript III Reverse Transcriptase and accompanying reagents.
7. 10 mM dNTP solution.
8. Concentrated T4 DNA Ligase 2M U/mL (NEB).
9. NEBNext® Quick Ligation Reaction Buffer.
10. Agencourt RNA Clean XP beads.
11. Qubit RNA HS Assay Kit, Qubit dsDNA HS Assay Kit, and Qubit™ Assay Tubes.
12. Hula mixer (gentle rotator mixer).
13. Magnetic separator, suitable for 1.5 mL Eppendorf tubes.

2.5 Software

1. *Guppy*, version 3.1.5 or later (https://community.nanoporetech.com/): base-calling algorithm
2. *Minimap2* (https://github.com/lh3/minimap2): mapping algorithm
3. *Samtools*, version 0.1.19 (https://github.com/samtools/samtools): sorting and manipulation of BAM files
4. *Sam2tsv*, version a779a30d6af485d9cd669aa3752465132cf21eec: conversion of BAM to plain text files and reorganization of the read-reference alignment
5. *EpiNano*, version 1.2 (https://github.com/enovoa/EpiNano): extraction of base-calling 'error' features from FASTQ files and BAM/SAM alignment files, and optionally current intensity information from *Nanopolish* event align outputs
6. *Nanopolish*, version 0.11.2 or later (https://github.com/jts/nanopolish): extraction of event information from FAST5 files

2.6. Datasets

1. Direct RNA sequencing data from *S. cerevisiae* polyA(+)-selected RNA, both from WT and ΔIME4 strains, can be found in SRA (SRP184486, FAST5) and GEO (code: GSE126213, FASTQ).
2. Direct RNA sequencing data from *S. cerevisiae* 25s ribosomal RNA, both from WT and snR34 knockout strains, can be found in the *EpiNano* GitHub repository (https://github.com/enovoa/EpiNano/tree/master/test_data/).
3. Direct RNA sequencing data from *in vitro* transcribed synthetic constructs ('curlcakes') containing unmodified ('unm') as well as m6A-modified ('mod') nucleosides can be found in SRA (code: SRP174366, FAST5) and GEO (code: GSE124309, FASTQ).

3 Methods

3.1. Preparation of modified and unmodified *in vitro* transcribed constructs to train *EpiNano*

3.1.1. Plasmid transformation and isolation

1. Adjust the water bath to 42 °C and place the competent *E. coli* cells into ice.

2. Add 3 µL of the Curlcake plasmid into a 25 µL volume of competent cells and mix well. Incubate on ice for 30-45 min.

3. Heat shock the cells at 42 °C for 45 sec, then incubate on ice for 5 min.

4. Add 500 µL warm SOC/SOB medium. Don't pipette and incubate at 37 °C for 1 hr shaking at 220 rpm (use thermomixer).

5. Spread 200 µL transformant on Agar Plates containing 100 µg/µl Ampicillin.

6. Incubate in a 37 °C incubator overnight.

7. Next day, pick a colony and inoculate it in 200 mL LB (with Ampicillin 100 µg/µl) for O/N culture (for Maxiprep).

8. Centrifuge the culture in 250 mL centrifuge vessels at 10,000 X g for 10 min (either disposable or reusable autoclaved ones) and isolate plasmids DNA using the Plasmid Maxi Kit according to the manufacturer's instructions, resuspending the final DNA pellet in 500 µl RNAse-free water.

3.1.2. Enzymatic Digestion of Plasmids and DNA cleanup

1. Digest 10 µg DNA (see ***Note 1)*** in 250 µL volume as outline below and incubate 4 hr (or O/N) at 37 °C.

BSA (100X)	2.5 µL
Cutsmart Buffer (10X)	25 µL
BamHI-HF	1 µL (20 units)

EcoRV-HF	1 µL (20 units)
DNA	x µL
dH2O	Up to 250 µL

2. To clean up the plasmid DNA, add one volume of phenol:chloroform:isoamyl alcohol (25:24:1) to your sample, shake by hand thoroughly for approximately 20 sec.

3. Centrifuge at room temperature for 5 min at 16,000 × g. Carefully remove the upper aqueous phase and transfer the layer to a fresh tube. Be sure not to carry over any phenol during pipetting.

4. Repeat the two previous steps until no protein is visible at the interface.

5. Mix an equal volume of chloroform with the aqueous phase. Shake briefly and centrifuge at 12,000 x g for 3-5 min.

6. Mix the upper phase with 0.1X sodium acetate and 2.5X absolute ethanol and 1 µL Glycogen or 2µL Pellet paint. Incubate for 15 min RT or overnight -20 °C or 1 hr at -80 °C.

7. Centrifuge the sample at 4 °C for 30 min at 16,000 × g to pellet the DNA.

8. Carefully remove the supernatant without disturbing the DNA pellet. Add 150 µL of 70 % ethanol.

9. Centrifuge the sample at 4 °C for 2 min at 16,000 × g. Carefully remove the supernatant.

10. Allow the pellet to air dry and resuspend the pellet in 25 µL of RNAse-free H2O.

11. Measure the DNA concentration using a Nanodrop or similar.

3.1.4. Agarose Gel Electrophoresis

Perform gel electrophoresis to confirm plasmid DNA digestion.

1. Dissolve 1 g Agarose in 100 mL 1xTBE/TAE Buffer in a microwavable flask and microwave for 1-3 min until the agarose is completely dissolved.

2. Let it cool on the benchtop for 5 min and add 10 µL Gel Red Nucleic Acid Stain (10,000X) and mix.

3. Pour the mixture into the gel container and allow to set.

4. Mix 1 µL digested DNA sample with 1 µL Loading Dye (6X) and 4 µL nuclease-free water.

5. Load the DNA sample into the well and run the gel for 30 min at 100 V.

6. Image the gel using a Bio-Rad gel imager or similar to ensure DNA is completely linearized.

3.1.5. *In vitro* transcription (IVT) using Ampliscribe T7-Flash Transcription Reaction

1. For each plasmid, set up an IVT reaction by combining the following reaction components from the AmpliScribe™ T7 High Yield Transcription Kit with linearized DNA from the step above at RT, in the order listed below (See **Note 2**). Substitution of m6ATP in the place of ATP will result in the generation of RNA containing m6A residues.

Component	**Volume**
RNase-Free water	Up to 20 µL
Linearized template DNA	1 µg
AmpliScribe T7-Flash 10X Reaction Buffer	2 µL
100 mM DTT	2 µL
100 mM ATP (or m6ATP)	1.8 µL
100 mM CTP	1.8 µL
100 mM GTP	1.8 µL

100 mM UTP	1.8 µL
Riboguard RNase Inhibitor	0.5 µL
Ampliscribe T7 Flash Enzyme Solution	2 µL
Total	20 µL

2. Incubate the reaction for 4 hr at 42 °C.

3. Add 2 µL of RNAse-Free DNAse I to the reaction and incubate for 20 min at 37 °C.

3.1.6. RNA Cleanup using RNeasy Qiagen Kit

1. Bring the volume of the IVT reaction to 100 µL with RNase-free water.

2. Follow the step by step instructions of the RNeasy Kit according to the manufacturer's protocol.

3. To elute RNA, pipette 20 µL RNase-free water directly onto the RNeasy Mini column membrane. Centrifuge for 15 min at ≥ 8000 x g to elute.

4. Add another 20 µL RNase-free water directly onto the RNeasy Mini column membrane and centrifuge for 15 min at ≥ 8000 x g to elute.

5. Measure the quality/quantity of the eluate.

3.1.7. PolyA tailing

1. Mix the following reagents to proceed with polyA tailing reaction:

Reagent	**Volume**
Purified RNA (sample) (step 3.1.5 above)	1-10 µg in 15.5 µL nuclease free water

RNAse Inhibitor (SUPERaseIN)	0.5 µL
10X E. coli Poly(A) Polymerase Reaction Buffer	2 µL
ATP (10 mM)	1 µL
E. coli Poly(A) Polymerase	1 µL
Total	20 L

2. Incubate the reaction at 37 °C for 20 min and proceed immediately to cleanup.

3.1.7. Bead Cleanup of RNA using RNA Clean XP Beads

1. Vortex the RNA clean XP Bead stock until homogenous and add 36 µL (1.8X) RNA clean beads to the RNA sample.

2. Mix by pipetting up and down 10x gently and incubate for 5 min at RT.

3. Place the reaction on the magnet and let it settle for 5-10 min.

4. Slowly aspirate the solution and discard.

5. Add 70 % fresh ethanol and incubate for 30 sec at RT. Remove ethanol completely and air dry for 2 min.

6. Add 20 µL water and pipette the beads up and down. Incubate 5 min at RT.

7. Use a magnetic stand to separate the beads from the RNA. Transfer the RNA solution into a new tube.

8. Measure the quality and quantity of the RNA and confirm the polyA tailing.

3.1.8. Quality check of PolyA tailed RNAs using TapeStation

1. Load both non-polyA-tailed and polyA-tailed IVT constructs into TapeStation (see **Note 3**) according to the manufacturer's instructions. Expected results are displayed in Fig. 1.

[Figure 1 near here]

3.2. Direct RNA Sequencing Library Preparation

3.2.1. Preparing input RNA

1. Pool each Curlcake (for unmodified and m6A modified separately) into a DNA LoBind tube that will contain 200 ng from each (800 ng total).
2. Adjust the volume to 9 µL with Nuclease-free water and mix thoroughly by inversion.
3. Spin down briefly in a microfuge.

3.2.2. Adapter ligation

1. In a 0.2 mL thin-walled PCR tube, mix the reagents in the following order including some components from the Direct RNA Sequencing Kit:

Reagent	**Volume (µL)**
5X NEBNext Quick Ligation Reaction Buffer	3
RNA	9
RT Adapter (RTA)	1
Concentrated T4 DNA Ligase	1.5
RNAse Inhibitor (SUPERase)	0.5

Total	15

2. Mix by pipetting and spin down.

3. Incubate the reaction for 10 min at RT. In the meantime, proceed to the reverse transcription step.

3.2.3. Reverse Transcription and cleanup

1. Mix the following reagents together to make the reverse transcription master mix:

Reagent	**Volume (µL)**
Nuclease-free water	9
10 mM dNTPs	2
5X First-Strand Buffer	8
0.1 M DTT	4
Total	23

2. Add the master mix to the 0.2 mL PCR tube containing the RT adaptor ligated RNA from the “RT Adapter ligation” step above. Mix by pipetting.

3. Add 2 µl SuperScript III reverse transcriptase to the reaction and mix by pipetting.

4. Place the tube in a thermal cycler and incubate at 50 °C for 50 min, 70 °C for 10 min, and bring the sample to 4 °C before proceeding to the next step.

5. Transfer the sample to a 1.5 mL DNA LoBind Eppendorf tube.

6. Resuspend the stock of Agencourt RNAClean XP beads by vortexing and add 72 µL beads to the reverse transcription reaction and mix by pipetting.

7. Incubate on a Hula mixer for 5 min at RT.

8. Prepare 200 µL of fresh 70 % ethanol in nuclease free water.

9. Spin down the sample and pellet on a magnet.

10. Keep the tube on the magnet and wash the beads with 150 µL 70% ethanol without disturbing the pellet.

11. Remove the ethanol and discard. Spin down tubes, place back on the magnetic rack and remove any residual ethanol.

12. Remove the tube from the magnetic rack and resuspend the pellet in 20 µL nuclease-free water. Incubate for 5 mins at RT.

13. Pellet the beads on the magnet until the eluate is clear and colourless.

14. Pipette 20 µL of the eluate into a clean 1.5 mL Eppendorf DNA LoBind tube.

15. Measure cDNA and RNA on a Qubit or similar.

3.2.3. RMX adapter ligation and cleanup

1. In a clean 1.5 mL Eppendorf DNA LoBind tube, mix the reagents in the following order:

Reagents	**Volume (µL)**
Reverse-transcribed RNA from the “Reverse Transcription” step	20
5X NEBNext Quick Ligation Reaction Buffer	8
RNA Adaptor (RMX)	6
Nuclease-free water	3
Concentrated T4 DNA ligase	3

2. Mix by pipetting and incubate for 10 min at RT.

3. Re-suspend the stock of Agencourt RNA Clean XP beads by vortexing and add 40 μL of beads to the adaptor ligation reaction and mix by pipetting.

4. Incubate on a Hula mixer for 5 min at RT.

5. Spin down the sample and pellet on a magnet. Keep the tube on a magnet and pipette of the supernatant.

6. Add 150 μL of the Wash Buffer (WSB) provided in the Direct RNA sequencing kit to the beads. Resuspend the beads by flicking the tube. Return the tube on the magnetic rack, allow beads to pellet and pipette of the supernatant. Repeat. Allow to air dry for 2 min.

7. Remove the tube from the magnetic rack and resuspend the pellet in 21 μl Elution Buffer. Incubate for 10 min at RT.

8. Pellet the beads on magnet until eluate is clear and colourless.

9. Remove and retain 21 μL of eluate into a clear 1.5 mL Eppendorf DNA LoBind tube.

10. Measure cDNA and RNA on Qubit or similar.

11. Mix the library with 17.5 μl water.

12. Add 37.5 μL RRB Buffer (Mix RRB by vortexing before using) and mix well. Library is now ready to be loaded to the flowcell (see **Note 4**).

3.3. Analysis of direct RNA sequencing datasets: base-calling and mapping

3.3.1. Base-calling

Base-calling should be performed using Oxford Nanopore Technologies' *guppy* base-caller, such as in the example shown below (see **Note 5**):

```
guppy_basecaller --device cuda:0 -c rna_r9.4.1_70bps_hac.cfg --
compress_fastq -i path/to/fast5_directory -r -s
/path/to/save/basecalling_out --fast5_out
```

3.3.2. Mapping

Map the base-called reads to the reference fasta sequences using *minimap2* [22], and keep only the mapped reads:

```
minimap2 --MD -t 6 -ax map-ont ref.fasta sample.reads.fastq | samtools view -hbS -F 3844 - | samtools sort -@ 6 - sample.reads
```

Alternatively, reads can also be aligned to a reference genome using the following command (see **Note 6**):

```
minimap2 --MD -t 6 -ax splice -k14 -uf  ref.fasta sample.reads.fastq | samtools view -hbS -F 3844 - | samtools sort -@ 6 - sample.reads
```

3.4 Extraction of features to detect RNA modifications in direct RNA sequencing datasets using *EpiNano*

The latest version of the *EpiNano* suite (version 1.2) consists of five main programs or modules:

- **`Epinano_Variants`** computes systematic base-calling 'errors' (mismatch, deletion, insertion, and per-base quality score) for each base along the mapped reads and reports their relative frequencies in a plain text file.
- **`Epinano_Current`** extracts raw current intensity values and dwell time for each reference base.
- **`Epinano_Predict`** trains models using the features extracted with the aforementioned two modules and makes predictions using trained models.
- **`Epinano_DiffErr`** predicts RNA modifications based on the differences in base-calling 'errors' between two samples (typically wild type and knock-out).
- **`Epinano_Plot`** produces scatterplots or barplots depicting the differences in base-calling 'errors' or modification probabilities between two samples, highlighting

positions that are identified by the algorithm as significantly altered, i.e., predicted as differentially modified.

EpiNano 1.2 can predict RNA modifications from direct RNA sequencing datasets using two distinct strategies: (i) ***EpiNano-SVM***, which employs pre-trained SVM models to predict RNA modifications, and (ii) ***EpiNano-Error***, which uses the differences between base-calling 'errors' (mismatches, deletions, insertions) between two samples, as well as alterations in per-base qualities, to predict RNA modifications (**Figure 2**). Both strategies rely on the fact that RNA modifications appear as systematic base-calling 'errors' in direct RNA sequencing datasets.

[Figure 2 near here]

3.4.1. Extraction of base-calling 'error' features using *EpiNano*

1. Clone the *EpiNano* repository from GitHub (https://github.com/enovoa/EpiNano) using *git*:

```
git clone https://github.com/enovoa/EpiNano.git
```

2. Extract base-calling errors using the module `Epinano_Variants`. This module relies on *sam2tsv* from the *jvarkit* toolkit (https://github.com/lindenb/jvarkit) to extract base qualities and compute variant frequencies from direct RNA sequencing data. The user must provide as input both a BAM file containing the mapped reads as well as a reference transcriptome or genome in FASTA format. The `$EPINANO_HOME` variable corresponds to the location of the *EpiNano* scripts folder.

```
python $EPINANO_HOME/Epinano_Variants.py -t 6 -R reference.fasta -b
sample.reads.bam -s /path/to/sam2tsv/sam2tsv.jar --type g
```

The '`--type`' flag indicates the type of reference that was used to obtain the bam file. If the reads were mapped to a genome reference with splicing-aware mapping options, '`--type g`' should be specified, and *EpiNano* will discriminate the reads mapped to the forward strand from those mapped to the reverse strand. Otherwise, by default, the script assumes the bam file was generated by mapping the reads to reference transcriptome and that the reads should only be mapped to the forward strand.

`Epinano_Variants` outputs two feature tables: (i) `sample.per.site.var.csv`, which contains base-calling 'error' information for each reference position, and (ii) `sample.per_site.5mer.csv`, which contains the same base-called features organized in slided 5-mer windows (see **Note 7**).

3.4.2. Extraction of current intensity values using *EpiNano*

The latest version of *EpiNano* (v.1.2) relies on the use of *Nanopolish* [23] to extract the current signal level information and collapse it on a single-position basis. We offer a custom bash script, which carries out *Nanopolish*'s *eventalign* function and further collapses the current intensity and dwell time values. `Epinano_Variants`, this will produce a file of per-position results consisting of raw current intensity values and their corresponding mean, median and standard deviations as well as a second file with results organized in 5-mer-windows.

```
sh $EPINANO_HOME/misc/Epinano_Current.sh -b sample.reads.bam -r
sample.reads.fastq -f reference.fasta  -t 6  -m g -d fast5_folder/
```

Finally, we can merge both variants and current features using the following script:

```
python $EPINANO_HOME/misc/Join_variants_currents.py
      --variants sample.per_site_var.5mer.csv
      --intensity sample.per_site_current.5mer.csv
      --outfile sample.5mer.all_features.csv
```

3.5. Predicting RNA modifications *in vivo* using trained SVM models (EpiNano-SVM)

3.5.1. Train *EpiNano* models

1. Label and merge the 'modified' and 'unmodified' datasets. To train a model, we first have to label the files containing *EpiNano*-extracted features that will be used for training the model ('`mod.per_site.5mer.csv`' and '`unm.per_site.5mer.csv`') by adding the corresponding labels ('`mod`' and '`unm`'), as shown below:

```
bash $EPINANO_HOME/misc/Epinano_LabelSamples.sh -m mod.per_site.5mer.csv -u
unm.per_site.5mer.csv -o combined.per_site_raw_feature.5mer.csv
```

2. Train the model using *EpiNano.* `Epinano_Predict` is the module to train *EpiNano* models using features that have been previously extracted using either `Epinano_Variants` and/or `Epinano_Current`, and is executed as shown below:

```
python $EPINANO_HOME/Epinano_Predict.py
          --train combined.per_site_raw_feature.5mer.csv
          --predict combined.per_site_raw_feature.5mer.csv
          --accuracy_estimation --out_prefix train_and_test
          --columns 8,13,23 --modification_status_column 26
```

While the user can choose to train the algorithm with one sample (`--train`) and test it on an independent sample (`--predict`), it is also possible to use the same input file both for training and testing the model, as depicted in the example above. In this scenario, `Epinano_Predict` will train the models with 50 % of the input data and make predictions

with the remaining 50 % of the data.

In the above command, '`--columns`' denotes the column numbers of features that are used for training models (in this case, corresponding to 'q3', 'mis3' and 'del3'), while '`--modification_status_column`' indicates the prior knowledge of the modification statuses, i.e., the labels '`mod`' and '`unm`'. Switching on `--accuracy_estimation` will report the accuracy of the trained model(s). Unless '`--kernel`' is used, `Epinano_Predict` will train models with multiple kernels. Finally, the user can visualize the accuracy of their trained models in the form of Receiver Operating Characteristic (ROC) curves (**Fig. 2**).

[Figure 3 near here]

3.5.2. Predict RNA modifications using trained SVM models

`Epinano_Predict.py` can predict RNA modifications on a given dataset using previously trained *EpiNano* models (specified with '`--model`'). In the example below, we employ a previously trained model '`q3.mis3.del3.MODEL.linear.model.dump`' that will predict m^6A modifications in RRACH k-mers on a dataset that is specified with '`--predict`' (see **Note 8**). This SVM model has been trained on RRm^6ACH and RRACH k-mers produced using *in vitro* transcription, using the steps described in section 3.4, and the features used to train the model correspond to q3, mis3 and del3, which correspond to the per-base quality, mismatch frequency and deletion frequency of the middle position of the k-mer. It is important to note that a given model should only be used to predict modifications on the same set of k-mers that were used to train the model, i.e. if the model is trained on GGACA k-mers, it should only be used to predict m^6A modifications on GGACA k-mers (see **Note 9**).

```
python $EPINANO_HOME/Epinano_Predict.py
      --model $EPINANO_HOME/models/q3.mis3.del3.MODEL.linear.model.dump
```

```
--predict sample.per_site.5mer.csv
--columns 8,13,23
--out_prefix sample_mod_prediction
```

EpiNano relies on systematic base-calling 'errors' caused by the presence of RNA modifications, and as such, it can be confounded by base-calling 'errors' that are present in the data, leading to a high number of false positives (see **Note 10**). To remove the false positives, we recommend coupling the sequencing run of interest to a knockout or knockdown condition. For example, if the user is sequencing the RNA of a given cell line, they should also sequence the matched METTL3 knockdown condition to remove false positive predictions. In addition, if *EpiNano* is run in 'paired' mode, if can be used with pre-trained SVM models that rely on the differences in the base-calling 'errors' observed between two samples (e.g. WT-KO) to predict the RNA modifications (**Fig. 4**).

Here we tested the performance of *EpiNano* 1.2 in *S. cerevisiae* wild type and ime4Δ direct RNA sequencing datasets, which are publicly available (see **Note 11**). The performance of *EpiNano* on known m^{6}A-modified sites from *in vivo* yeast mRNAs is depicted in **Fig. 4,** using two different pre-trained models, which are available in the *EpiNano* GitHub repository. We find that SVM models trained on base-calling 'error' differences perform slightly better than those relying on absolute base-calling feature values.

[Figure 4 near here]

3.6. Predicting RNA modifications *in vivo* from base-calling 'error' differences (EpiNano-Error)

In section 3.5 we have showcased the use of base-calling 'error' features to train Support Vector Machine (SVM) models to predict m^{6}A RNA modifications, using the

`Epinano_Predict` module, which was the original strategy employed by *EpiNano* to predict RNA modifications [21]. In the *EpiNano* 1.2 suite, we now provide a new module, `Epinano_DiffErr`, that can predict RNA modifications by identifying those positions that show differential base-calling 'errors' - previously extracted using the `Epinano_Variants` - when comparing two samples (e.g. a wild type and knock-out condition).

```
Rscript $EPINANO_HOME/Epinano_DiffErr.R -k ko.per.site.var.csv -w
wt.per.site.var.csv -d 0.1 -t 3 -p -o diffErr -f mis
```

In the example above, `Epinano_DiffErr` predicts RNA modifications using mismatch frequency differences (`-f mis`). We should note that distinct RNA modification types affect distinct base-calling 'error' features, and therefore, the user should choose whichever feature is most affected by the RNA modification type that is being studied. `Epinano_DiffErr` also offers the possibility of using the combination of all base-calling 'errors' simultaneously (`-f sum_err`).

To identify RNA-modified sites, `Epinano_DiffErr` relies on two metrics. The first one is based on the z-score deviance of error frequencies between two samples. The second relies on fitting a linear regression model between the features of the two samples, and modifications are then determined by identifying data points with significant residuals (inferred from z-scores or as Bonferroni corrected p-values through t-test of the studentized residuals). The thresholds for these two metrics can be adjusted by the user using parameters `-t` and `-d`, respectively.

While `Epinano_DiffErr` can only identify RNA-modified sites that are changing between the two conditions/samples studied (i.e. it cannot predict RNA modifications '*de novo*'), it

can be applied, in principle, to any RNA modification type – as long as the RNA modification type affects the current intensity and/or base-called features-.

Finally, the user can choose to visualize the predicted RNA-modified sites using the `Epinano_Plot` module:

```
Rscript $EPINANO_HOME/Epinano_Plot.R diffErr.delta-mis.prediction.csv
```

This module takes as input a comma-separated file with predictions, such as the one generated by `Epinano_DiffErr` in the previous step, and will highlight the predicted sites in scatterplots or barplots (**Fig. 5**).

[Figure 5 near here]

4. Notes

1. DNA volume should never be more than 25 % of the total volume. Enzyme volume should never be more than 10 % of total volume.

2. If the reaction is prepared at a colder temperature, a cloudy solution will appear, which indicates the precipitation of spermidine and DTT.

3. A Bioanalyzer can also be used instead of a Tape Station.

4. Keep the library on ice if it is not immediately loaded.

5. New versions of *guppy* base-caller are released every few months. The *guppy* base-caller is available from the Nanopore community (https://community.nanoporetech.com/). The current code and downstream examples used in this chapter correspond to *Guppy* version 3.1.5.

6. We recommend users to try different aligners and alignment parameters and find an optimized approach to read mapping.

7. If the reference FASTA used is large (i.e. not synthetic sequences, such as is the case of the 'curlcakes'), we recommend splitting the dataset into smaller subsets. This will greatly reduce the computation time and required memory.

8. This model, which has been trained on RRACH k-mers base-called with Guppy 3.1.5, is available in the *EpiNano* GitHub repository (https://github.com/enovoa/EpiNano).

9. There is a strong dependency between base-calling 'errors' and sequence context. Thus, if the user chooses to train on diverse k-mers simultaneously, we recommend minimizing the diversity of the k-mers included in the training.

10. False positives can also be caused by low coverage. RNA modifications should be predicted on sites with high coverage. We recommend a minimum coverage of 20-30 reads for a k-mer to be included in the analysis.

11. FAST5 data used in this work are the same from [21] and can be obtained from SRA database through the accession code SRP174366. Intermediate datasets used for this Chapter can be found at https://github.com/novoalab/EpinanoBookChapter.

12. Raw features across the 3 replicates were merged as previously described [21] using the following pseudo-code:

```
if (probM1 ≥ 0.5 and ProbM2 ≥ 0.5 and probM3 ≥ 0.5):
  probM = 1
else:
  probM = (probM1 + probM2 + probM3) / 3
 if (probM_wt/probM_ko > 1.5):
    prediction = 'mod'
 else:
    prediction = 'unm'
```

Acknowledgements

We thank all members of the Novoa lab for their valuable insights and discussion. We thank Rebeca Medina for obtaining the TapeStation image used for Fig. 1. OB is supported by an international PhD fellowship (UIPA) from the University of New South Wales. This work was supported by the Australian Research Council (DP180103571 to EMN) and the Spanish Ministry of Economy, Industry and Competitiveness (MEIC) (PGC2018-098152-A-100 to EMN). We acknowledge the support of the MEIC to the EMBL partnership, Centro de Excelencia Severo Ochoa and CERCA Programme / Generalitat de Catalunya.

References

1. Cohn WE, Volkin E (1951) Nucleoside-5′-phosphates from ribonucleic acid. Nature 167:483–484

2. Adams JM, Cory S (1975) Modified nucleosides and bizarre 5’-termini in mouse myeloma mRNA. Nature 255:28–33

3. Desrosiers R, Friderici K, Rottman F (1974) Identification of methylated nucleosides in messenger RNA from Novikoff hepatoma cells. Proc Natl Acad Sci U S A 71:3971–3975

4. Dubin DT, Taylor RH (1975) The methylation state of poly A-containing messenger RNA from cultured hamster cells. Nucleic Acids Res 2:1653–1668

5. Perry RP, Kelley DE, Friderici K, Rottman F (1975) The methylated constituents of L cell messenger RNA: evidence for an unusual cluster at the 5’ terminus. Cell 4:387–394

6. Jia G, Fu Y, Zhao X, Dai Q, Zheng G, Yang Y, Yi C, Lindahl T, Pan T, Yang Y-G, He C (2011) N6-methyladenosine in nuclear RNA is a major substrate of the obesity-associated FTO. Nat Chem Biol 7:885–887

7. Zheng G, Dahl JA, Niu Y, Fedorcsak P, Huang C-M, Li CJ, Vågbø CB, Shi Y, Wang W-L, Song S-H, Lu Z, Bosmans RPG, Dai Q, Hao Y-J, Yang X, Zhao W-M, Tong W-M, Wang X-J, Bogdan F, Furu K, Fu Y, Jia G, Zhao X, Liu J, Krokan HE, Klungland A, Yang Y-G, He C (2013) ALKBH5 is a mammalian RNA demethylase that impacts RNA metabolism and mouse fertility. Mol Cell 49:18–29

8. Dominissini D, Moshitch-Moshkovitz S, Schwartz S, Salmon-Divon M, Ungar L, Osenberg S, Cesarkas K, Jacob-Hirsch J, Amariglio N, Kupiec M, Sorek R, Rechavi G (2012) Topology of the human and mouse m6A RNA methylomes revealed by m6A-seq. Nature 485:201–206

9. Meyer KD, Saletore Y, Zumbo P, Elemento O, Mason CE, Jaffrey SR (2012) Comprehensive Analysis of mRNA Methylation Reveals Enrichment in 3′ UTRs and near Stop Codons. Cell 149:1635–1646

10. Hu Y, Ouyang Z, Sui X, Qi M, Li M, He Y, Cao Y, Cao Q, Lu Q, Zhou S, Liu L, Liu L, Shen B, Shu W, Huo R (2020) Oocyte competence is maintained by m6A methyltransferase KIAA1429-mediated RNA metabolism during mouse follicular development. Cell Death Differ. https://doi.org/10.1038/s41418-020-0516-1

11. Lee H, Bao S, Qian Y, Geula S, Leslie J, Zhang C, Hanna JH, Ding L (2019) Stage-specific requirement for Mettl3-dependent m6A mRNA methylation during haematopoietic stem cell differentiation. Nat Cell Biol 21:700–709

12. Zhao BS, He C (2015) Fate by RNA methylation: m6A steers stem cell pluripotency. Genome Biol 16:43

13. Geula S, Moshitch-Moshkovitz S, Dominissini D, Mansour AA, Kol N, Salmon-Divon M, Hershkovitz V, Peer E, Mor N, Manor YS, Ben-Haim MS, Eyal E, Yunger S, Pinto

Y, Jaitin DA, Viukov S, Rais Y, Krupalnik V, Chomsky E, Zerbib M, Maza I, Rechavi Y, Massarwa R, Hanna S, Amit I, Levanon EY, Amariglio N, Stern-Ginossar N, Novershtern N, Rechavi G, Hanna JH (2015) Stem cells. m6A mRNA methylation facilitates resolution of naïve pluripotency toward differentiation. Science 347:1002–1006

14. Lence T, Akhtar J, Bayer M, Schmid K, Spindler L, Ho CH, Kreim N, Andrade-Navarro MA, Poeck B, Helm M, Roignant J-Y (2016) m6A modulates neuronal functions and sex determination in Drosophila. Nature 540:242–247

15. Haussmann IU, Bodi Z, Sanchez-Moran E, Mongan NP, Archer N, Fray RG, Soller M (2016) m6A potentiates Sxl alternative pre-mRNA splicing for robust Drosophila sex determination. Nature 540:301–304

16. Helm M, Motorin Y (2017) Detecting RNA modifications in the epitranscriptome: predict and validate. Nat Rev Genet 18:275–291

17. Li X, Xiong X, Yi C (2016) Epitranscriptome sequencing technologies: decoding RNA modifications. Nat Methods 14:23–31

18. Linder B, Grozhik AV, Olarerin-George AO, Meydan C, Mason CE, Jaffrey SR (2015) Single-nucleotide-resolution mapping of m6A and m6Am throughout the transcriptome. Nat Methods 12:767–772

19. Novoa EM, Mason CE, Mattick JS (2017) Charting the unknown epitranscriptome. Nat Rev Mol Cell Biol 18:339–340

20. Jonkhout N, Tran J, Smith MA, Schonrock N, Mattick JS, Novoa EM (2017) The RNA modification landscape in human disease. RNA 23:1754–1769

21. Liu H, Begik O, Lucas MC, Ramirez JM, Mason CE, Wiener D, Schwartz S, Mattick JS, Smith MA, Novoa EM (2019) Accurate detection of m6A RNA modifications in native RNA sequences. Nature Communications 10:4079.

22. Li H (2018) Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics 34:3094–3100

23. Loman NJ, Quick J, Simpson JT (2015) A complete bacterial genome assembled de novo using only nanopore sequencing data. Nat Methods 12:733–735

24. Lorenz DA, Sathe S, Einstein JM, Yeo GW (2020) Direct RNA sequencing enables m6A detection in endogenous transcript isoforms at base-specific resolution. RNA 26:19–28

25. Parker MT, Knop K, Sherwood AV, Schurch NJ, Mackinnon K, Gould PD, Hall AJ, Barton GJ, Simpson GG (2020) Nanopore direct RNA sequencing maps the complexity of Arabidopsis mRNA processing and m6A modification. Elife 9: . https://doi.org/10.7554/eLife.49658

26. Price AM, Hayer KE, McIntyre ABR, Gokhale NS, Della Fera AN, Mason CE, Horner SM, Wilson AC, Depledge DP, Weitzman MD (2019) Direct RNA sequencing reveals m6A modifications on adenovirus RNA are necessary for efficient splicing. bioRxiv 865485

Figure captions

Fig. 1. Tapestation image of the Curlcake1 (CC1) products, *in vitro* transcribed with or without m^{6}A modifications, before and after polyA tailing. Ladder illustrates the size of distinct bands on the electronic gel. Each IVT product shows increased size upon polyA tailing.

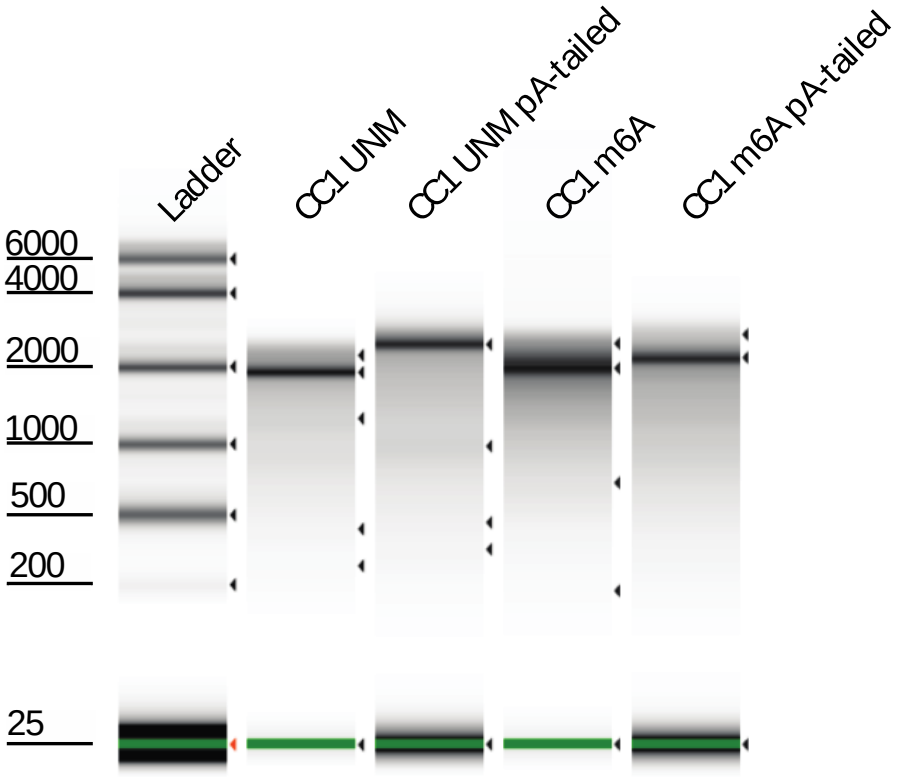

Fig. 2. Overview of the five main modules included in the *EpiNano* 1.2 suite. The latest version of *EpiNano* can predict RNA modifications using two distinct strategies: i) *EpiNano-Error*, which detects RNA modifications using differential base-calling 'errors' that are detected between two samples (typically wild type and knockout), and ii) *EpiNano-SVM*, which detects RNA modifications using Support Vector Machines (SVM), where the SVM models have been pre-trained with modified and unmodified datasets.

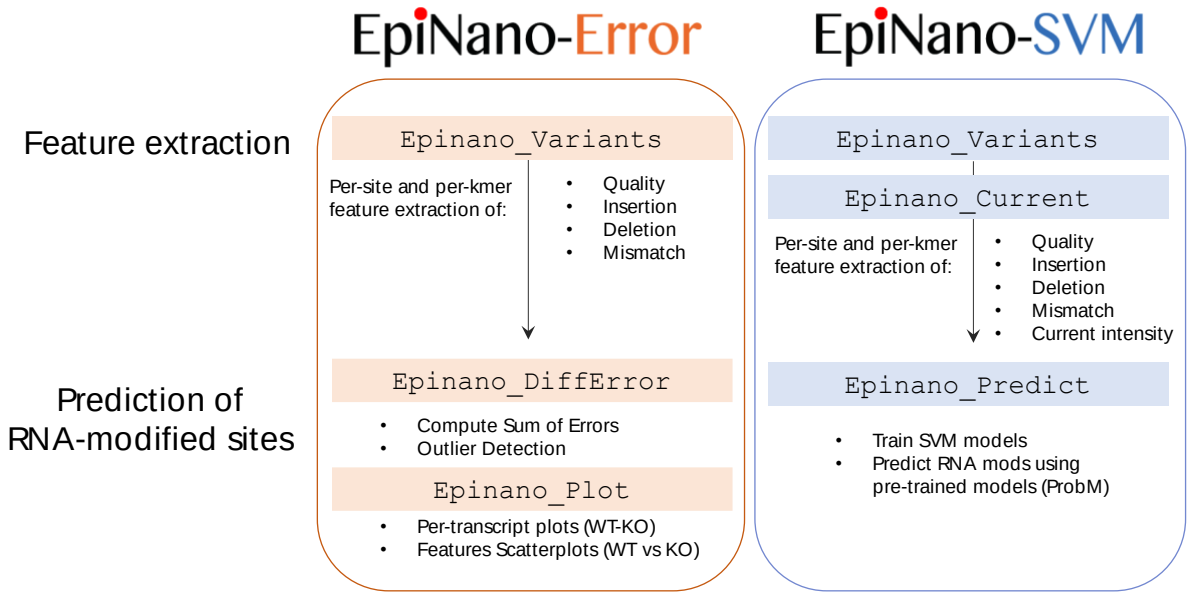

Fig. 3. ROC curves depicting the modification detection performance using *in vitro* test data (not used for training) which includes 263 RRACH k-mers. (A) All three SVM models were trained with the same subset of features (q3, mis3 and del3) described previously [21], using three different kernels (linear, poly and rbf). **(B)** The models were trained using the difference of these features (Δq3, Δmis3 and Δdel3) between the modified and unmodified positions, in a pairwise manner. AUC represents the area under the curve.

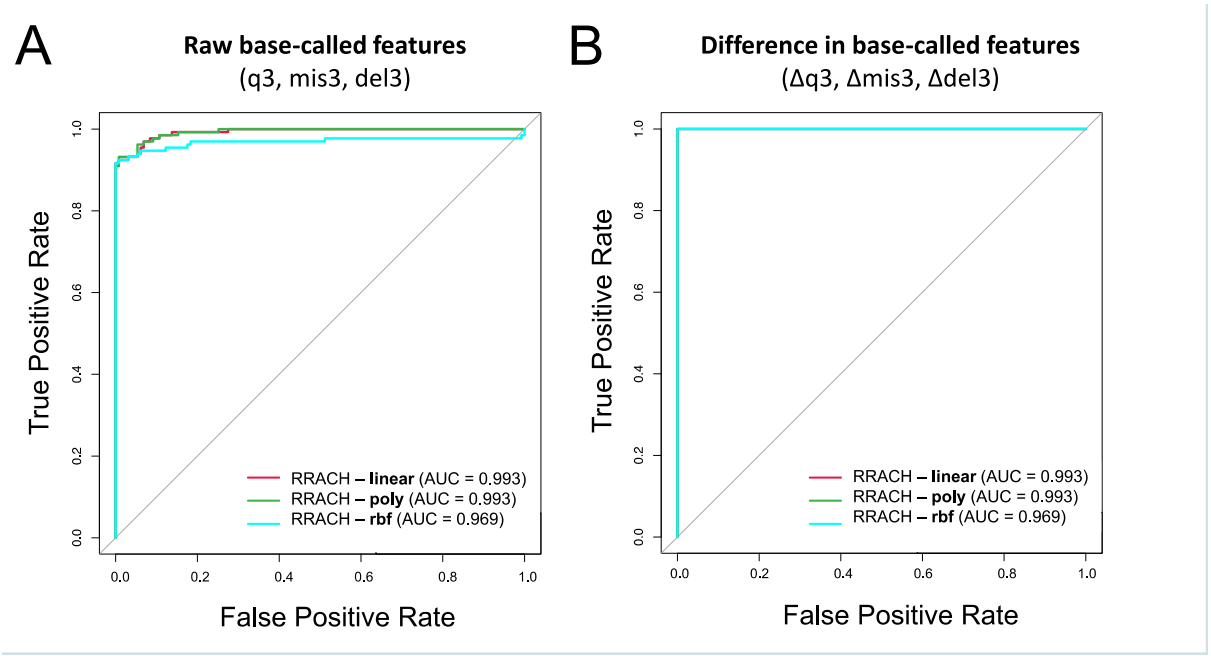

Fig. 4. ROC curve depicting the performance of m^{6}A modification detection in *in vivo* data. All shown models were trained with a linear kernel but using distinct features. Raw features across replicates were combined as previously described [21] (see **Note 12**).

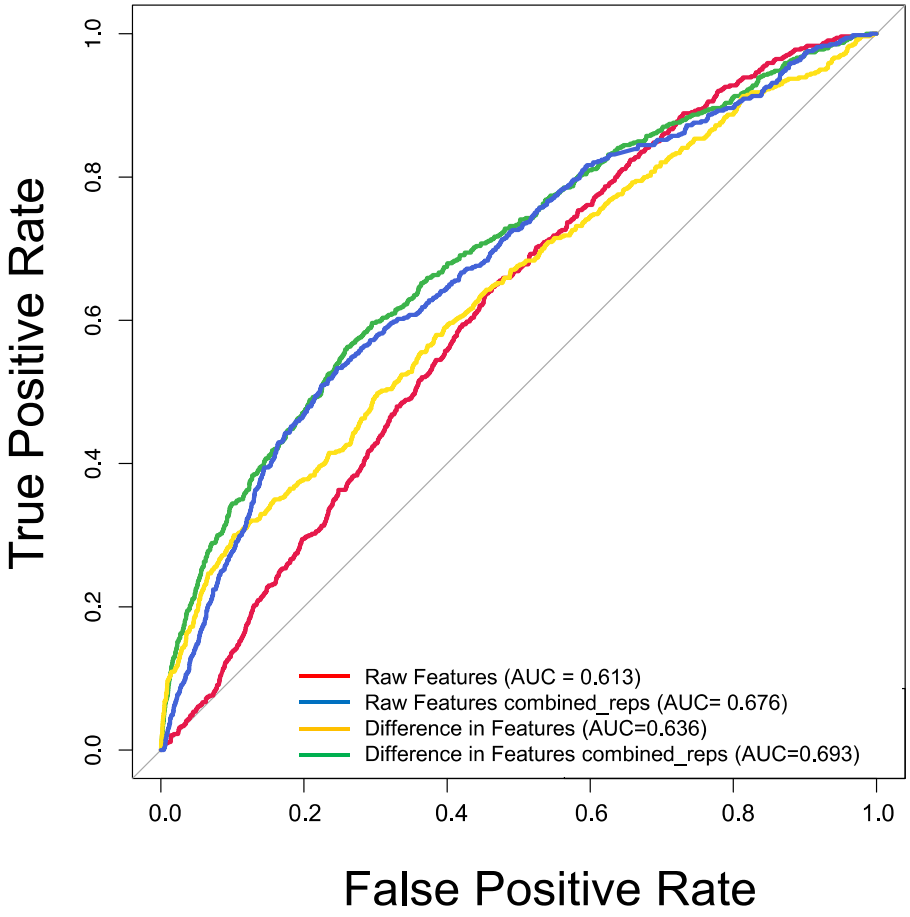

Fig 5. Scatterplots and barplots generated by Epinano_Plot. The RNA-modified sites that are known to be affected by the knock-out are positions 2826 and 2880. These plots can be generated using test data from `$EPINANO_HOME/test_data`.