Nucleic Acids Research, 2025, **53**, gkae1168
https://doi.org/10.1093/nar/gkae1168
Advance access publication date: 10 December 2024
Methods

Detecting a wide range of epitranscriptomic modifications using a nanopore-sequencing-based computational approach with 1D score-clustering

Ivan Vujaklija[1,*], Siniša Bidin[1], Marin Volarić[2], Sara Bakić[3,4], Zhe Li[3], Roger Foo[5], Jianjun Liu[3,5] and Mile Šikić[1,3,*]

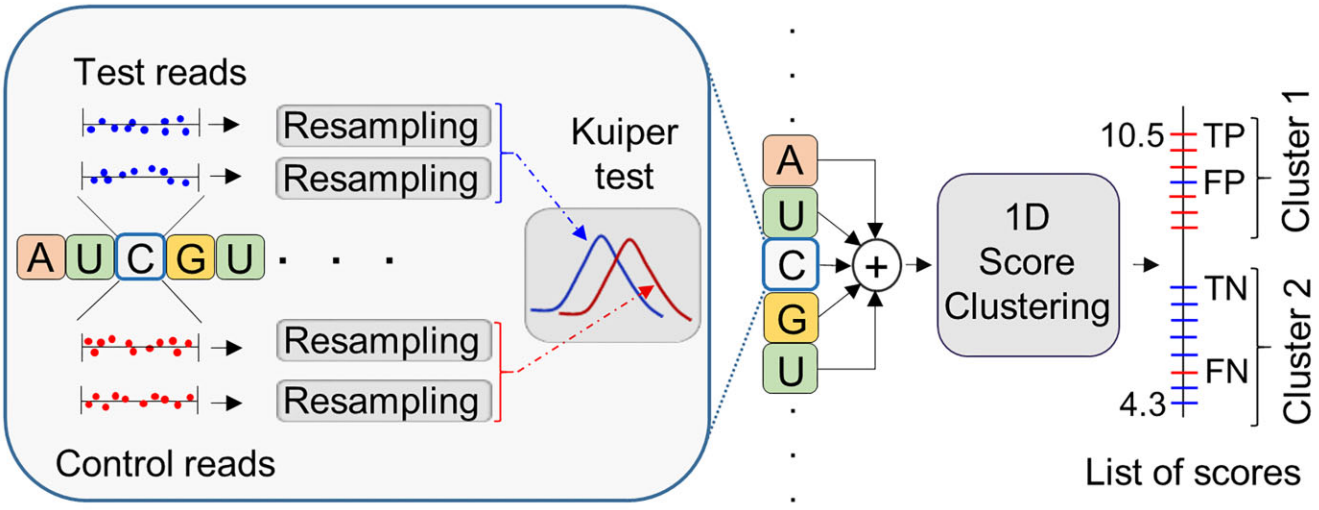
[1]Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, 10000 Zagreb, Croatia
[2]Laboratory of non-coding DNA, Division of Molecular Biology, Ruđer Bošković Institute, Bijenička cesta 54, 10000 Zagreb, Croatia
[3]Genome Institute of Singapore, Agency for Science, Technology and Research (A*STAR), 1 Create Way, Singapore 138602, Singapore
[4]School of Computing, National University of Singapore, 13 Computing Drive, Singapore 117417, Singapore
[5]Department of Medicine, Yong Loo Lin School of Medicine, National University of Singapore, 1E Kent Ridge Road, Singapore 119228, Singapore

[*]To whom correspondence should be addressed. Tel: +385 1 6129 781; Email: mile_sikic@gis.a-star.edu.sg or mile.sikic@fer.hr
Correspondence may also be addressed to Ivan Vujaklija. Tel: +385 1 6129 781; Email: ivan.vujaklija@fer.hr

Abstract

To date, over 40 epigenetic and 300 epitranscriptomic modifications have been identified. However, current short-read sequencing-based experimental methods can detect <10% of these modifications. Integrating long-read sequencing technologies with advanced computational approaches, including statistical analysis and machine learning, offers a promising new frontier to address this challenge. While supervised machine learning methods have achieved some success, their usefulness is restricted to a limited number of well-characterized modifications. Here, we introduce Modena, an innovative unsupervised learning approach utilizing long-read nanopore sequencing capable of detecting a broad range of modifications. Modena outperformed other methods in five out of six benchmark datasets, in some cases by a wide margin, while being equally competitive with the second best method on one dataset. Uniquely, Modena also demonstrates consistent accuracy on a DNA dataset, distinguishing it from other approaches. A key feature of Modena is its use of 'dynamic thresholding', an approach based on 1D score-clustering. This methodology differs substantially from the traditional statistics-based 'hard-thresholds.' We show that this approach is not limited to Modena but has broader applicability. Specifically, when combined with two existing algorithms, 'dynamic thresholding' significantly enhances their performance, resulting in up to a threefold improvement in F1-scores.

Graphical abstract



Received: January 28, 2024. **Revised:** October 30, 2024. **Editorial Decision:** November 3, 2024. **Accepted:** November 22, 2024

Introduction

Over the past few decades, it has become increasingly clear that epigenetic and epitranscriptomic modifications are crucial in regulating a wide range of functions across all forms of life. In particular, over 40 types of DNA (1,2) and >300 types of RNA modifications (3) have been reported. These modifications affect major biological processes, including those responsible for transcription, pre-messenger RNA (pre-mRNA) splicing, nuclear export, translation (4), cellular differentiation (5,6), mRNA stability (7), immune cell biology (8), neuronal development and brain plasticity (9–11), depression (12), embryonic development (13,14), viral lifecycle (15), inflammation (16), cardiovascular disease (17,18), cancer (9,19,20), aging (16,21) and others.

Many experimental methods have been developed to detect epigenetic/epitranscriptomic modifications. While short-read sequencing coupled with antibody or chemically-based detection is extensively employed, these methods face significant limitations, e.g. (22). For instance, the available repertoire of antibodies and chemicals is quite restricted, allowing the detection of <10% of presently known RNA modifications using these approaches (23). Moreover, when employing different antibodies, these methods can only identify one modification at a time and often lack quantitative precision. Additionally, the multiple ligations and extensive PCR amplification utilized in short-read sequencing methods result in the loss of information, impacting the accurate detection and quantification of DNA modifications (24).

An alternative to short-read sequencing is third generation long-read sequencing. Pacific Biosciences' Single Molecule Real-Time sequencing (SMRT) and Oxford Nanopore Technologies' nanopore sequencing are two long-read sequencing methods. Although SMRT sequencing has proven successful in detecting various epigenetic modifications (e.g. 4mC, 5mC, 5hmC, 6mA) (25), a primary drawback of this method is that many modifications lack a substantial effect on polymerase dynamics. Consequently, this diminishes the sensitivity of the SMRT method in detecting these modifications (24). Moreover, in the case of RNA modifications, SMRT has been tested only as a proof of concept (24,26).

Nanopore sequencing, on the other hand, has the potential to detect (in principle) all RNA/DNA modifications, which makes it a promising alternative to short-read sequencing. Namely, chemical modifications can change the current intensity signals and/or dwell time compared to unmodified nucleotides and downstream computational methods can detect these changes. Indeed, since the original proof of concept study (27), it has been shown that different DNA and RNA modification types can be detected in this way.

Existing computational methods, however, are focused on detecting a few well-characterized and abundant modifications, such as 5mC or 6mA in DNA (28,29) or m^6A in RNA (30,33). This is a direct consequence of the inherent limitations of the underlying supervised machine learning algorithms. Examples include those based on Hidden Markov models, e.g. (27,31), Recurrent and/or Convolutional Neural Networks, e.g. Megalodon (ONT), (28,29,32,33), and Support Vector Machines (30). These and other supervised learning algorithms require sufficiently large and high-quality training datasets for each modification type (34).

Given the vast number of potential modifications and the aforementioned limitations of experimental methods, creating ample high-quality training datasets necessary for supervised learning algorithms is extremely challenging (35). Consequently, supervised algorithms are limited in scope to a small number of well-characterized and abundant modifications. Moreover, even when training examples are available, models trained and tested in a specific setting (e.g. on a particular organism or the same organism exposed to specific environmental conditions) may be error-prone and unreliable in other settings (e.g. the same organism exposed to different environmental conditions). This is a well-known problem of overfitting, which arises when training datasets are not representative of the deployment conditions (34). The complexity of biological systems and the vast repertoire of their responses to changes in environmental conditions creates an enormous variability of biological settings, which consequently represents another considerable challenge for supervised learning algorithms (34). Moreover, due to their computational complexity, supervised algorithms are currently constrained to simultaneously detect only one or two modification types (34). Finally, supervised algorithms are even, in principle, unable to detect novel or rare modifications since they require training examples. Therefore, the simultaneous identification of a diverse repertoire of genomic or transcriptomic modifications (including rare or novel types) remains an elusive goal for supervised learning algorithms.

Unsupervised learning algorithms, on the other hand, can address all of these shortcomings, and the necessity for developing such methods has been recently recognized (2). The trade-off is that unsupervised algorithms detect only the modification site and not the modification type. Crucially, this limitation can be addressed since it is well-known that most types of modifications occur within specific sequence or structural contexts (36). This fact can be leveraged to determine the modification type. For example, a 'hit' within the DRACH motif (where D = A/G/T, R = A/G, H = A/C/T) strongly indicates m^6A. In summary, the main advantage of the unsupervised approach lies in its ability to identify a wide range (in principle all) of modifications simultaneously without the need for large and complex training datasets for each modification type or biological setting. Additionally, since unsupervised methods do not require training datasets, they resist overfitting.

Two types of unsupervised methods/algorithms in epigenetic/epitranscriptomic settings are signal-based and basecaller-error-based (23,37). Both align negative/control reads and native/test reads to a reference sequence. In the signal-based approaches, the process involves aggregating raw signal samples from different reads position-wise, separately for native/test and negative/control reads. This aggregation results in two distinct distributions, and the distance between these distributions is quantified by using statistical tests such as the well-known two-sample Kolmogorov–Smirnov test (KS-test for short, 38). Basecaller-error-based methods, on the other hand, leverage the fact that modified positions frequently show a higher incidence of 'basecaller-errors' (e.g. mismatches). Thus, by comparing 'basecaller-error profiles' between negative/control and native/test samples at specific sites, the presence of modified nucleotides can be inferred.

This study presents a signal-based unsupervised algorithm with distinctive computational novelties that set it apart from similar algorithms. We constructed twelve new benchmark datasets with unique advantages and demonstrated that our

algorithm, Modena, significantly outperforms different types of unsupervised algorithms. We also show that Modena's 1D score-clustering approach, when integrated with other algorithms, can substantially improve their performance, which could be of broader interest.

Materials and methods

Modena algorithm

Like other unsupervised algorithms used in the epigenetic/epitranscriptomic setting, Modena (Figure 1; novel steps are shown in grey) requires two sets of data: a negative/control set containing reads where modifications have been completely or partially removed (e.g. by *in vitro* transcription, gene knockout, gene knockdown, inhibitor treatment) and a second set containing native/test reads. Both sets of reads have to be basecalled and resquiggled. Here we used Guppy (v. 6.1.7) for basecalling and Tombo (v. 1.5) for resquiggling. Note that basecalling and resquiggling are just pre-processing steps and not a key feature of our algorithm. Modena application (https://github.com/sbidin/modena) supports both Tombo and f5c for resquiggling. Resquiggling bins signal samples into discrete events and assigns them to positions on the reference sequence (Figure 1A, Step 1). As illustrated in Figure 1A (Step 1), events of different reads aligned to the same reference position often differ in the number of assigned signal samples. To ensure consistency, we resample each event (sampling with replacement) to contain the same number of signal samples. In Figure 1A, for clarity, only 10 signal samples are assigned to each event after the resampling step (Step 2). In reality, each event contains 30 signal samples after Step 2. This number is a hyperparameter and was determined on the validation dataset (Supplementary Figure S1). Note that hyperparameters (e.g. number of clusters in k-means clustering or sample size in this setting) are user-defined variables which affect algorithm outputs but do not change its underlying principle/mathematical formulation.

Next, all signal samples from negative/control and native/test reads assigned to the same position are combined into two distributions. In Step 3 (Figure 1A), the distance statistic between these two distributions is computed position-wise using the Kuiper test (39). This computation is performed independently for each position (Figure 1A, Step 3). In Step 4 (Figure 1B), these distance statistics/'scores' from five neighbouring positions (central position and adjacent positions −2, −1, +1 and +2) are summed to produce a single 'distance-sum score'. This is an important step since many studies have shown that modifications affect signals of neighbouring upstream and downstream positions (32,37,40–42). Note that this procedure (Step 4) was reported before (43). The 'window-size' of five was chosen both based on literature (43), and for its theoretic appeal. We also tested other 'window-sizes' (three and seven) on the DNA validation set, but a 'window-size' of five showed the best performance. Following Step 4, each position is assigned a specific 'distance-sum score', and all positions are ranked according to this score. Finally, a 1D clustering algorithm partitions the 'distance-sum scores' into two clusters (Figure 1B, Step 5). Positions assigned to the 'higher-scores cluster' are classified as positive, while those assigned to the 'lower-scores cluster' are classified as negative.

Novel steps in Modena

Resampling (Step 2)

The resampling step (Figure 1A, Step 2) includes sampling with the replacement of signal samples assigned to each event at all positions. Each event contains 30 signal samples after this step. This number is a hyperparameter and was determined on the validation dataset (Supplementary Figure S1).

Kuiper test (Step 3)

Given a sample realization of size n, i.e. an n-tuple $(X_1 = x_1 \dots X_n = x_n)$, where $X_i \sim F$ are independent and identically distributed random variables, the empirical distribution function is a stepwise function defined by

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} 1\,(x_i \le x) \tag{1}$$

where $1(x_i \le x)$ is an indicator function which equals 1 if $x_i \le x$, and is zero otherwise. Given two empirical distribution functions F_n and G_m, with sample sizes n and m, respectively, the two-sample Kolmogorov–Smirnov test (38) statistic is defined by

$$KS = \sup_x \left|F_n(x) - G_m(x)\right| \tag{2}$$

Kuiper test (39) is related to the Kolmogorov–Smirnov test, and its statistic is defined by

$$K = \sup_x \left\{F_n(x) - G_m(x)\right\} + \sup_x \left\{G_m(x) - F_n(x)\right\} \tag{3}$$

In this study, we used the Astropy implementation of the Kuiper test and the Scipy implementation of the Kolmogorov–Smirnov test.

1D score-clustering (Step 5)

The 1D clustering problem (sometimes called univariate clustering) was first described and solved in (44). Given M-ordered 'objects' (represented by numbers 1 to M), the goal is to find an optimal partition of these 'objects' into K 'homogenous groups' (note that we use terms groups and clusters as synonyms). Each 'object' $n \in \{1 \dots M\}$ has an associated value $v_n \in \mathbb{R}$ and weight $w_n \in \mathbb{R}$. A partition P into K-groups (i.e. clusters, where $K \le M$) can be defined by a K-tuple $(x_1, \dots x_K)$ where each $x_i \in \{1, \dots M\}$ represents the first element of group i, where $i \in \{1, \dots K\}$ and $x_i < x_{i+1}$. Since the first element of the first group is always one, $x_1 = 1$. Likewise, x_K equals the first element of the K-th group. Homogeneity of a partition P, $H(P)$ is the sum of homogeneities of its K constituent groups. For example, given a group $\mathcal{G}_n$ comprising objects k to l $(k < l)$, one can define homogeneity by

$$H(\mathcal{G}_n) = \sum_{i=k}^{l} w_i (v_i - \bar{v}(\mathcal{G}_n))^2 \tag{4}$$

where $\bar{v}(\mathcal{G}_n)$ is a weighted arithmetic mean of group $\mathcal{G}_n$ (i.e. its centroid):

$$\bar{v}(\mathcal{G}_n) = \sum_{i=k}^{l} (w_i \cdot v_i) \,/\, (l - k + 1)\,. \tag{5}$$

The homogeneity of the partition $H(P)$, is simply the sum of homogeneities of its constituent groups: $H(P) = \sum_{n=1}^{K} H(\mathcal{G}_n)$. Our case is a particular case of the above-described general problem setting. Namely, in our setting 'objects' are scores, where each score is formally defined by its rank i, and v_i is simply the score value (as explained previously). All weights w_i are set to 1, and number of groups/clusters is two (i.e. $K = 2$).

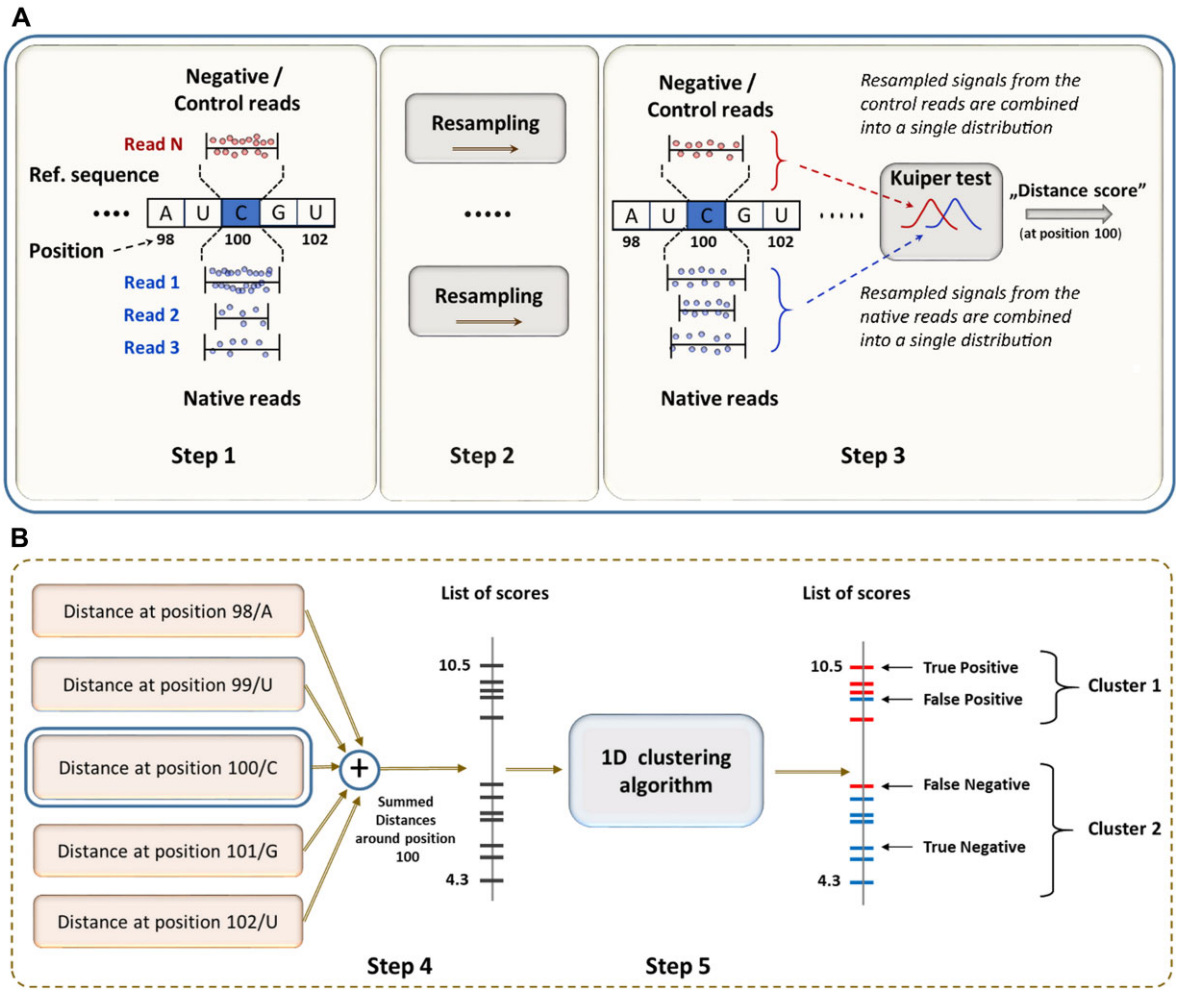


Figure 1. Modena flowchart. The algorithm comprises five steps. The input to Modena (Step 1) consists of two sets of basecalled and resquiggled reads: one from a control dataset (e.g. *in-vitro* transcribed reads) and the other from a test dataset (e.g. native reads). Step 2 involves sampling with replacement from the signals of each event at all positions. Each event contains a variable number of signal samples before Step 2 and 30 signal samples afterwards (only 10 are shown here). In Step 3, these resampled signals are combined into two distributions, and the distance statistic (or distance for short) is computed using the Kuiper test. In Step 4, the distances of five neighbouring positions are summed to produce a single score. Finally, in Step 5, scores are clustered into two groups using the 1D-clustering algorithm. Note that all boxes in Step 4 (Figure 1B, left) have the internal structure shown in Figure 1A.

Optimal partition in the 1D setting can be found by a dynamic programming procedure. Several algorithms solve the above-stated problem optimally but differ in time complexity. Here, we used 'kmeans1d' Python library, which implements an algorithm firstly described in (44) and further elaborated in (45) and has $O(kn)$ time complexity (where k equals the number of clusters). Note that the well-known k-means clustering algorithm [i.e. more precisely, Lloyd's algorithm (46)] is not guaranteed to find the optimal solution and should, therefore, be avoided in the 1D clustering setting.

Definition of positives/negatives in the test datasets

In this study, all modified positions and positions within the ±2 nucleotides distance from the modified position are defined as positive. For example, if position 100 is modified, then positions 98, 99, 100, 101 and 102 are defined as positive. We consider the reasons for using this definition to be of general importance for benchmarking algorithms in the unsupervised setting and, therefore discuss them in detail in Supplementary Discussion (S1).

Definition of correctly identified modifications

A modified position is considered correctly identified if an algorithm predicts as positive at least one position within ±2 nt distance from the truly modified position. For details, see Supplementary Discussion (S1).

Validation dataset

Previously published CpG methylation of *Escherichia coli* K12 ER2925 strain (27) was used as a validation dataset for determining the sample size after resampling. The dataset was downloaded from the European Nucleotide Archive (accession number PRJEB13021).

Constructing ribosomal RNA benchmark datasets

Recently published data (47) comprising ribosomal RNA (rRNA) of *E. coli* (16S and 23S subunits) and *S. cerevisiae* (18S

and 25S subunits) was used to generate the first benchmark dataset. Each ribosomal subunit dataset contains two subsets: one composed of native/test reads and a second consisting of negative/control reads. Negative/control reads have been generated by *in vitro* transcription, which removed all modifications. The original rRNA dataset (47) was downloaded from BioProject under accession number PRJNA634693.

Sample structure

To benchmark five selected algorithms, we generated 10 independent samples from the native/test and 10 independent samples from the negative/control datasets, each consisting of eight subsamples with approximately uniform coverage-depths of 10, 50, 75, 100, 200, 500, 1000 and 2000. Since the total number of modifications per rRNA subunit was relatively low (especially for the *E. coli* 16S subunit), we combined results (i.e. scores) from the four subunits into a single subsample test dataset. This resulted in a total of 80 test datasets (10 samples × 8 coverage-depths). To achieve approximately uniform coverage-depth (for each subsample), we used the procedure outlined by pseudocode shown below:

```
for sample in (1 to 10)
  for subunit in (Ec 16S; Ec 23S; Sc 18S; Sc 25S):
    choose multifast5 directory
    sort reads by length descending
    current_set=empty
     curent_set_size=0
    for cov_depth in (10 to 2000):
      while current_set_size<cov_depth:
        read= next longest read from sorted list
        if read_length>= 3/4 ref_seq_length
          add read to current_set
          increase current_set_size by one
        else
          discard the remaining reads
          choose another unused directory
          sort new reads by length descending
      subsample[sample,subunit,cov_depth]=current_set
```

This code is run independently on both the negative/control and positive/test datasets. As an illustration, the coverage-depths of *E. coli* 16S (*in vitro*) rRNA subunit for Sample 1 (all coverage-depths) are shown in Supplementary Figure S2.

In addition, a recently published human 18S rRNA (48) was used to construct the second benchmark dataset. To achieve the approximately uniform coverage-depth for each subsample, we used the same procedure as above (the only difference being that there is only a single subunit). The subsamples with coverage-depths of 1000 and 2000 could not be constructed due to a limited number of reads covering at least 3/4 of the subunit length. For the same reason, there were insufficient reads to construct more than one test sample.

Pseudouridylation, 2′-O-methylation and 'other modifications' datasets

We constructed three additional benchmark datasets from the *E. coli*/*S. cerevisiae* dataset. The first, 'pseudouridylation dataset', was constructed by filtering out all non-ψ modification sites (along with their ±2 nt neighbourhoods). The second, '2′-O-methylation dataset', was constructed by filtering out all non-Am/Gm/Cm/Um modification sites (along with their ±2 nt neighbourhoods). The third subset ('other-modifications dataset') was constructed by filtering out all ψ and all Am/Gm/Cm/Um modification sites (along with their ±2 nt neighbourhoods), thus keeping only non-pseudouridylation and non-2′-O-methylation modification sites.

We also constructed three benchmark datasets ('pseudouridylation dataset', '2′-O-methylation dataset' and 'other-modifications' dataset) from the human 18S rRNA dataset by using the same procedures as above.

Synthetic sequences with m^{6}A ('curlcakes') benchmark dataset

The third benchmark dataset was constructed from a previously published dataset (30), and consists of four 'curlcakes' (2329, 2543, 2678 and 2795 nt) which contain m^{6}A instead of adenine. These sequences include all possible k-mers. Due to the limited number of reads that cover at least 3/4 of the 'curlcake' length, only one sample with eight subsamples of coverage-depths ranging from 10 to 2000 was constructed using the same procedure as above. Similar to the *E. coli* and *S. cerevisiae* rRNA datasets, for a given-coverage-depth results (i.e. scores) obtained from the four 'curlcakes' were combined.

Oligos with different modification types benchmark dataset

The fourth benchmark dataset was taken from (37). This dataset consists of three short oligos (∼ 100 nt each). Each oligo contains three artificially induced modifications at known positions: Oligo 1: three m^{6}A; Oligo 2: one I, one m^{5}C, one Ψ and Oligo 3: one m$^6{}_2$A, one m^{1}G and one Am. Again, the same procedure was used to construct three independent samples, each containing eight subsamples with uniform coverage-depths ranging from 10 to 2000. Note that there were not enough reads to construct more than three independent samples. For benchmarking, all three oligos (i.e. their scores) were combined (for a given coverage-depth) into a single benchmark dataset, as was done for the rRNA and the 'curlcakes' datasets.

Stoichiometry dataset

The stoichiometry dataset was constructed from the 'oligos dataset', described above. For a given coverage-depth and modified/unmodified condition, we mixed reads from the three independent samples into a single dataset. Next, for a given coverage-depth and condition=modified, we mixed positive/negative reads in 25:75, 50:50 and 75:25 proportions, and performed sampling with replacement 10 times. In this way, for a given coverage-depth, and modified condition we obtained 10 samples, each containing three subsamples with modified to unmodified reads ratios of 25:75, 50:50 and 75:25. This was done for all coverage-depths, ranging from 20 to 2000. Note that we used a coverage-depth of 20 instead of 10 since $0.25 \times 10 = 2.5$, and at such small coverage-depths, taking either two or three modified reads makes a substantial difference.

DNA dataset

As a final dataset we used in-house generated DNA dataset with methylated cytosines in the CpG context. The *E. coli* K12 MG1655 strain was grown in 1× LB Broth Miller (1st Base, Singapore) without antibiotics at 37°C at 160 rpm shaking. The genomic DNA of *E. coli* was extracted using PureLink Genomic DNA Mini Kit (Thermo Fisher, USA). To generate the

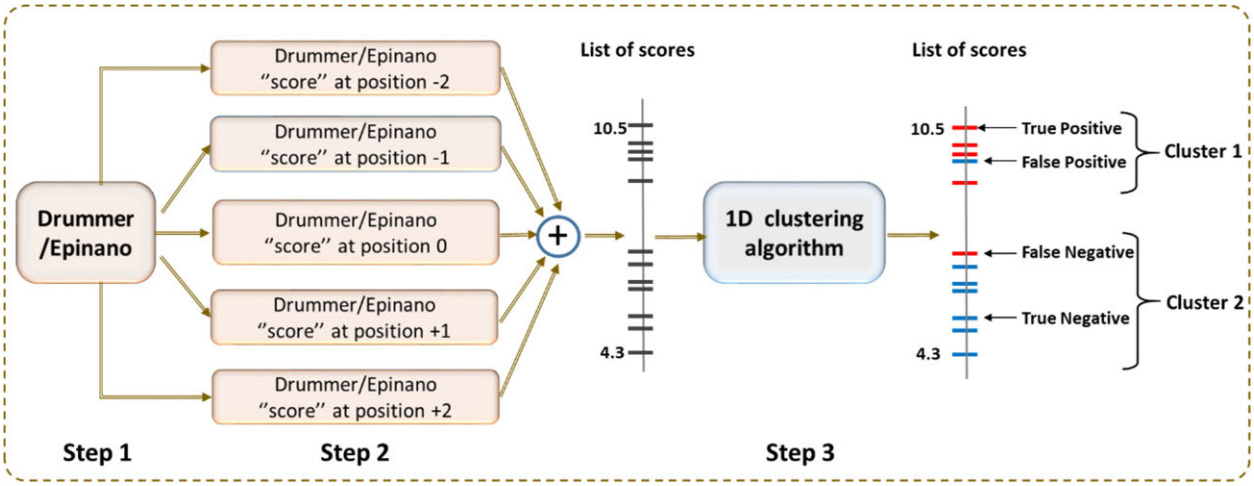


Figure 2. Flowchart of 1D score-clustering applied to Drummer and Epinano. As shown, the procedure consists of three steps. In Step 1, the standard Drummer or Epinano-DSE algorithm is run. In Step 2, the output scores (G-test statistic or z-score) from five neighbouring positions (positions −2, −1, 0, 1 and 2) are summed into a single score. The output of Step 2 is a ranked list of scores. Next, in Step 3, 1D clustering algorithm is applied to the list of ranked scores, partitioning them into two clusters. Step 3 is crucial and is equivalent to Modena's Step 5 in Figure 1B.

negative control, modifications on the native genomic DNA were wiped out using the REPLI-g Mini Kit (Qiagen, Germany). Subsequently, the positive control was produced by treating the whole genome amplified (WGA) sample (i.e. the negative control) with M.SssI methyltransferase (NEB, USA), as per the recommended conditions. Both negative and positive control reaction mixtures underwent the same purification and treatment protocols from this point onwards. They were digested with Proteinase K (Ambion, Thermo Fisher) at 50°C for 30 min and purified by phenol-chloroform extraction, following standard protocols, using UltraPure™ Phenol:Chloroform:Isoamyl Alcohol (Thermo Fisher). A single chloroform extraction was performed to eliminate residual phenol. The DNA was then precipitated using 2.5 volumes of absolute ethanol and 10% volume of 3 M sodium acetate (pH 5.5, Ambion), following standard protocols. It was subsequently washed once with 70% ethanol. The resultant pellet was dried for 5 min and slowly hydrated at 4°C in 10 mM Tris (pH 7.5) buffer for over 72 h. The resulting DNA fragments were quantified and quality-checked using Qubit, Nanodrop, and the Agilent Genomic DNA ScreenTape on a 2200 TapeStation system. The resultant DNA was then prepared for the library and sequenced on a MinION R9.4.1 (FLO-MIN106), adhering to the official library prep ligation protocol for SQK-LSK109 and standard sequencing parameters.

Algorithms used for comparison

We have selected four algorithms to benchmark our algorithm. The selection was made based on the recent reviews (23,34,49) and according to the following criteria:

1. Since many algorithms rely on the same underlying principle (e.g. basecaller-errors), our goal was to encompass various approaches. Therefore, we chose *Drummer* (50) and Epinano (51) as they are both based on basecaller-errors, albeit utilizing different statistics. Nanocompore (37) is a signal-based algorithm relying on bivariate Gaussian mixture model. Note that we used two different Epinano algorithms, which are both based on the basecaller-error approach: Epinano Delta Sum Error (Epinano-DSE for short) and Epinano Linear Regression (Epinano-LR for short).
2. Selected algorithms were published recently: Nanocompore and Epinano in 2021 and Drummer in 2022. This was done since newer algorithms generally show improved performance over the older algorithms;
3. Selected algorithms were published in peer-reviewed scientific literature, have well-defined default thresholds, and have been used in benchmark studies before.

Modifying Drummer and Epinano algorithms with the 1D score-clustering approach

Just like in the case of Modena, both Drummer and Epinano use two sets of input reads. One set is control/negative, while the other is native/test (Figure 2, novel steps are shown in grey). For each position, G-test statistic or z-score of 'error differences' between two samples are computed by Drummer and Epinano-DSE, respectively (Figure 2, Step 1). Up until this point, these are just original Drummer and Epinano algorithms. Next, in Step 2 (Figure 2), G-test statistics (Drummer) or z-scores (Epinano-DSE) of five neighbouring positions (central position flanked by two neighbouring positions) are summed into a single score (43). This step improves the ranking of all algorithms used in this study which is important as good ranking is necessary for both 'dynamic thresholding' (used by Modena) and the commonly used 'hard threshold' approach (usually determined by pre-specified *P*-value cut-off) to be applied successfully. Finally, in the crucial Step 3 (analogous to Step 5 of Modena, shown in Figure 1B), the 1D clustering algorithm partitions scores into two clusters. Positions assigned to a cluster with higher scores are classified as positive, whereas positions assigned to a cluster with lower scores are classified as negative.

Results and discussion

Modena outperformed other algorithms on the two rRNA benchmark datasets

E. coli and S. cerevisiae rRNA dataset

As a first dataset we constructed a new benchmark dataset based on the recently published data comprising rRNA of

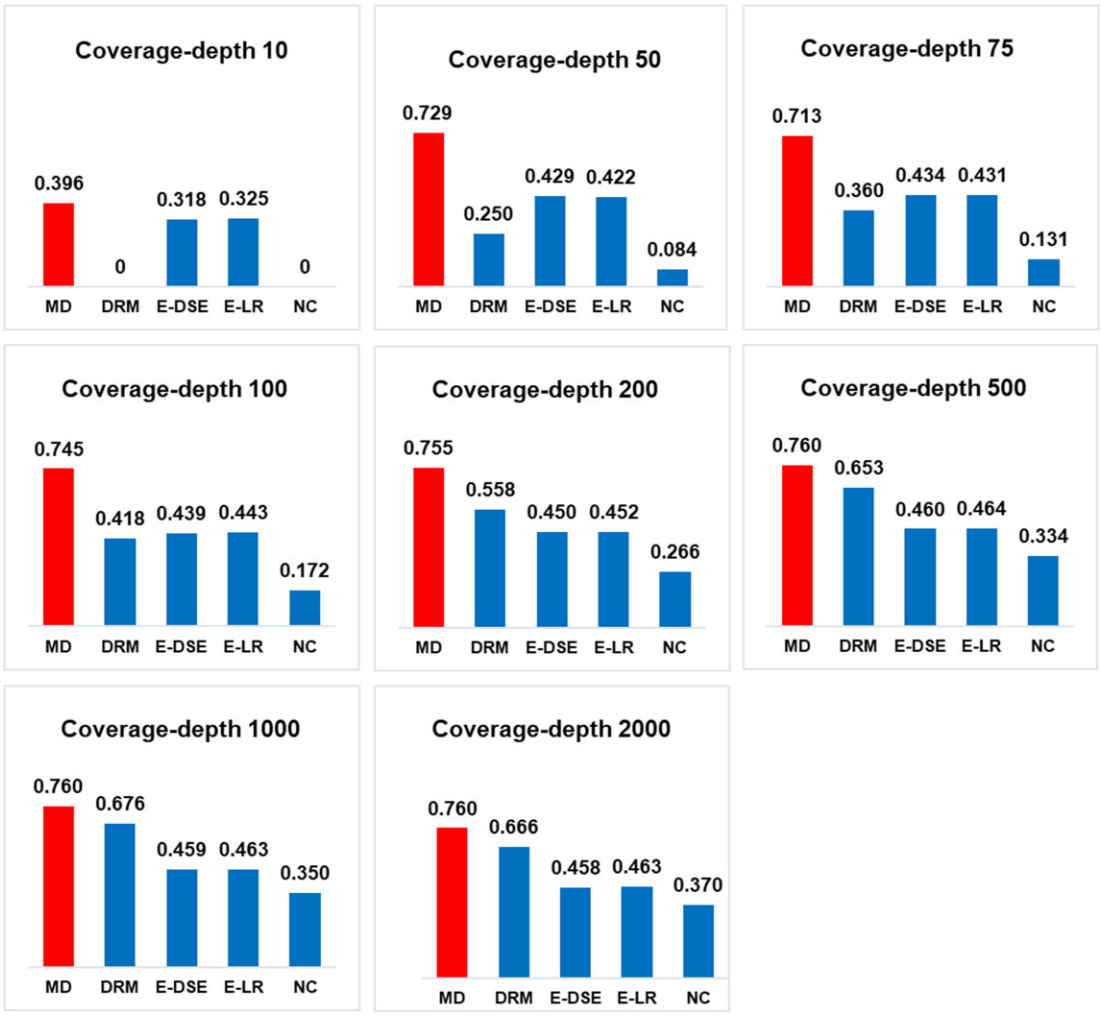


Figure 3. Average F1-scores of the five algorithms compared in this study on the *E. coli* and *S. cerevisiae* rRNA test dataset (**NC**: Nanocompore; **DRM**: Drummer; **E-DSE:** Epinano Delta-Sum-Error; **E-LR**: Epinano Linear Regression). The *E. coli* and *S. cerevisiae* rRNA datasets comprise 10 independent samples. Each sample contains eight subsamples with coverage-depths ranging from 10 to 2000. Different coverage-depths were used since algorithm performance depends on the coverage-depth, as indicated by recent studies (37,50) and also confirmed by our results. Note that all positions are treated as either positive or negative since unsupervised algorithms, do not distinguish between different modification types. In line with this, we do not compute separate F1-scores for each modification type separately, but rather only one F1-score for the whole dataset (for the given coverage-depth). As shown, Modena outperformed other algorithms across all coverage-depths; in some cases by a large margin (e.g. at coverage-depths of 50, 75, 100 and 200). The performance of all algorithms was very stable across the 10 independent samples (Supplementary Data S1). Thus, although the figure above shows average F1-scores, the results are highly consistent across all Samples 1–10.

E. coli and *S. cerevisiae* (47) (see the 'Materials and methods' section). There are several advantages to the new benchmark dataset: (i) it contains 23 types of modifications, which is more than in previously used studies; (ii) these modifications are well-characterized (high accuracy of ground-truth labels is crucial for algorithm benchmarking; see Supplementary Discussion S2) and abundant (47); (iii) native molecules are used instead of artificial constructs; and (iv) we constructed 10 independent samples, each comprising eight subsamples with varying coverage-depths of 10, 50, 75, 100, 200, 500, 1000 and 2000. This was done to facilitate comparison, as algorithm performance can be strongly affected by coverage-depth (37,50). Additionally, test datasets with non-uniform coverage-depths would significantly complicate the interpretation of results, as it would be more difficult to attribute the algorithm's good or poor performance to other factors, such as modification type. That said, all algorithms used in this study (including Modena) also work with datasets with non-uniform coverage-depths.

To benchmark Modena, four algorithms were chosen for comparison: Drummer (50), Nanocompore (37), Epinano Distance-Sum-Error (51) and Epinano Linear-Regression (51). Criteria for selecting algorithms for benchmarking Modena are described in the 'Materials and methods' section. Since the F1-scores of all algorithms across all 10 samples varied only slightly, only the average F1-scores are shown in Figure 3, while detailed results are provided in Supplementary Data S1. As shown, Modena outperformed all four algorithms across all coverage-depths; in some cases, by a large margin (e.g. at coverage-depths of 50, 75, 100 and 200).

Human 18S rRNA

Taking into account a recently published study (48) that benchmarked different algorithms on the human 18S rRNA,

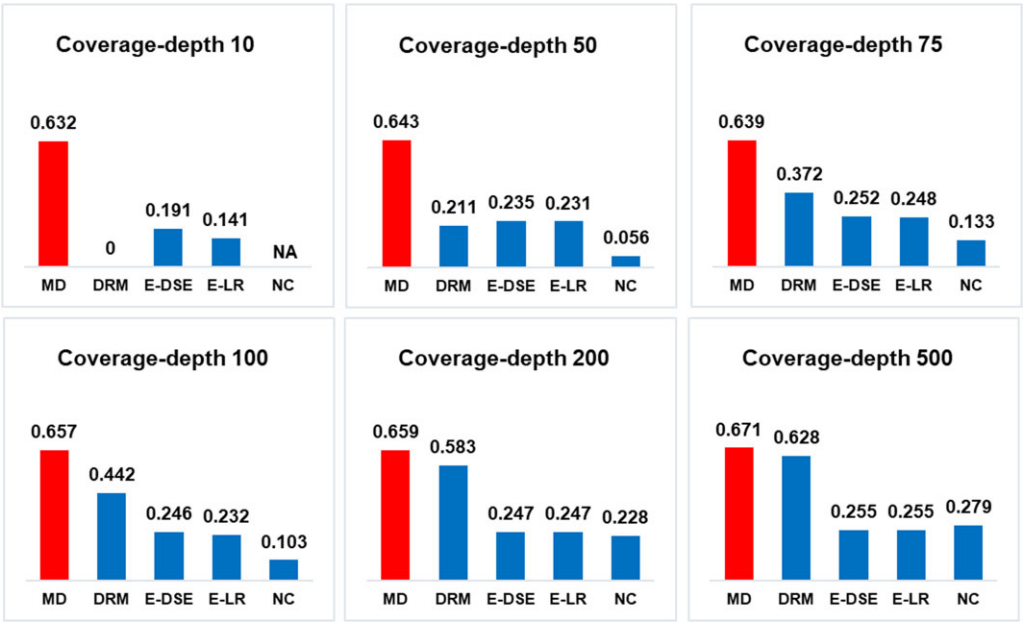


Figure 4. F1-scores of the five algorithms on the *Homo sapiens* 18S rRNA test dataset (**NC**: Nanocompore; **DRM**: Drummer; **E-DSE:** Epinano Delta-Sum-Error; **E-LR**: Epinano Linear Regression). Due to the limited number of reads of sufficient length, only one sample with subsamples with coverage-depths ranging from 10 to 500 was constructed.

as well as the availability of reliable ground-truth labels [i.e. modified positions have been mapped with a highly accurate method (52)], we decided to use human 18S rRNA dataset (48) as an additional benchmark. Moreover, although the relative abundance of particular modifications in *S. cerevisiae* (18S and 25S) and human 18S rRNA are similar, the modification stoichiometry levels in human 18S rRNA are generally quite lower. Namely, while ∼75% of modified positions in *S. cerevisiae* rRNA have stoichiometry levels above 90%, only about 58% of modified positions in human 18S rRNA do (52,53), thus making the human 18S rRNA a more challenging benchmark dataset. Unlike in the previous case with the *S. cerevisiae* and *E. coli* rRNA dataset, we could not construct subsamples with coverage-depths above 200 due to a limited number of reads covering at least 3/4 of the subunit length. Thus, we loosened this criterion to 2/3. Note that even with this lower value, we could not construct subsamples with coverage-depths of 1000 and 2000. Additionally, there were not enough reads to construct more than one sample. However, we do not see this as a serious limitation, as the variability in F1-scores (as well as Recall, Precision, Accuracy and Specificity) varied only slightly across the 10 independent samples in the *E. coli* and *S. cerevisiae* cases (Supplementary Data S1). Thus, we expect this to be the case here as well. As shown in Figure 4 and Supplementary Data S2, Modena again outperformed other algorithms across all available coverage-depths, although its F1-scores were somewhat lower. This decrease could be due to lower stoichiometry levels in human 18S rRNA mentioned earlier.

Algorithm performance depends on the modification type and sequence context

The two most prevalent modification types in both *E. coli*/*S. cerevisiae* and the human 18S rRNA datasets are 2′-O-methylation (Am/Gm/Cm/Um) and pseudouridylation (ψ). In line with this, three additional datasets were constructed from the *E. coli*/*S. cerevisiae* dataset as described in the 'Materials and methods' section: (i) 'pseudouridylation dataset' containing only ψ, (ii) '2′-O-methylation dataset' containing only Am, Gm, Cm and Um, and (iii) 'other modifications dataset' containing all other modifications. These three datasets were analysed by the five algorithms, and the results are shown in Supplementary Table S1. While Modena has the highest F1-scores on the 'pseudouridylation' and '2′-O-methylation datasets' (except for the coverage-depth of 10), no algorithm can be singled out as the best on the 'other-modifications dataset'. In particular, Modena has the highest F1-scores at coverage-depths of 50, 100 and 1000; Drummer has the highest F1-scores at coverage-depths of 200 and 500, while Epinano-LR has the highest F1-scores at coverage-depths of 10, 75 and 2000 (although difference to Modena at the coverage-depth of 2000 is negligible). As shown, F1-scores are generally lower for the 'other modifications dataset' than for the other two datasets (Supplementary Table S1). It is important to emphasize that F1-scores are generally sensitive to the relative ratio of positives to negatives. Namely, lowering the ratio of positives to negatives will, in general, lower the F1-score (54,55). While this is not a problem when comparing the performance of different algorithms on the same dataset (as done here), one should be cautious when comparing the performance of the same algorithm across different datasets. For example, the lower F1-scores of Modena, Drummer and Nanocompore on the 'other-modifications dataset', compared to the 'pseudouridylation dataset', does not necessarily indicate that these algorithms are 'worse' at recognizing 'other modifications', since their weaker performance on the 'other modifications dataset' could potentially also be due to the lower ratio of positives to negatives in that dataset.

Using the same procedure, we also generated datasets for pseudouridylation, 2′-O-methylation and 'other modifications' from the human 18S rRNA dataset (see the 'Materials and methods' section). As shown in Supplementary Table S2, Modena again has the highest F1-scores on the pseudouridy-

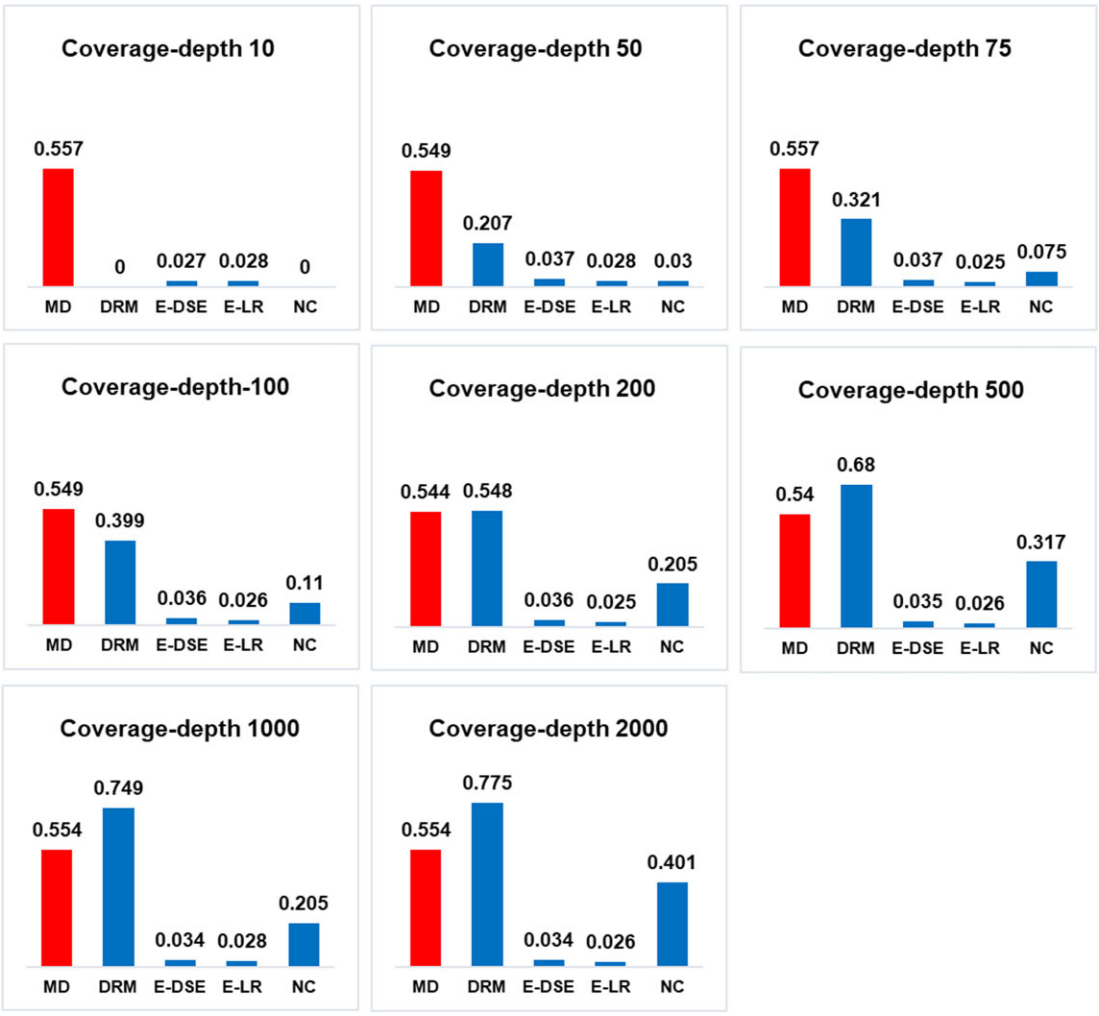


Figure 5. F1-scores of the five algorithms on the artificial 'curlcakes test dataset' (**NC**: Nanocompore; **DRM**: Drummer; **E-DSE:** Epinano Delta-Sum-Error; **E-LR**: Epinano Linear Regression). This dataset comprises four curlcakes with m^6A modification in all sequence contexts. Due to the limited number of reads of sufficient length, only one sample comprising eight subsamples with coverage-depths ranging from 10 to 2000 was constructed. Interestingly Modena showed strong performance at the coverage-depth 10 subsample. On the other hand, Drummer had zero F1-score on the coverage-depth of 10, but significantly outperformed Modena at coverage-depths 500–2000. Out of all benchmark datasets, Modena had the worst performance on this dataset (excluding the coverage-depth of 10 case).

lation and methylation datasets (across all coverage depths) and has the highest F1-scores on the 'other-modifications' dataset at coverage-depths of 10, 50, 200 and 500, while Drummer has the highest F1-scores at coverage-depths of 75 and 100. These results are consistent with those shown in Supplementary Table S1, with some minor differences that are expected, since the *E. coli*/*S. cerevisiae* and human 18S rRNA datasets, while somewhat similar, are different in terms of the number and types of modifications, as well as their respective k-mer contexts.

Next, we analysed the performance of algorithms with respect to each modification type. This was done for Sample 1 for the *E. coli* and *S. cerevisiae* (Supplementary Data S3) and for the human 18S rRNA (Supplementary Data S4).

Results follow the same trend as in Figures 3 and 4. Namely, Modena, Epinano-DSE and Epinano-LR show stable performance across coverage-depths, whereas Drummer and Nanocompore improve with higher coverage-depths. In general, given the modification type and coverage-depth, it is not the case that all test examples are either recognized or missed by any algorithm. Instead, some are identified while others are missed. This shows that the same modification types produce different 'error signatures', depending on the sequence context and coverage-depth, which is in line with recent literature (40). Thus, only the same modification type within a same k-mer context can be considered truly unique (or more precisely similar) in terms of basecaller-error patterns and distributions shifts. This should not be surprising since the sequence context (i.e. k-mer) significantly affects current intensity as well as dwell times. However, some modifications (e.g. m62A) do seem to be generally more easily identifiable than others (e.g. m^6A). While Modena was better at recognizing m^3U compared to other algorithms (irrespective of coverage-depth), it was the only algorithm that missed 3 out of 3 m^6A in the two rRNA datasets. It was unclear if these were just statistical outliers or if Modena cannot detect m^6A modifications in general. To investigate this in detail, we used the artificial m^6A curlcake dataset from (30). This particular dataset set has highly reliable ground-truth labels which is crucial for conducting benchmarking (see Supplementary Discussion S2).

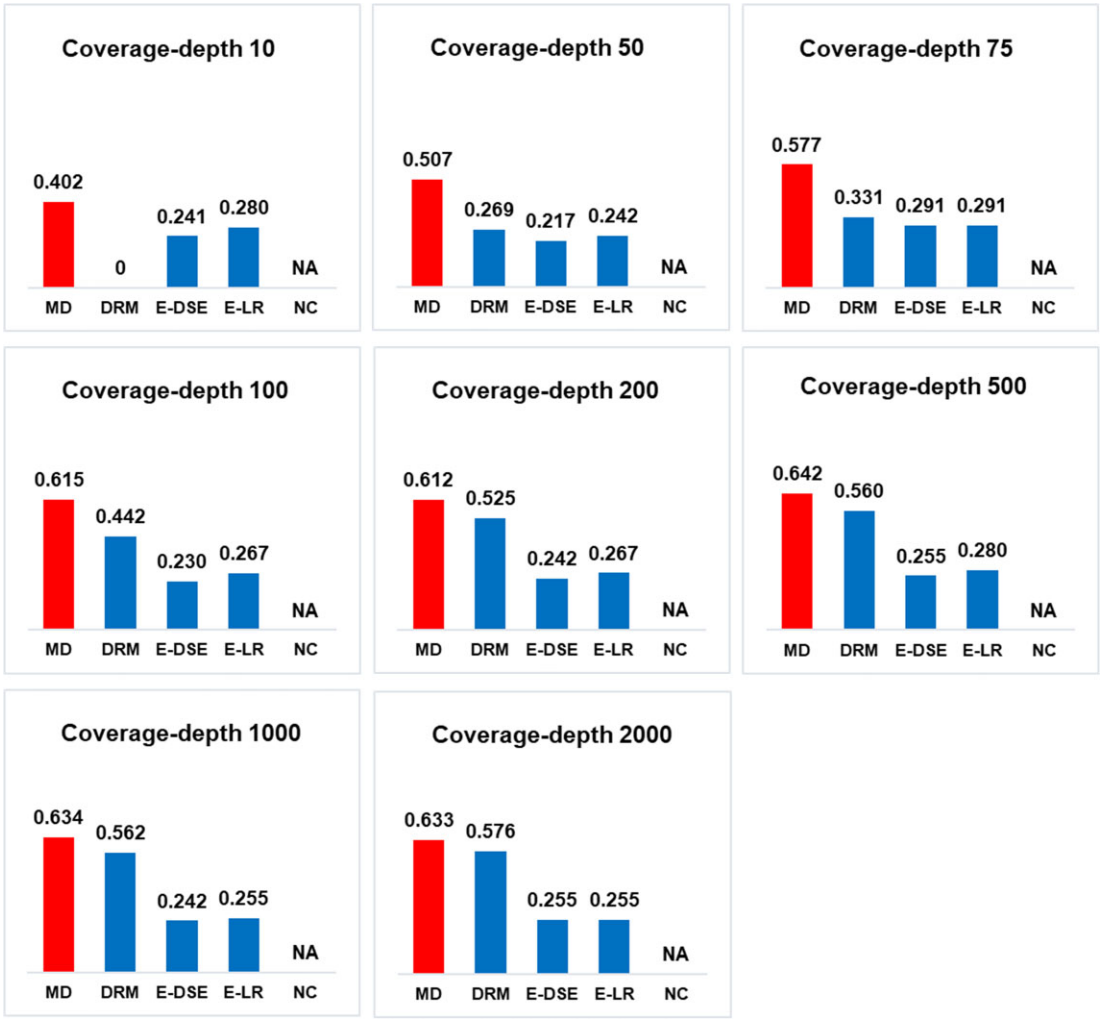


Figure 6. Average F1-scores of the five algorithms on the artificial 'oligos dataset' (**NC**: Nanocompore; **DRM**: Drummer; **E-DSE**: Epinano Delta-Sum-Error; **E-LR**: Epinano Linear Regression). This dataset comprises three short oligos, each containing three modifications (Oligo 1: three m^6A; Oligo 2: one I, one m^5C, one Ψ; Oligo 3: one $m^6{}_2A$, one m^1G one Am). Due to the limited number of reads with sufficient length, only three independent samples comprising eight subsamples with coverage-depths ranging from 10 to 2000 were constructed.

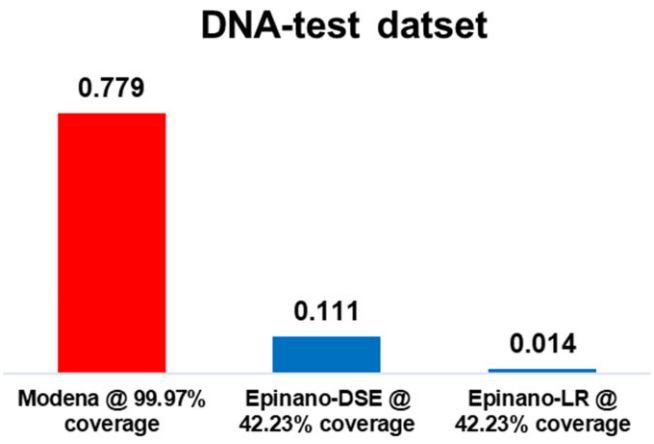


Figure 7. F1-scores of the three algorithms compared on the DNA methylation test dataset (**DSE:** Delta-Sum-Error; **LR**: Linear Regression). 'Coverage' is the percentage of genomic positions for which the respective algorithm made a prediction. Note that test datasets are not identical since Epinano's F1-score was computed based on the 42.23% of predicted positions, whereas Modena's F1-score was computed based on 99.97% of predicted positions.

As shown in Figure 5 and Supplementary Data S5, Modena's performance was indeed somewhat weaker on this particular modification type, which is in line with the fact that it failed to detect m^6A in the rRNA datasets. While Modena still outperformed other algorithms at lower coverage-depths (10–100) and had very similar performance as Drummer at coverage-depth of 200, Drummer clearly outperformed Modena at coverage-depths of 500, 1000 and 2000. However, it is important to emphasize that if the goal is to identify a specific modification type like m^6A, for which a supervised algorithm (e.g. m6ANet) exists, a supervised algorithm should be preferred since these algorithms are specifically trained for that modification. Finally, although this dataset features only one modification type, it is still suitable for benchmarking because the same modifications produce different 'error signatures' depending on the sequence context, as previously stated.

Modena shows robust performance across different stoichiometry levels

To test Modena's performance across different stoichiometry levels, we used another benchmark dataset (37). This dataset ('oligos dataset') is artificial and has highly reliable ground-

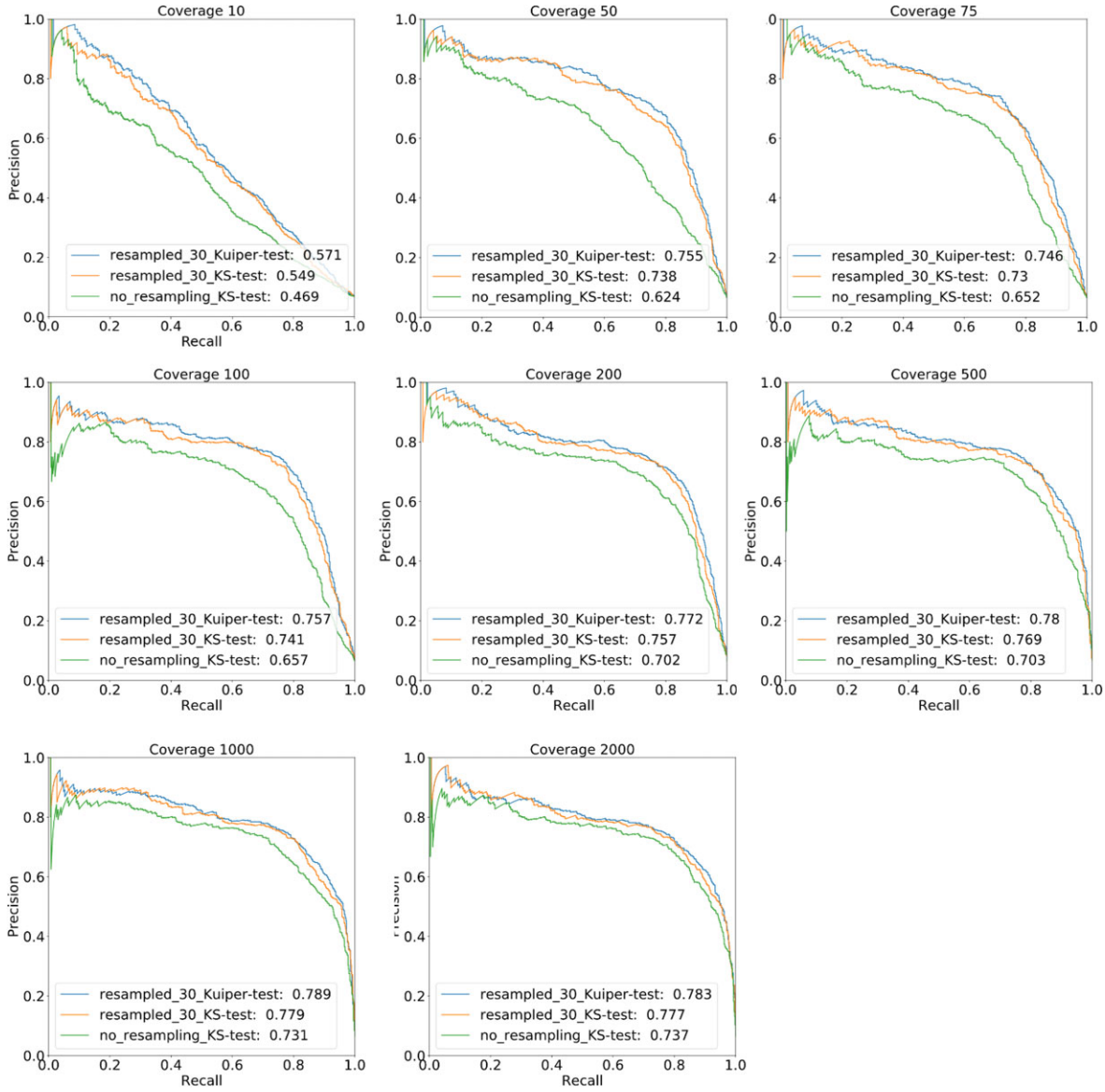


Figure 8. Precision–Recall curves (PR curves) for Sample 1 (*E. coli and S. cerevisiae* rRNA dataset) for different coverage-depths. As shown, resampling increases the area under the PR curves (i.e. AUPRC scores) across all coverage-depths. Kuiper test further improves AUPRC scores across all coverage-depths, although to a lesser extent.

truth labels, making it suitable for testing algorithm performance across various stoichiometry levels. Specifically, rRNA datasets are *in vivo* datasets where the stoichiometries of modified positions vary. In contrast, the artificial m^{6}A 'curlcakes' and 'oligos' datasets have very high stoichiometry levels of modified positions and are ideal for *in silico* stoichiometry tests. Out of the two, 'oligos-dataset' comprises seven different modification types as opposed to just one in the m^{6}A 'curlcakes' benchmark dataset, and was therefore chosen for testing stoichiometry. We first analysed the performance of all algorithms at ∼100% stoichiometry levels. Note that we were not able to test Nanocompore due to several 'bus errors', which are presumably caused by short read lengths (i.e. short transcript lengths). However, considering its performance on other datasets, it is highly unlikely that Nanocompore would be competitive to Modena. Following the approach described in the 'Materials and methods' section, we were able to construct three independent samples, each comprising eight subsamples with coverage-depths ranging from 10 to 2000. Detailed results are shown in Supplementary Data S6, and average F1-scores are shown in Figure 6. As shown, the same general trends are observed as before: (i) Modena outperforms other algorithms at all coverages, (ii) Modena, Epinano-DSE and Epinano-LR show stable performance irrespective of the coverage-depth, and (iii) while uncompetitive at lower coverages, Drummer becomes competitive with Modena at coverage-depths >200.

Next, to test the ability of different algorithms to detect modifications at different stoichiometry levels (25%, 50% and 75%), we followed a standard approach (40) and replaced a given percentage of modified with unmodified reads (see the 'Materials and methods' section). Average results across 10 independent samples are shown in Table 1. As shown, the same general trend can be observed. Modena strongly outperforms Drummer at lower coverage-depths, while Epinano-DSE and Epinano-LR are quite stable with re-

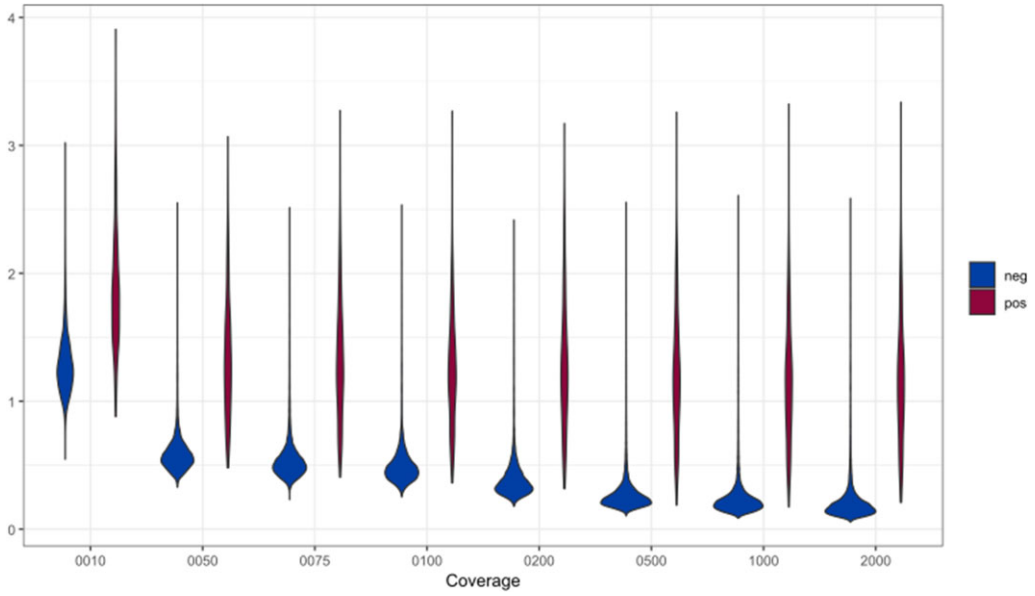


Figure 9. Violin plots of Modena score distributions for positive and negative test cases across different coverage-depths for Sample 1 of the *E. coli*/*S. cerevisiae* benchmark dataset are shown. Two well-separated clusters can be seen for all coverage-depths. The final Step 5 of our algorithm (1D score-clustering) leverages this separation to determine the classification threshold. Note that this represents a different paradigm from the standardly used *P*-value based thresholds. As shown in our study, this approach is not limited to Modena and can, in principle, be applied to any threshold-based unsupervised algorithm.

Table 1. Average F1-scores of the five algorithms on the artificial 'oligos dataset' with different stoichiometry levels (**NC**: Nanocompore; **DRM**: Drummer; **E-DSE:** Epinano Delta-Sum-Error; **E-LR**: Epinano Linear Regression)

Average F1-scores						
Coverage-depth		**20**			**50**	
Stoichiometry level	**25%**	**50%**	**75%**	**25%**	**50%**	**75%**
Modena	0.189	0.266	0.406	0.212	0.365	0.505
Drummer	0	0	0.017	0	0.088	0.206
Epinano-DSE	0.075	0.155	0.210	0.163	0.242	0.240
Epinano-LR	0.068	0.19	0.239	0.146	0.247	0.243
Coverage-depth		**75**			**100**	
Stoichiometry level	**25%**	**50%**	**75%**	**25%**	**50%**	**75%**
Modena	0.204	0.448	0.505	0.258	0.467	0.557
Drummer	0.026	0.186	0.294	0.068	0.257	0.35
Epinano-DSE	0.226	0.235	0.240	0.219	0.243	0.243
Epinano-LR	0.223	0.239	0.262	0.204	0.258	0.266
Coverage-depth		**200**			**500**	
Stoichiometry level	**25%**	**50%**	**75%**	**25%**	**50%**	**75%**
Modena	0.384	0.53	0.595	0.494	0.586	0.639
Drummer	0.177	0.367	0.448	0.344	0.502	0.56
Epinano-DSE	0.218	0.247	0.247	0.234	0.251	0.251
Epinano-LR	0.217	0.258	0.273	0.239	0.266	0.273
Coverage-depth		**1000**			**2000**	
Stoichiometry level	**25%**	**50%**	**75%**	**25%**	**50%**	**75%**
Modena	0.549	0.618	0.624	0.568	0.617	0.623
Drummer	0.435	0.566	0.575	0.516	0.573	0.575
Epinano-DSE	0.232	0.251	0.259	0.232	0.247	0.255
Epinano-LR	0.232	0.273	0.270	0.232	0.262	0.259

For a given coverage-depth and modified/unmodified condition, we mixed reads from the three independent samples into a single dataset (with one positive and one negative subset) and performed sampling with replacement 10 times. In this way, for a given coverage-depth, and modified/unmodified condition we obtained 10 samples, each comprising three subsamples with modified to unmodified read ratios of 25:75, 50:50 and 75:25. Modena outperformed other algorithms for all combinations except for the coverage-depth 75, at 25% stoichiometry level.

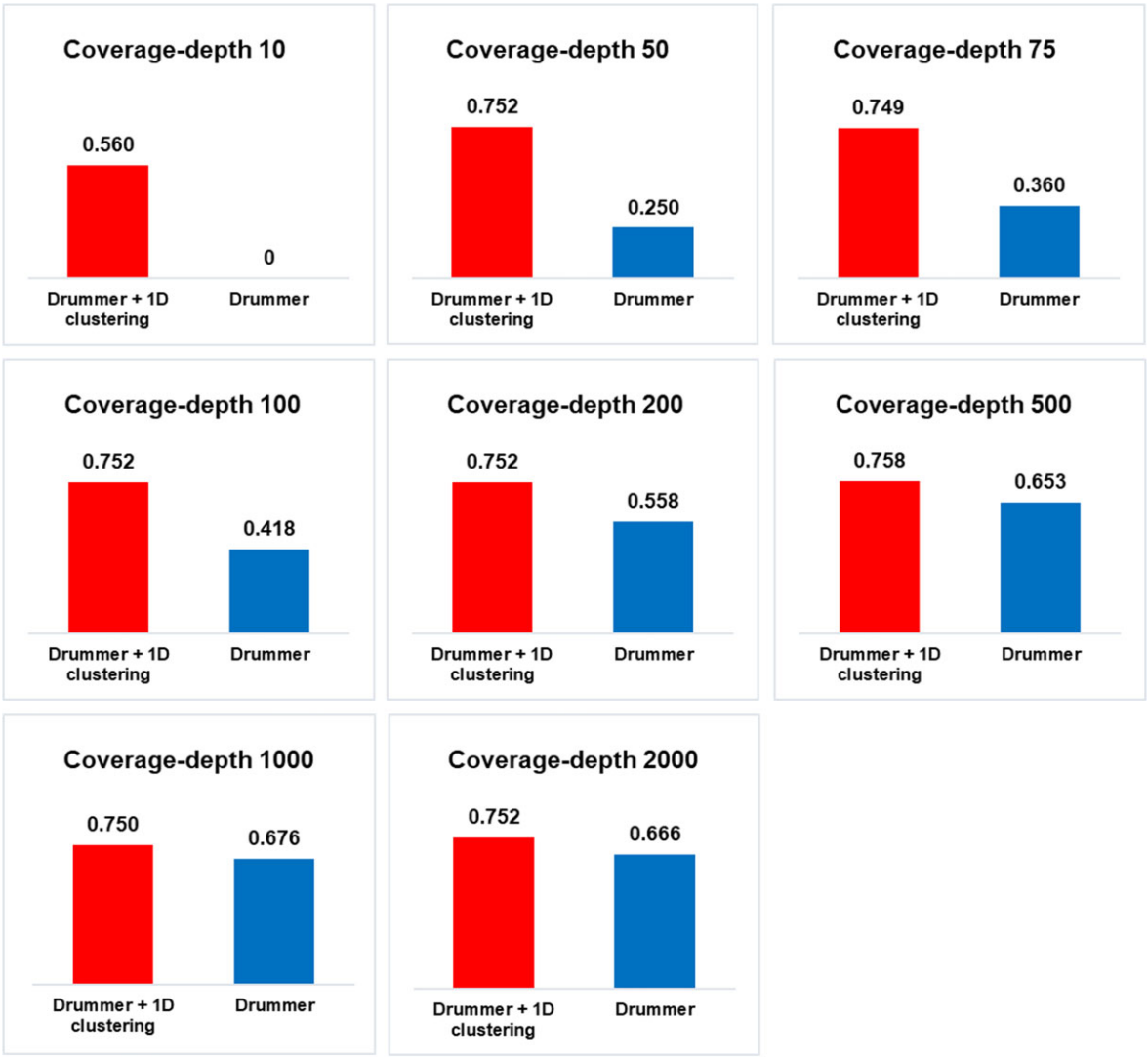


Figure 10. Average F1-scores (for Samples 1 through 10, *E.coli*/*S. cerevisiae* dataset) with coverage-depths ranging from 10 to 2000 are shown. **Drummer**: original Drummer algorithm with *P*-value and odds ratio-based threshold; **Drummer + 1D clustering**: Drummer algorithm (i.e. G-test statistic) with 1D score-clustering step (see Figure 2). For detailed results across all samples, see Supplementary Table S5 and Supplementary Data S8.

spect to coverage-depths and modification stoichiometry. We also analysed the performance of different algorithms with respect to stoichiometry on the *in vivo* human 18S rRNA dataset. Not surprisingly, the detection rates of all algorithms were much lower for positions with very low stoichiometries. However, increasing the coverage-depth improves detection rates (Supplementary Data S7).

Modena shows consistently strong performance on the DNA test dataset

Finally, we tested Modena's performance on a DNA test dataset. Although the authors of Nanocompore, Drummer and Epinano did not address their applicability to a DNA setting, we assumed these algorithms should, in principle, be applicable to DNA molecules as well. We selected DNA methylation as a test dataset for two reasons. Firstly, the methyl group (CH_3) is a 'small molecule', which is not expected to significantly alter current intensity signals, thus making it more challenging for computational algorithms to detect. Secondly, the 5mC modification has been utilized in several previous studies as a DNA test dataset (27,28).

To create the negative/control sample, all modifications on the native genomic DNA were removed as described in the 'Materials and methods' section. The positive sample was created by treating the WGA sample with M.SssI methyltransferase, which converts nearly all cytosines in a CpG context to 5mC (27).

As shown in Figure 7, Modena performed very well on this dataset, whereas the results of other algorithms suggest they are likely optimized for the RNA setting. Modena achieved an F1-score of 0.779 (Recall 0.769 and Precision 0.789), covering 99.97% of the genome, meaning that the algorithm predicted 99.97% of genomic positions as either positive or negative. In contrast, the second-best method, Epinano Distance-Sum-Error, achieved an F1-score of 0.111 (with Recall 0.059 and Precision 0.843) but covered only 42.23% of the genome. The genome coverages of Drummer and Nanocompore were just 0.016% and 0.229%, respectively, and were therefore not analysed. Additionally, we also tested Modena's performance on only the 42.23% of genomic positions that were predicted

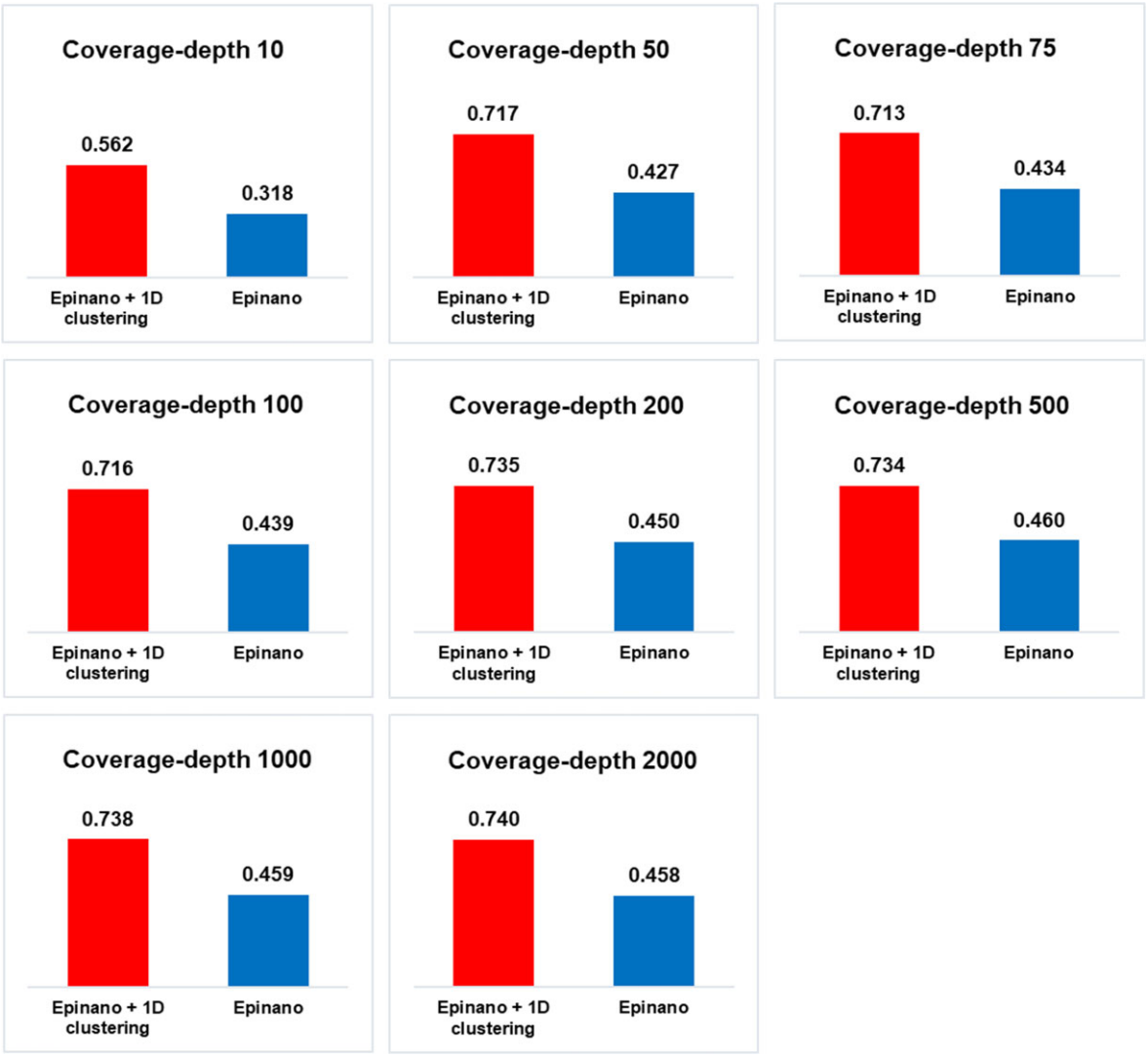


Figure 11. Average F1-scores (for Samples 1 through 10, *E.coli*/*S. cerevisiae* dataset) with coverage-depths ranging from 10 to 2000 are depicted. **Epinano:** Epinano-DSE algorithm with z-score based threshold; **Epinano + 1D clustering**: Epinano-DSE algorithm with 1D score-clustering step (see Figure 2). For detailed results across all samples, see Supplementary Table S6 and Supplementary Data S8.

(as positive or negative) by Epinano-DSE. On these positions, Modena achieved an F1-score of 0.762 (Recall 0.713 and Precision 0.818).

To the best of our knowledge, and according to recent reviews (56,57), the only algorithms developed for the detection of DNA modifications are either supervised, e.g. Dorado (ONT), Remora (ONT), Megalodon (ONT) or (28,29,58), or are unsupervised, but provide only ranking score as their output (43,59). Therefore, they cannot be compared to Modena in any meaningful way since Modena is an unsupervised classifier and its output are class labels, not ranking scores. That said, if the goal is to find a particular modification type for which a supervised algorithm exists (e.g. 5mC), a supervised algorithm should be used since these are specifically trained for that modification. Modena and other unsupervised algorithms should be used when no such supervised algorithm/tool is available.

Analysis of specific steps in Modena showed the usefulness of resampling and the Kuiper test

In order to assess the importance of specific steps in the Modena's 'pipeline', we conducted an ablation test by omitting the resampling step and/or replacing the Kuiper test with the standard KS-test. Both resampling and the Kuiper test have improved ranking (Figure 8; green curves represent standard KS-test without resampling, orange curves represent KS-test with resampling and blue curves represent Kuiper test with resampling) and F1-scores (Supplementary Table S3), although the benefit of using the Kuiper test instead of the KS-test was more modest. Additionally, we compared the running times of the algorithms (Supplementary Table S4). As shown, Nanocompore was the slowest with a mean running time of 444.7 s, followed by Modena with a mean running time of 298.7 s, then by Drummer (125.0 s) and the fastest Epinano (37 s).

Principles of 1D score-clustering approach

Importantly, Modena employs a non-standard 1D score-clustering approach to determine the classification threshold in the final step (Step 5 in Figure 1B). This represents a different paradigm from the commonly used 'hard-threshold' based approaches, typically determined by *P*-values. To illustrate this approach, the Modena output-score distributions for Sample 1 are displayed in Figure 9.

As shown in Figure 9, the score distributions of positives and negatives form two distinct clusters, and 1D score-clustering leverages this. A successful 1D score-clustering requires two conditions: (i) good ranking (which is also a requirement for any standard *P*-value based threshold) and (ii) the existence of two 'well-separated' clusters. The aim of resampling (Step 2 in Figure 1A), Kuiper test (Step 3 in Figure 1A) and distance summing (Step 4 in Figure 1B) is to achieve 'good ranking', and these steps are specific to Modena. The final 1D score-clustering step (Step 5 in Figure 1B) is more general and can, in principle, be applied to any threshold-based unsupervised algorithm. Its success, though, depends on satisfying the two conditions mentioned above.

The requirement for two well-separated clusters (i.e. condition two) offers another advantage over the standard approach. Namely, the existence of two clusters can be relatively quickly established either by simple visual inspection (Supplementary Figure S3) or by employing quantitative measures for assessing the 'goodness of clustering' (60). On the other hand, *P*-value based cut-offs require strict mathematical conditions to be met, which is often not the case in practice. Otherwise, they are not theoretically justified and might be greatly misleading in their interpretation.

Adding the 1D score-clustering step significantly improves Drummer and Epinano

We applied 1D score-clustering approach to the three comparison algorithms to demonstrate (as a proof-of-concept) that 1D score-clustering is not limited to Modena. First, we examined the PR curves of Drummer, Epinano and Nanocompore (Supplementary Figures S4–S6). Since the areas under the PR curves were not particularly high, we summed scores (i.e. G-test statistics in the case of Drummer and z-scores in the case of Epinano-DSE) of the five neighbouring positions (i.e. central position and positions −2, −1, +1 and + 2) for all three algorithms (Step 2, Figure 2). This was done to improve rankings (i.e. areas under the PR curves) since 'good ranking' is a prerequisite for applying 1D score-clustering, as mentioned earlier. Summing the scores of neighbouring positions is a general approach reported previously (43). This significantly improved the rankings of all algorithms. Specifically, Drummer and Epinano rankings were improved considerably after this step (Supplementary Figures S4 and S5), while Nanocompore rankings, although also improved (Supplementary Figure S6), still remained quite low in the 'absolute sense' (ranging from ∼0.124 to ∼0.54). Nanocompore was therefore discarded from further analyses since suboptimal ranking will not yield good classification results, neither through the 'dynamic' 1D score-clustering approach nor the standard *P*-value based 'hard-threshold' approaches. Since 'good ranking' (measured by an AUPRC score) has a limited value in the absence of a classification threshold, the crucial next step is to determine the classification threshold. Here, we applied Modena's 1D score-clustering approach instead of standardly used *P*-values and/or other measures like odds ratios (used for example, in Drummer).

As shown in Figures 10 and 11, the application of 'score-summing' and 1D score-clustering (Steps 2 and 3, Figure 2), led to significant improvements in the F1-scores of Drummer and Epinano-DSE. The improvement was particularly striking in the case of Drummer. For example, incorporating the 1D score-clustering step increased the average F1-score at coverage-depth of 50 from 0.25 to 0.752, which is a three-fold increase. Notably, both Drummer and Epinano belong to a different class of algorithms (basecaller-error-based) compared to Modena (signal-based).

Next, we analysed the ability of Drummer + 1D score-clustering and Epinano + 1D score-clustering to detect specific modification types on the *E. coli* and *S. cerevisiae* dataset (Sample 1). In general, detection rates of specific modifications by all algorithms on the *E. coli* and *S. cerevisiae* datasets are quite similar (Supplementary Data S9). However, there are also some differences between them. For example, the m^4Cm modification was identified by both Modena and Epinano + 1D but missed by Drummer + 1D. Additionally, two ac4C modifications are correctly identified by Modena (across all coverage-depths), but are missed by Epinano + 1D (across all coverage depths). One of these modifications is correctly identified by Drummer + 1D, but only at lower coverage-depths (50–100). Another example is m^5C, where all five positions are identified by Modena across all coverage depths. In contrast, Drummer + 1D identifies two out of five positions, and Epinano + 1D detects two or three out of five, depending on the coverage-depth. Another example is Cm, where Modena detects 10 out of 11 Cm modifications. In contrast, Drummer + 1D and Epinano + 1D identify six or seven of those, depending on the coverage-depth. These examples illustrate that algorithms do not return identical results.

We also conducted a Precision/Recall analysis. The summary is shown in Table 2 below, while detailed results can be found in Supplementary Data S8. As shown, Drummer + 1D clustering outperforms Modena in terms of F1-scores at lower coverage-depths (10–100), while Modena outperforms Drummer + 1D at coverage-depths above 100. Drummer + 1D also outperforms Epinano + 1D at all coverage-depths (except at a coverage-depth of 10, where there is a very small difference in favour of Epinano + 1D). Interestingly, Modena has substantially higher Recall rates across all coverage-depths, whereas Drummer + 1D and Epinnao + 1D outperform Modena in terms of Precision at all coverage-depths. This shows that Modena discovers more modifications, but it has a higher False Positive rate. This suggests that these algorithms are fundamentally different and that it might be beneficial to combine them.

In summary, a vast number of epigenetic and epitranscriptomic modifications, coupled with significant limitations of existing experimental methods, limit the application of supervised learning algorithms to a small number of well-characterized modifications. Consequently, the utilization of unsupervised machine learning algorithms becomes essential. Given the recent successes of several unsupervised algorithms in various biological settings (37,42,50), this study focused on advancing computational aspects. Here, we developed a new signal-based unsupervised learning algorithm, Modena, which features three novel steps: (i) resampling, (ii)

Table 2. Average Precision, Recall and F1-scores of Modena compared to 'hybrid algorithms'

Coverage-depth 10	Modena	Drummer + 1D clustering	E-DSE + 1D clustering
Recall/Sensitivity	0.802	0.713	0.647
Precision	0.267	0.470	0.505
F1	0.396	0.560	0.562
Coverage-depth 50	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.736	0.646	0.616
Precision	0.721	0.900	0.861
F1	0.729	0.752	0.717
Coverage-depth 75	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.756	0.635	0.608
Precision	0.680	0.913	0.860
F1	0.713	0.749	0.713
Coverage-depth 100	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.761	0.642	0.611
Precision	0.731	0.907	0.864
F1	0.745	0.752	0.716
Coverage-depth 200	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.774	0.641	0.637
Precision	0.736	0.910	0.870
F1	0.755	0.752	0.735
Coverage-depth 500	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.784	0.647	0.635
Precision	0.737	0.913	0.869
F1	0.760	0.758	0.734
Coverage-depth 1000	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.792	0.637	0.641
Precision	0.730	0.911	0.869
F1	0.760	0.750	0.738
Coverage-depth 2000	**Modena**	**Drummer + 1D clustering**	**E-DSE + 1D clustering**
Recall/Sensitivity	0.799	0.639	0.644
Precision	0.725	0.913	0.870
F1	0.760	0.752	0.740

Kuiper test and (iii) 1D score-clustering. Additionally, we constructed several new benchmark datasets offering advantages over previously used test datasets. Modena showed excellent performance on various datasets (Figures 3–7, Table 1 and Supplementary Data S1, S2, S5 and S6). To the best of our knowledge, Modena is currently the only unsupervised algorithm which detects both epigenetic and epitranscriptomic modifications.

Finally, we provided a proof-of-concept demonstration of how Modena's 'dynamic clustering' approach, which relies on the 1D clustering of output scores, offers an attractive alternative to the commonly used 'hard-thresholds', based on *P*-values. Specifically, when the 1D score-clustering step was combined with two existing algorithms, it led to a considerable improvement in their performance (Figures 10 and 11, and Supplementary Tables S5 and S6). In some cases, this improvement was quite remarkable, achieving up to a threefold increase in F1-score (Figure 10, coverage-depth of 50). Our results suggest that the 1D score-clustering approach should, in principle, have broader applicability in various unsupervised learning contexts. However, exploring this broader applicability extends beyond this study's scope and is a topic for future research.

Data availability

Escherichia coli and *S. cerevisiae* rRNA test datasets are available at https://zenodo.org/records/12659159 (Samples 1–5) and https://zenodo.org/records/12659174 (Samples 6–10).

The DNA test dataset is available at https://zenodo.org/records/10031901.

Other test datasets are available at https://zenodo.org/records/13236618.

Modena code is available at https://github.com/sbidin/modena (https://doi.org/10.5281/zenodo.14036835).

Supplementary data

Supplementary Data are available at NAR Online.

Acknowledgements

S.Ba. has been supported by the SINGA A*STAR scholarship. M.Š. has been supported by funding from GIS, A*STAR, Singapore. *E. coli* strain K12 MG1655 was a gift kindly given to us by Dr Swaine Chen's lab at the Genome Institute of Singapore, A*STAR, Singapore. The authors would like to thank Prof. Bojan Basrak, Prof. Stjepan Šebek and Darko Brborović for their valuable comments related to statistical analysis. Author I.V. would like to express gratitude to Prof. Daniel P. Depledge for his insightful comments during the manuscript revision process.

Author contributions: M.Š. and I.V. conceived the project. I.V. designed Modena, S.Bi. implemented it with help from I.V. and M.V. I.V., M.V., S.Ba. and S.Bi. performed bioinformatics analysis and evaluation with the contribution of M.Š. Z.L. performed ONT sequencing of *E. coli* DNA data. I.V. and M.S. organized the manuscript. I.V., M.S., S.Bi. M.V., S.Ba. and Z.L. wrote the manuscript with help from R.F. and J.L. M.S. supervised the project. M.Š., R.F. and J.L. provided mentorship and support during the project.

Funding

AI Singapore [AISG-RPKS-2019-001 to M.Š.]. Funding for open access charge: Genome Institute of Singapore, A*STAR.

Conflict of interest statement

M.Š. has received travel funds to speak at events hosted by Oxford Nanopore Technologies. Oxford Nanopore Technologies and AI Singapore jointly funded the AI-driven De Novo Diploid Assembler project led by M.Š.

References

1. Sood,A.J., Viner,C. and Hoffman,M.M. (2019) DNAmod: the DNA modification database. *J. Cheminform.*, **11**, 30.
2. Ding,H., Bailey,A.D., Jain,M., Olsen,H. and Paten,B. (2020) Gaussian mixture model-based unsupervised nucleotide modification number detection using nanopore-sequencing readouts. *Bioinformatics*, **36**, 4928–4934.
3. Boccaletto,P., Stefaniak,F., Ray,A., Cappannini,A., Mukherjee,S., Purta,E., Kurkowska,M., Shirvanizadeh,N., Destefanis,E., Groza,P., *et al.* (2022) MODOMICS: a database of RNA modification pathways. 2021 update. *Nucleic Acids Res.*, **50**, D231–D235.
4. Shi,H., Chai,P., Jia,R. and Fan,X. (2020) Novel insight into the regulatory roles of diverse RNA modifications: re-defining the bridge between transcription and translation. *Mol. Cancer*, **19**, 78.
5. Geula,S., Moshitch-Moshkovitz,S., Dominissini,D., Mansour,A.A., Kol,N., Salmon-Divon,M., Hershkovitz,V., Peer,E., Mor,N., Manor,Y.S., *et al.* (2015) Stem cells. m6A mRNA methylation facilitates resolution of naïve pluripotency toward differentiation. *Science*, **347**, 1002–1006.
6. Vu,L.P., Pickering,B.F., Cheng,Y., Zaccara,S., Nguyen,D., Minuesa,G., Chou,T., Chow,A., Saletore,Y., MacKay,M., *et al.* (2017) The N6-methyladenosine (m6A)-forming enzyme METTL3 controls myeloid differentiation of normal hematopoietic and leukemia cells. *Nat. Med.*, **23**, 1369–1376.
7. Boo,S.H. and Kim,Y.K. (2020) The emerging role of RNA modifications in the regulation of mRNA stability. *Exp. Mol. Med.*, **52**, 400–408.
8. Cui,L., Ma,R., Cai,J., Guo,C., Chen,Z., Yao,L., Wang,Y., Fan,R., Wang,X. and Shi,Y. (2022) RNA modifications: importance in immune cell biology and related diseases. *Signal Transduct. Target. Ther.*, **7**, 334.
9. Jiang,X., Liu,B., Nie,Z., Duan,L., Xiong,Q., Jin,Z., Yang,C. and Chen,Y. (2021) The role of m6A modification in the biological functions and diseases. *Signal Transduct. Target. Ther.*, **6**, 74.
10. Behm,M., Wahlstedt,H., Widmark,A., Eriksson,M. and Öhman,M. (2017) Accumulation of nuclear ADAR2 regulates A-to-I RNA editing during neuronal development. *J. Cell Sci.*, **130**, 745–755.
11. Ekdahl,Y., Farahani,H.S., Behm,M., Lagergren,J. and Öhman,M. (2012) A-to-I editing of microRNAs in the mammalian brain increases during development. *Genome Res.*, **22**, 1477–1487.
12. Gross,J.A., Pacis,A., Chen,G.G., Drupals,M., Lutz,P.-E., Barreiro,L.B. and Turecki,G. (2017) Gene-body 5-hydroxymethylation is associated with gene expression changes in the prefrontal cortex of depressed individuals. *Transl. Psychiatry*, **7**, e1119.
13. Batista,P.J., Molinie,B., Wang,J., Qu,K., Zhang,J., Li,L., Bouley,D.M., Lujan,E., Haddad,B., Daneshvar,K., *et al.* (2014) m6A RNA modification controls cell fate transition in mammalian embryonic stem cells. *Cell Stem Cell*, **15**, 707–719.
14. Mendel,M., Chen,K.-M., Homolka,D., Gos,P., Pandey,R.R., McCarthy,A.A. and Pillai,R.S. (2018) Methylation of structured RNA by the m6A writer METTL16 is essential for mouse embryonic development. *Mol. Cell*, **71**, 986–1000.
15. Rajendren,S. and Karijolich,J. (2022) The impact of RNA modifications on the biology of DNA virus infection. *Eur. J. Cell Biol.*, **101**, 151239.
16. Sun,J., Cheng,B., Su,Y., Li,M., Ma,S., Zhang,Y., Zhang,A., Cai,S., Bao,Q., Wang,S., *et al.* (2022) The potential role of m6A RNA methylation in the aging process and aging-associated diseases. *Front. Genet.*, **13**, 869950.
17. Gatsiou,A. and Stellos,K. (2023) RNA modifications in cardiovascular health and disease. *Nat. Rev. Cardiol.*, **20**, 325–346.
18. Liu,C., Gu,L., Deng,W., Meng,Q., Li,N., Dai,G., Yu,S. and Fang,H. (2022) N6-methyladenosine RNA methylation in cardiovascular diseases. *Front. Cardiovasc. Med.*, **9**, 887838.
19. Chen,H., Yao,J., Bao,R., Dong,Y., Zhang,T., Du,Y., Wang,G., Ni,D., Xun,Z., Niu,X., *et al.* (2021) Cross-talk of four types of RNA modification writers defines tumor microenvironment and pharmacogenomic landscape in colorectal cancer. *Mol. Cancer*, **20**, 29.
20. Berdasco,M. and Esteller,M. (2022) Towards a druggable epitranscriptome: compounds that target RNA modifications in cancer. *Br. J. Pharmacol.*, **179**, 2868–2889.
21. Horvath,S. and Raj,K. (2018) DNA methylation-based biomarkers and the epigenetic clock theory of ageing. *Nat. Rev. Genet.*, **19**, 371–384.
22. McIntyre,A.B.R., Gokhale,N.S., Cerchietti,L., Jaffrey,S.R., Horner,S.M. and Mason,C.E. (2020) Limits in the detection of m6A changes using MeRIP/m6A-seq. *Sci. Rep.*, **10**, 6590.
23. Furlan,M., Delgado-Tejedor,A., Mulroney,L., Pelizzola,M., Novoa,E.M. and Leonardi,T. (2021) Computational methods for RNA modification detection from nanopore direct RNA sequencing data. *RNA Biol.*, **18**, 31–40.
24. Lucas,M.C. and Novoa,E.M. (2023) Long-read sequencing in the era of epigenomics and epitranscriptomics. *Nat. Methods*, **20**, 25–29.
25. Flusberg,B.A., Webster,D.R., Lee,J.H., Travers,K.J., Olivares,E.C., Clark,T.A., Korlach,J. and Turner,S.W. (2010) Direct detection of DNA methylation during single-molecule, real-time sequencing. *Nat. Methods*, **7**, 461–465.
26. Vilfan,I.D., Tsai,Y.-C., Clark,T.A., Wegener,J., Dai,Q., Yi,C., Pan,T., Turner,S.W. and Korlach,J. (2013) Analysis of RNA base modification and structural rearrangement by single-molecule real-time detection of reverse transcription. *J. Nanobiotechnology*, **11**, 8.
27. Simpson,J.T., Workman,R.E., Zuzarte,P.C., David,M., Dursi,L.J. and Timp,W. (2017) Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods*, **14**, 407–410.
28. Ni,P., Huang,N., Zhang,Z., Wang,D.-P., Liang,F., Miao,Y., Xiao,C.-L., Luo,F. and Wang,J. (2019) DeepSignal: detecting DNA

methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics*, **35**, 4586–4595.
29. Liu,Q., Fang,L., Yu,G., Wang,D., Xiao,C.-L. and Wang,K. (2019) Detection of DNA base modifications by deep recurrent neural network on Oxford Nanopore sequencing data. *Nat. Commun.*, **10**, 2449.
30. Liu,H., Begik,O., Lucas,M.C., Ramirez,J.M., Mason,C.E., Wiener,D., Schwartz,S., Mattick,J.S., Smith,M.A. and Novoa,E.M. (2019) Accurate detection of m6A RNA modifications in native RNA sequences. *Nat. Commun.*, **10**, 4079.
31. Rand,A.C., Jain,M., Eizenga,J.M., Musselman-Brown,A., Olsen,H.E., Akeson,M. and Paten,B. (2017) Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods*, **14**, 411–413.
32. Nguyen,T.A., Heng,J.W.J., Kaewsapsak,P., Kok,E.P.L., Stanojević,D., Liu,H., Cardilla,A., Praditya,A., Yi,Z., Lin,M., *et al.* (2022) Direct identification of A-to-I editing sites with nanopore native RNA sequencing. *Nat. Methods*, **19**, 833–844.
33. Hendra,C., Pratanwanich,P.N., Wan,Y.K., Goh,W.S.S., Thiery,A. and Göke,J. (2022) Detection of m6A from direct RNA sequencing using a multiple instance learning framework. *Nat. Methods*, **19**, 1590–1598.
34. Acera Mateos,P., Zhou,Y., Zarnack,K. and Eyras,E. (2023) Concepts and methods for transcriptome-wide prediction of chemical messenger RNA modifications with machine learning. *Brief. Bioinform.*, **24**, bbad163.
35. Begik,O., Mattick,J.S. and Novoa,E.M. (2022) Exploring the epitranscriptome by native RNA sequencing. *RNA*, **28**, 1430–1439.
36. Liu,J., Huang,T., Yao,J., Zhao,T., Zhang,Y. and Zhang,R. (2023) Epitranscriptomic subtyping, visualization, and denoising by global motif visualization. *Nat. Commun.*, **14**, 5944.
37. Leger,A., Amaral,P.P., Pandolfini,L., Capitanchik,C., Capraro,F., Miano,V., Migliori,V., Toolan-Kerr,P., Sideri,T., Enright,A.J., *et al.* (2021) RNA modifications detection by comparative Nanopore direct RNA sequencing. *Nat. Commun.*, **12**, 7198.
38. Smirnov,N.V. (1939) On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Moscow Univ.*, **2**, 3–14.
39. Kuiper,N.H. (1960) Tests concerning random points on a circle. *Indag. Math.*, **63**, 38–47.
40. 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.
41. Tourancheau,A., Mead,E.A., Zhang,X.-S. and Fang,G. (2021) Discovering multiple types of DNA methylation from bacteria and microbiome using nanopore sequencing. *Nat. Methods*, **18**, 491–498.
42. 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.
43. Liu,Q., Georgieva,D.C., Egli,D. and Wang,K. (2019) NanoMod: a computational tool to detect DNA modifications using Nanopore long-read sequencing data. *BMC Genomics*, **20**, 78.
44. Fisher,W.D. (1958) On grouping for maximum homogeneity. *J. Am. Stat. Assoc.*, **53**, 789–798.
45. Wu,X. (1991) Optimal quantization by matrix searching. *J. Algorithms*, **12**, 663–673.
46. Lloyd,S. (1982) Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, **28**, 129–137.
47. Stephenson,W., Razaghi,R., Busan,S., Weeks,K.M., Timp,W. and Smibert,P. (2022) Direct detection of RNA modifications and structure using single-molecule nanopore sequencing. *Cell Genomics*, **2**, 100097.
48. Naarmann-de Vries,I.S., Zorbas,C., Lemsara,A., Piechotta,M., Ernst,F.G.M., Wacheul,L., Lafontaine,D.L.J. and Dieterich,C. (2023) Comprehensive identification of diverse ribosomal RNA modifications by targeted nanopore direct RNA sequencing and JACUSA2. *RNA Biol.*, **20**, 652–665.
49. Abebe,J.S., Verstraten,R. and Depledge,D.P. (2022) Nanopore-based detection of viral RNA modifications. *mBio*, **13**, e0370221.
50. Abebe,J.S., Price,A.M., Hayer,K.E., Mohr,I., Weitzman,M.D., Wilson,A.C. and Depledge,D.P. (2022) DRUMMER—Rapid detection of RNA modifications through comparative nanopore sequencing. *Bioinformatics*, **38**, 3113–3115.
51. Liu,H., Begik,O. and Novoa,E.M. (2021) EpiNano: detection of m6A RNA modifications using Oxford Nanopore direct RNA sequencing. *Methods Mol Biol.*, **2298** , 31–52.
52. Taoka,M., Nobe,Y., Yamaki,Y., Sato,K., Ishikawa,H., Izumikawa,K., Yamauchi,Y., Hirota,K., Nakayama,H., Takahashi,N., *et al.* (2018) Landscape of the complete RNA chemical modifications in the human 80S ribosome. *Nucleic Acids Res.*, **46**, 9289–9298.
53. Taoka,M., Nobe,Y., Yamaki,Y., Yamauchi,Y., Ishikawa,H., Takahashi,N., Nakayama,H. and Isobe,T. (2016) The complete chemical structure of *Saccharomyces cerevisiae* rRNA: partial pseudouridylation of U2345 in 25S rRNA by snoRNA snR9. *Nucleic Acids Res.*, **44**, 8951–8961.
54. Jeni,L.A., Cohn,J.F. and De La Torre,F. (2013) Facing imbalanced data–recommendations for the use of performance metrics. In *Humaine Association Conference on Affective Computing and Intelligent Interaction, 02-05 September 2013, Geneva, Switzeraland*. IEEE, Piscataway, NJ, USA. pp. 245–251.
55. Siblini,W., Fréry,J., He-Guelton,L., Oblé,F. and Wang,Y.-Q. (2020) Master your metrics with calibration. In *Advances in Intelligent Data Analysis XVIII. IDA 2020. Lecture Notes in Computer Science*. Berthold,M., Feelders,A. and Krempl,G. (eds.) **Vol. 12080**, Springer, Cham, Switzerland.
56. Wan,Y.K., Hendra,C., Pratanwanich,P.N. and Göke,J. (2022) Beyond sequencing: machine learning algorithms extract biology hidden in Nanopore signal data. *Trends Genet.*, **38**, 246–257.
57. Liu,Y., Rosikiewicz,W., Pan,Z., Jillette,N., Wang,P., Taghbalout,A., Foox,J., Mason,C., Carroll,M., Cheng,A., *et al.* (2021) DNA methylation-calling tools for Oxford Nanopore sequencing: a survey and human epigenome-wide evaluation. *Genome Biol.*, **22**, 295.
58. Bonet,J., Chen,M., Dabad,M., Heath,S., Gonzalez-Perez,A., Lopez-Bigas,N. and Lagergren,J. (2022) DeepMP: a deep learning tool to detect DNA base modifications on Nanopore sequencing data. *Bioinformatics*, **38**, 1235–1243.
59. Stoiber,M., Quick,J., Egan,R., Lee,J.E., Celniker,S., Neely,R.K., Loman,N., Pennacchio,L.A. and Brown,J. (2017) *De novo* identification of DNA modifications enabled by genome-guided nanopore signal processing. bioRxiv doi: https://doi.org/10.1101/094672, 10 April 2017,preprint: not peer reviewed.
60. Saitta,S., Raphael,B. and Smith,I.F.C. (2008) A comprehensive validity index for clustering. *Intell. Data Anal.*, **12**, 529–548.

Received: January 28, 2024. **Revised:** October 30, 2024. **Editorial Decision:** November 3, 2024. **Accepted:** November 22, 2024