Estimating polyadenylated tail lengths from direct RNA sequencing signals

Nanopore RNA Consortium

1 Model Description

To estimate the number of bases in the polyadenylated tails of mRNA reads, we developed a predictive model that combines a hidden markov model with an estimator of the translocation rate through the pore. The hidden markov model uses linear-chain state transitions to perform a segmentation of the raw sequencing signal of an mRNA read; the estimated translocation rate is used in conjunction with the segmentation to estimate the tail length, which we elaborate upon below.

In the rest of this supplementary note, we follow Oxford Nanopore Technologies' nomeclature in referring to the sequential raw current measurement values corresponding to a read sequenced via the direct RNA protocol [1] as its *squiggle*, and individual values of a squiggle as *samples*; squiggles are oriented in the direction of time, i.e. in the 3'-to-5' orientation with respect to the strand. We simultaneously refer to a given sequenced mRNA molecule and its sequence of nucleotides as a *read*. Atypically, every mRNA read that we consider below is assumed to be oriented in the 3'-to-5' direction; this is to match the orientation of the direct RNA protocol, which sequences reads in the 3'-to-5' direction.

1.1 Signal Segmentation via Hidden Markov Model

A hidden markov model, which we call the *Segmentation HMM*, is used to segment the squiggle of a read into distinct *regions* appearing sequentially. Biologically, each sequenced read consists of a sequencing adapter (which we call the *leader* region), the RT splint adapter (which we call the *adapter* region), the *polyadenylated tail*, and the coding *transcript*, respectively, from 3' to 5' [1]. The segmentation HMM contains one state for each of these regions connected sequentially via linear chain state transitions. We additionally include two states to handle "jumps" in the squiggle that are due to idiosyncrasies specific to nanopore sequencing, which we explain below.

We assume each state has an associated emission distribution and treat the raw samples of a squiggle as realizations from one of these distributions, dependent on a latent state. For a squiggle $\vec{s} = (s_1, \ldots, s_n)$ with associated latent states $\vec{h} = (h_1, \ldots, h_n)$ — where each h_i is a label representing a region of the read — we have that

$$\forall i: \; s_i \sim p(s|h_i) = \epsilon_i(s),$$

where $\epsilon_i(\cdot)$ is the emission distribution for state h_i. In our HMM, we use Gaussian, Gaussian mixture, and uniform distributions to model emissions. We use the Viterbi algorithm to infer $\vec{h}$ from any given $\vec{s}$.

Prior to running the Viterbi algorithm, we apply a global linear rescaling on all samples of the squiggle to remove per-read variations from the base model. The coefficients of the linear transformation[1] are estimated individually for each read using the same procedure as in [2]. Following [2], we refer to a segmentation of a squiggle $\vec{s}$ into a sequence

$$\vec{e} = (\langle \mu_1, \sigma_1, \delta_1 \rangle, \ldots, \langle \mu_K, \sigma_K, \delta_K \rangle)$$

of contiguous samples (called *events*) as the *event sequence* associated to the squiggle. Samples associated to a single event approximately correspond to a 5-mer residing in the pore at the time of sampling. The event sequence associated to a squiggle is determined by a segmentation algorithm[2] provided by Oxford Nanopore.

[1] The linear rescaling is implemented as a part of the *SquiggleRead* class in nanopolish: https://github.com/jts/nanopolish.

[2] https://github.com/jts/nanopolish/blob/master/src/thirdparty/scrappie/event_detection.c#L268

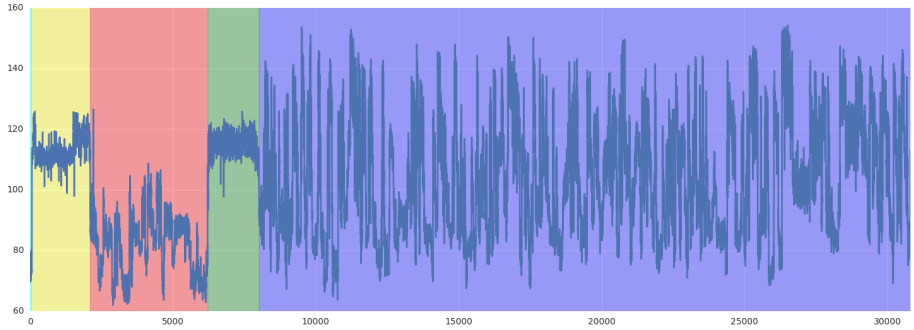

Figure 1: An example of a squiggle segmentation generated by the hidden markov model. Distinct regions, from left to right: *start* (cyan), *leader* (yellow), *adapter* (red), *poly(A) tail* (green), and *transcript* (purple). Two samples flagged as "cliffs" can be observed in the poly(A) tail.

To fit the emission distributions, we use a two-stage bootstrapped procedure where manually-tuned emissions were used in an initial HMM before fitting emissions on the samples of the passing segmentations via maximum likelihood; this is elaborated below in the subsection on the emission distributions. We devote the rest of this section to explaining the state transitions and the emission distributions of the segmentation HMM in further detail.

1.1.1 State Transitions

The hidden states of the Segmentation HMM have the following names (single-letter label in parentheses) and interpretations:

- *START (S)*: an optional state appearing before the *LEADER* segment.
- *LEADER (L)*: the sequencing adapter attached to, and sequenced prior to, the RT splint adapter.
- *ADAPTER (A)*: the RT splint adapter sequence attached to the polyadenylated region as a part of the direct RNA sequencing protocol.
- *POLYA (P)*: the polyadenylated region of a read.
- *CLIFF (C)*: a state that models brief sequencing artifacts within the polyadenylated region.
- *TRANSCRIPT (T)*: the coding sequence of a read.

The states L, A, P, and T are connected via one-way transitions in a linear chain, representing their biologically-expected order of appearance in an mRNA squiggle. *START* is an optional state to account for a short open-pore signal that appears in some reads before the *LEADER* segment. *CLIFF* is a state that models sequencing errors that appear in the *POLYA* region; these are short, sparse regions within the *POLYA* region, occurring for < 10 samples at a time and typically representing $< 1\%$ of the length of the *POLYA* region, that would otherwise cause a mis-segmentation if not modelled. We observed that erroneous 1-sample artifacts of atypically high or low current level caused the segmentation HMM to fail unless we added a *CLIFF* state to model them. As the number of samples in each of the four regions represented by states L, A, P, T is typically fairly large — on the order of thousands of raw samples per region — the weight on the self-loop of each state is much higher than that of a transition to the next state. We set the probability of a self-loop for L to 0.9, for A to 0.95, for P to 0.89, and for T to 1.0, since the latter represents the final region of a read in the 3'-to-5' direction. A full diagram of the state transitions is provided in Figure 2.

1.1.2 Emission Distributions

Emissions are modelled with Gaussian, uniform, and Gaussian mixture distributions. The following emission distributions are used:

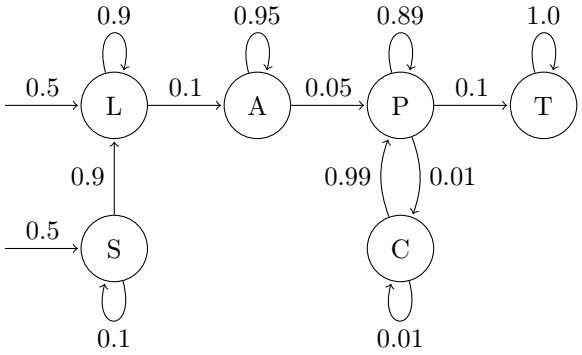

Figure 2: The state transitions of the segmentation HMM. Edges without an origin node on the left indicate the initial state probabilities.

- *START*: $\mathcal{N}(\mu = 70.2737, \sigma^2 = 3.7743)$
- *LEADER*: $\mathcal{N}(\mu = 110.973, \sigma^2 = 5.237)$
- *ADAPTER*: $0.874 \times \mathcal{N}(\mu = 79.347, \sigma^2 = 8.3702) + 0.126 \times \mathcal{N}(\mu = 63.3126, \sigma^2 = 2.7464)$
- *POLYA*: $\mathcal{N}(\mu = 108.883, \sigma^2 = 3.257)$
- *CLIFF*: $\mathcal{U}([70.0, 140.0])$
- *TRANSCRIPT*: $0.346 \times \mathcal{N}(\mu = 79.679, \sigma^2 = 6.966) + 0.654 \times \mathcal{N}(\mu = 105.784, \sigma^2 = 16.022)$

Emission distributions were fitted with a two-stage bootstrapped approach. For each region of the squiggle corresponding to a state, we made an initial estimate of the mean current level and variance of the current levels, and ran the segmentation HMM on each read using these as the parameters of initial emission distributions, before manually filtering the resulting segmentations based on quality. Sample values from each of the S, L, A, P, T regions were aggregated from each of the filtered segmentations, and Gaussians were fitted via maximum likelihood estimation to each squiggle region to obtain the above emission distributions, while each Gaussian mixture was fitted via 100 iterations of expectation-maximization. The number of Gaussian components in each mixture distribution was chosen to be equal to the number of observed peaks in the kernel density estimate of the sample data for each region. The uniform emission distribution for the *CLIFF* state was not fitted with this approach; the upper and lower limits for the uniform distribution were chosen based on manually-tuned observed upper and lower bounds for all samples across all datasets.

1.2 Estimation of the Polyadenylated Tail Length

Fix a read R. Given a segmentation

$$\langle L_0, A_0, P_0, T_0 \rangle$$

of a squiggle

$$\vec{s} = (s_1, \ldots, s_n)$$

with associated events

$$\vec{e} = (\langle \mu_1, \sigma_1, \delta_1 \rangle, \ldots, \langle \mu_K, \sigma_K, \delta_K \rangle),$$

where each component of the segmentation represents the starting index of its respective region — e.g. s_{P_0} is the first sample in the poly(A) tail — we compute an estimate of the number of nucleotides in the poly(A) region by multiplying the duration of time spent in the poly(A) region by the *read rate*, the rate at which the nucleotides of a read translocate through the pore during sequencing. The translocation rate of a read varies as it is being sequenced; hence we instead use the reciprocal of the median event duration as a proxy for a uniform sequence read rate. We found that using the median event duration gave poly(A) tail length estimates that were more robust to read rate differences across different reads than other read-level summary statistics such as the mean event duration.

Our estimator of the polyadenylated tail length is given by

$$\hat{n}_{p(A)} := \frac{|T_0 - P_0|}{\rho \cdot \text{med}(\vec{\delta})} - 5,$$

where:

- $\hat{n}_{p(A)}$ is the estimated number of nucleotides in the polyadenylated region of the read;
- $\text{med}(\vec{\delta}) = \text{med}(\{\delta_i\}_{i=1}^{K})$ is the median event duration from events in the read, in seconds;
- $|T_0 - P_0|$ is the number of samples in the polyadenylated region, as indicated by the segmentation;
- ρ is the sample rate (in $\frac{\text{samples}}{\text{sec}}$) of the nanopore sequencer, i.e. the number of current level samples observed per second; and
- a constant term is subtracted from the quotient term to adjust for the k-mer size associated to the event sequence (in our case, 5).

The sample rate ρ is a fixed constant set by the nanopore sequencer hardware whereas the median event duration differs for each read.

1.3 Reproducibility

The polyadenylated tail length estimator is implemented in the `polya` subprogram of nanopolish:

https://github.com/jts/nanopolish

The analyses performed on the datasets in the accompanying paper may be reproduced by running the associated pipeline, implemented as a Makefile:

https://github.com/paultsw/polya_analysis

2 References

1. Garalde, D., et al. Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods* **15**, pages 201-206 (2018).
2. Simpson, J.T., Workman, R.E., Zuzarte, P.C., David, M., Dursi, L.J., Timp, W. Detecting DNA cytosine methylation using nanopore sequencing. *Nature Methods* **14**, pages 407-410 (2017).