*Applied Comparative Genomics*, 2019

|  |
| --- |
| Poly(A)-Tail Length Estimation from long ONT sequencing: Motivation and Tool Comparison  Remy Schwab1,2 and Luchao Qi1,2  1Department of Biomedical Engineering, Johns Hopkins University Whiting School of Engineering  2Team RemChao, Johns Hopkins University  Abstract  **Polyadenylation is a co-transcriptional process in which an adenosine monophosphate molecule is added to the 3’ end of mRNA. These molecules, which consist solely of adenines, are known as Poly(A) tails and their lengths (the number of bases that make up the molecule) are believed to have some important biological implications. Along with the advent of native RNA sequencing, two tools, nanopolish and tailfindr, have proposed methods for estimating the lengths of Poly(A) tails using the Oxford Nanopore platform. Our project aims to compare these tools.** |

# Introduction

Poly(A) tails are necessary for translocation of mRNA to the cytoplasm and directly influence a transcript’s stability and translation. So while the importance of poly(A) tails is somewhat understood, what effect the length of the poly(A) tail has is not. The initial poly(A) tail length after polyadenylation is reported to be about 250nt however this becomes dynamic after it has left the nucleus and the data so far is difficult to unify. For example, even though only a short poly(A) tail is needed to prevent degradation, hyper-adenylated transcripts degrade quickly in the nuclease. Additionally, there has been evidence showing that long tails during embryonic development are positively correlated with translational efficiency but in *C. elegans* shorter tails are shown to be more actively translated.

Early attempts to measure poly(A) tail length were low-throughput and were transcript-specific. High throughput methods enabled transcriptome wide resolution but were severely limited by short read lengths and technical artifacts related to PCR.

The Oxford Nanopore platform is able to sequence mRNA molecules. The long read lengths and library preparation protocol allow for sequencing of the full transcript. However, it is very difficult to sequence repetitive regions with Oxford Nanopore and so two tools have attempted to reconcile this.

# Methods

Nanopolish is a suite of algorithms for analyzing Oxford Nanopore Sequencing data. It is used to “polish” de novo genome assemblies, detect base modifications (m6a, m5c), call variants, and also has a component, nanopolish polya, to estimate poly(A) tail lengths.

This method combines a hidden markov model with an estimator of the translocation rate of the read through the pore. The foundation of the algorithm involves separating the raw “squiggle” into states which correspond to the states that make up the HMM. These states are denoted as follows:

* Start (S): an optional state appearing before the “leader”
* Leader (L): the sequencing adaptor, attached to, and sequenced prior to the RT splint adaptor
* Adaptor (A): RT splint adaptor which is directly attached to the poly(A) tail
* PolyA (P): the polyadenylated region of the read
* Cliff (C): a state to model brief sequencing artifacts within the poly(A) region of the read
* Transcript (T): The coding sequence

Below is a breakdown into these states from an example squiggle:



Figure 1: Example Segmented Squiggle. Note the read is fed through the pore in the 3’-5’ direction so the last (purple) segment is the transcript. The poly(A) tail segment is shown in green.

And below is a figure showing an example of the HMM:

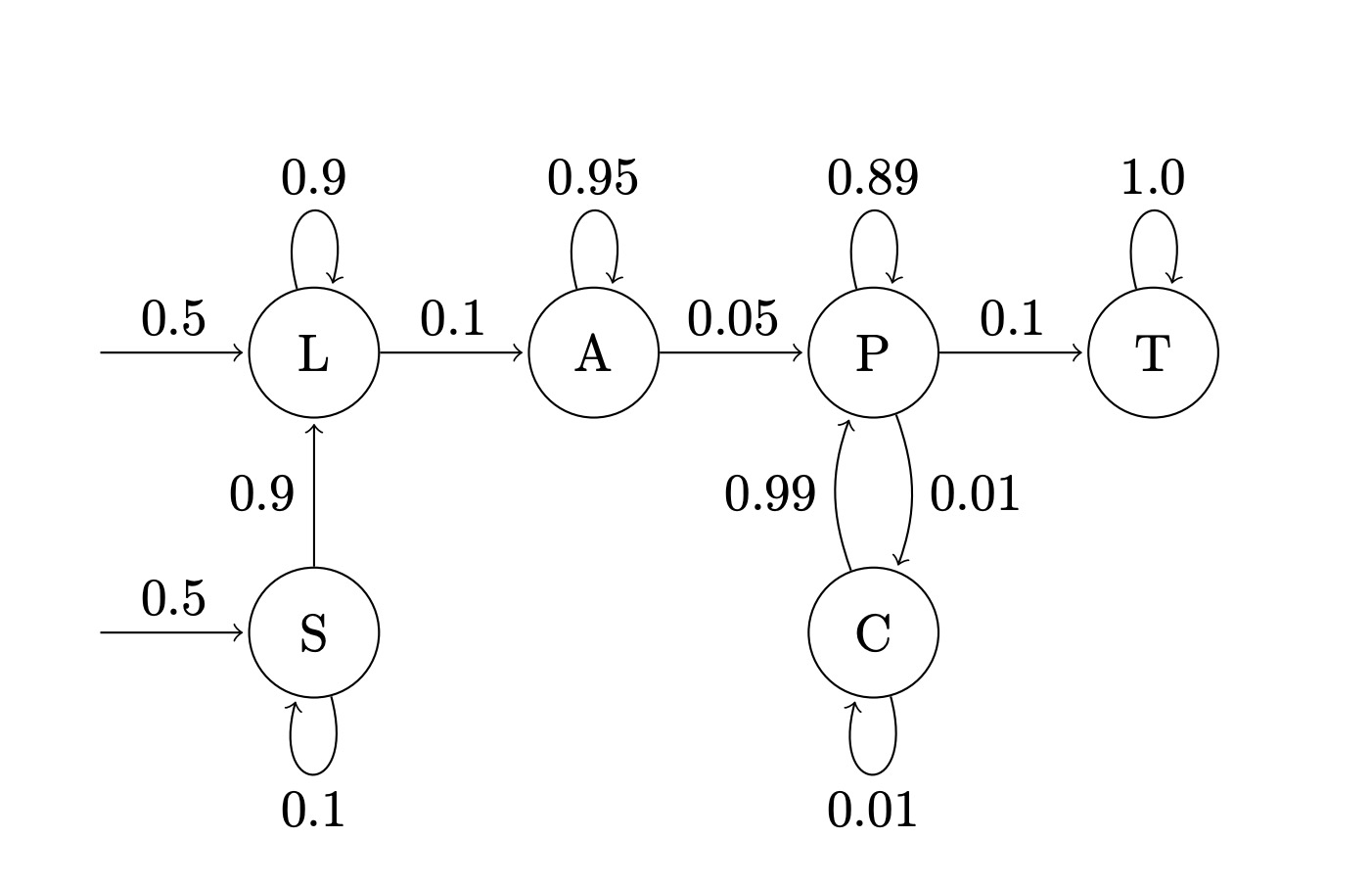


Figure 2:

The tailfindr algorithm works as follows:

* Read fast5 file
* Z-normalize raw data
* Clip signal values exceeding 3 sd
* Smoothen by sliding window
* Define poly(A)-containing segment by thresholding smoothened signal by 0.3 (this defines the rough poly(A) boundaries
* Compute mean of every 25 samples of clipped signal within rough poly(A) boundaries
* Compute slope between every two consecutive points of mean signal
* Get precise poly(A) boundaries by shrinking rough poly(A) boundaries until slope signal is confined in +/- 0.3 range
* Normalize tail length by read specific translocation rate

This is shown in the below figure:

A screenshot of text

Description automatically generated

# Results

The dataset we are trying to apply these tools to is the Oxford Nanopore poly(A) standards (linked [here](https://www.ebi.ac.uk/ena/data/view/PRJEB28423)). However, despite the fact that these files are already available with basecalls, they are not compatible with tailfindr because they require some metadata table. Additionally, basecalling is a very computationally heavy task and requires the proprietary software “albacore” or “guppy”. I am currently looking into a potential software “hack” around this.

# Discussion

Tailfindr did perform a comparison of its tool with nanopolish. It reported the accuracies as comparable. Probably the biggest difference between these tools is that nanopolish does require alignment in order to estimate the tail lengths and tailfindr does not. Nanopolish is also pretty difficult to install.

Acknowledgements

Thank you to Roham Razaghi from the Timp lab!

References

1. Jalkanen AL, Coleman SJ, Wilusz J. Determinants and implications of mRNA poly(A) tail size--does this protein make my tail look big?. *Semin Cell Dev Biol*. 2014;34:24–32. doi:10.1016/j.semcdb.2014.05.018
2. Tailfindr
3. Nanopolish
4. ONT standards dataset
5. Direct RNA Seq