Elsevier Editorial System(tm) for Genomics
Manuscript Draft

Manuscript Number: GENO-D-14-00009R1

Title: SCOPE++: Sequence Classification Of homoPolymer Emissions

Article Type: Methods Paper

Keywords: polyadenylation
transcriptome
hidden markov model

Corresponding Author: Dr. John Karro,

Corresponding Author's Institution: Miami University

First Author: James Morton

Order of Authors: James Morton; Patricia Abrudan; Nathanial Figueroa; Chun Liang, Ph.D.; John Karro

Abstract: mRNA polyadenylation is a poorly understood process critical to gene expression and regulation in eukaryotes. To study this process, it is important to identify poly(A) tails accurately in transcriptome sequencing data and differentiate them from artificial adapter sequences added in the sequencing process -- a problem complicated by the presence of sequencing errors and potential post-transcriptional modifications. To address this problem we have created SCOPE++ to find precise border of homopolymers in raw mRNA sequence reads. Based on a Hidden Markov Model approach, SCOPE++ is developed to accurately identify homopolymers in error-prone EST/cDNA data or RNA-Seq data. In a series of tests, we demonstrate that our tool can precisely identify poly(A) tails with near perfect accuracy at the speed needed for high-throughput applications, providing a valuable resource for polyadenylation research.

Suggested Reviewers: Renyi Liu
renyiliu@ucr.edu

William Ray
ray.29@osu.edu

Shin-Han Shiu
shius@msu.edu

Robin Buell
buell@msu.edu

Opposed Reviewers:

July 3, 2014
Genomics (Elsevier)
RE: Response to reviewers (Manuscript: Sequence Classification of homoPolymer Emissions)

Dr. Quackenbush:

We want to thank the reviewer for the comments on our manuscript, which led to a significant improvement in this paper. We specifically thank them for suggesting the two comparisons (of SCOPE++ versus a basic tool, and of SCOPE++ versus a near-trivial HMM), which, as the reviewer was correct in observing, significantly adds to the overall study.

In the following we address all comments point by point.

Reviewer Comment: "you really need to proof this manuscript much, much better, before it's ready for publication"

Response: We apologize, and very much appreciate the fact that the reviewer could look past this. It looks like I inadvertently uploaded the wrong draft of the paper (complete in substance, but before proofreading / editing was completed). The resubmitted version should be in closer to final form and largely free of such errors.

We will not further address points related to grammar / style / English in this response. In all cases the reviewer was correct and the problems have been fixed.

Addressed in paper: Throughout.

Reviewer Comment: "You could, and should, do a lot better with respect to referencing appropriate concepts. You've provided a reasonable background of literature supporting your contention that caring about detecting homopolymers is appropriate, but barely any references regarding the technical approach. Your likely audience is biological researchers interested in polyadenylation. They won't know what the Viterbi algorithm, or Baum-Welch optimization), and they won't be able to evaluate whether these choices were appropriate."

Response: We have added references to materials on HMMs (the Durbin book, and the classic 1986 Rabiner overview paper), as well as a few words of explanation when first introducing Viterbi and Baum-Welch. Note that we opted to emphasize the Durbin book, as opposed to original papers. We feel that while difficult, Durbin is still easier for more biologically oriented researchers to understand than technical papers on the subject.

Addressed in paper: In the first paragraph of Methods we explicitly refer the interested reader to *Durbin et al.* (as well as mention Rabiner). Following this we have added citations to Durbin as appropriate.

Reviewer Comment: "you do things like compare algorithmic results to human-results, based on an apparent assumption that the human annotation is error-free"

Response: We clearly did not explain this well. (Or, perhaps, did explain it well – and obscured understanding with the grammatical problems.) This is not what we are doing, and I've rewritten the relevant text to try to clarify. But let me provide an alternate explanation here.

We are *not* comparing our tool output to human annotations, but instead using those human annotated sets as the basis for creating simulated benchmark sets. We want to compare different tools, but in order to do that we need a benchmark set where we know the correct annotations: we cannot differentiate a true positive from a false positive if we don't know where the tails actually lie. We were unable to find any existing benchmark sets, so the other option was to test them on simulated sequences. But we wanted to be careful that our model wasn't over-simplified – possibly leaving out unidentified structural characteristics of the sequence which impact tool result quality. Thus we developed our "semi-synthetic" sequences as follows:

1) Take a set of actual sequences and hand-annotate them for poly(A) tails. We had six sets covering three organisms from three different sequencing technologies, and randomly picked 500 sequences from each of the six sets for the hand annotation. (Not fun; thank goodness for the existence of undergraduates.)
2) We next took each sequence and "purified" the annotated tail – meaning that we replaced each non-A character within the putative tail with an A[1]. What we then had was a "semi-synthetic" sequence – a real sequence with a possibly synthetic tail embedded in it. Whether or not the humans made any errors, with the "cleaning" we have introduced a tail which may be biologically meaningless, but should be identified as a poly(A) tail by any search tool. Further, any characteristics of the surrounding sequence that might have an impact on a tool are still present, as that portion was unchanged from the original.
3) From this we could generate simulated sequences with arbitrary (controllable) error that could serve as benchmarks for quality measurement.

This is clearly not perfect, and there is more we could do to increase the accuracy of the model. But we believe that this is a very good first-order approximation, providing us with the ability to test various tools despite the lack of actual benchmark sets.

Addressed in paper: We have rewritten the "Benchmark Sets" section of Methods (trying to describe the above more concisely).

Reviewer Comment: "More detail on the actual HMM would be appreciated. For example, if you have only one "in a homopolymer" internal state, but this state accepts errors, what prevents it from accepting two long homopolymers, separated by a large number (but small proportional to the length of the two homopolymers) of non-homopolymer bases?"

Response: Certain potential errors are likely unavoidable for any method that is based solely on sequence analysis. (Example: if a non-tail A base happens to fall next to the

[1] Half our sets were from the complementary strand, thus containing poly(T) tails – where the appropriate adjustments were made.

poly(A) tail, it is inevitably going to be included as part of the tail – and it seems unlikely that this could even be verified as a mistake through only sequence analysis.) We argue that any tool will suffer from such problems. In the case of closely-neighboring tails in specific, in practice they do not tend to be close enough that SCOPE++ will merge them, thus the amount of error this will introduce is negligible.

Addressed in paper: We have tried to add more details, and explain the above, in what is now the last paragraph of the HMM section.

Reviewer Comment: "What makes the HMM approach more appropriate than a simple regular expression that matches the AA leader and trailer, wrapped around a "no more than N% non-A" match?"

Response: This was a great idea, and descriptions of the results have been added to the paper (paragraph 2 of Results – we included a text description only, as the results plots were uninteresting, and not worth extending the length of the paper to include). Doing this analysis also highlighted the (unsurprising) fact that identifying whether a poly(A) tail is present is fairly easy (even in the presence of obfuscating errors) – but pinning down the end-points is quite difficult. It is for the second that the complexity of the HMM is required.

Addressed in paper: Discussion has been added in what is now the second paragraph of the Results section.

Reviewer Comment: "Someone curious about the generalizability of the approach to other domains might be interested in why the Viterbi algorithm is not proving to be computationally overwhelming, and why Baum-Welch optimization shows no meaningful improvements. A credible argument could be made that this is because the HMM is both trivial, and a poor structural match for the underlying biological phenomenon. I don't believe that this is actually the case here, but, those observations would be consistent with a trivially simple, inappropriate HMM - refuting this wouldn't be a bad thing to do."

There are two points here:

1. It is our belief that Baum-Welch does not help because our preliminary estimation of the HMM parameters are fairly close to the values it will find; spending time on Baum-Welch to improve the estimation is not worth the effort, as there is little improvement to be had. (To put it another way: the parameter estimation problem in this case is too simple to bother with Baum-Welch.) We have added a note to this effect.
2. As suggested, we implemented an alternative HMM model consisting of two states ("poly(A)" and ("non-poly(A)"). As predicted, it failed. The tool had a sensitivity below 0.1. We have added a new paragraph explaining this, and drawing the conclusion that our HMM must be reflecting the biological structure in some way that the basic HMM fails to do. (We again thank the reviewer for this suggestion; it really does add something to the paper.)

Addressed in paper: The first point is addressed in the third paragraph of the "Parameter Estimation" section. The second point is covered in what is now the third paragraph of **Results**.

Reviewer Comment: "Can the approach be extended to arbitrary (non homogenous) tandem repeats?"

Response: I've added a discussion to the conclusion, but I am hesitant to say too much without further study. Attempting to identify complex sequences ("complex" in the information-theoretic sense) could add enough complexity to the HMM to drive up the runtime to undesirable levels. However, searching for low-complexity repeated sequences (e.g. satellite sequences such as those that cause Huntington's disease) might be more in the range of what the approach can cover. This could be worth following up on, but is beyond the scope of what we are presenting here.

Addressed in paper: We have added a paragraph to the conclusion in which this is discussed.

We believe this covers all points made by the reviewer, and look forward to hearing from you on this paper.

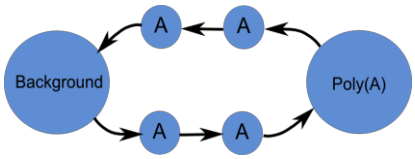
Yours Truly,

Dr. John Karro
Associate Professor, Miami University
Department of Computer Science and Software Engineering, Department of Microbiology, Department of Statistics
karroje@miamiOH.edu

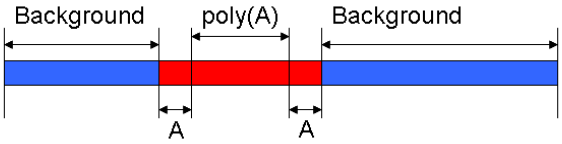
Highlights:

- Tool for collecting data necessary for study of alternative polyadenylation.
- First tool capable of annotating both ends of poly(A) tails in mRNA.
- HMM-base approached allowing for data-specific training.
- Near-perfect accuracy at speeds appropriate for high-throughput applications.

Figure(s)

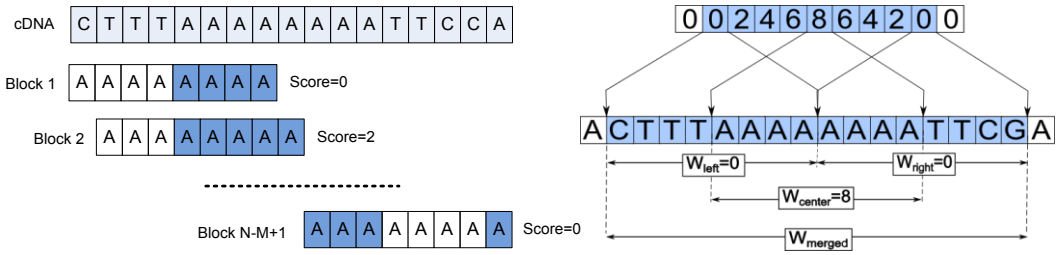


(a)

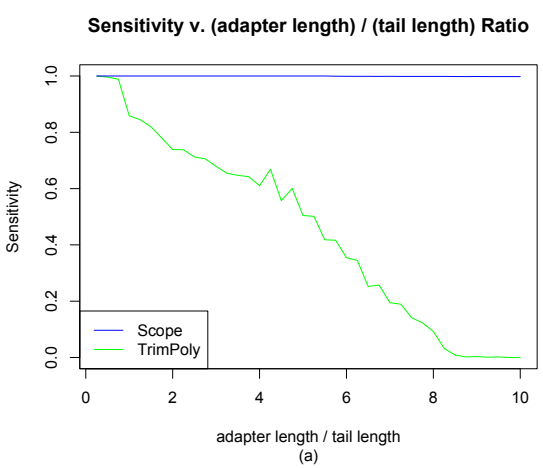


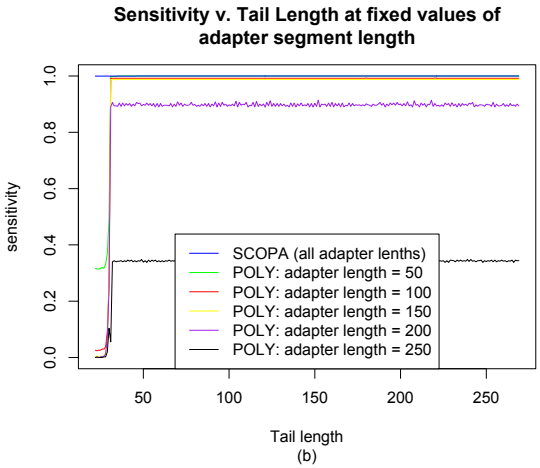
(b)

Figure(s)

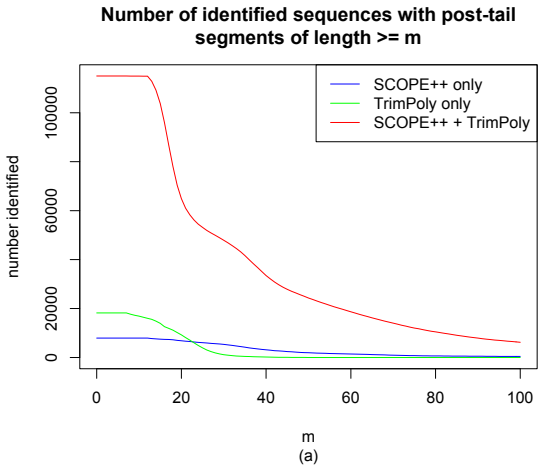

cDNA
C T T T A A A A A A A A T T C C A
Block 1
A A A A A A A A
Score=0
Block 2
A A A A A A A A
Score=2
Block N-M+1
A A A A A A A A
Score=0
0 0 2 4 6 8 6 4 2 0 0
A C T T T A A A A A A A T T C G A
W_left=0
W_right=0
W_center=8
W_merged

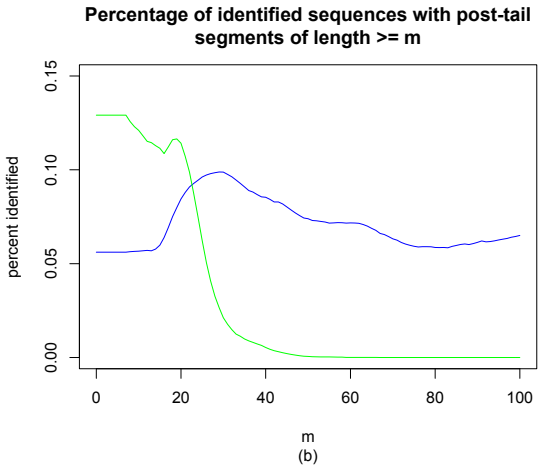
Figure(s)


Sensitivity v. (adapter length) / (tail length) Ratio
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
Scope
TrimPoly
adapter length / tail length
(a)


Sensitivity v. Tail Length at fixed values of adapter segment length
sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
50
100
150
200
250
SCOPA (all adapter lenths)
POLY: adapter length = 50
POLY: adapter length = 100
POLY: adapter length = 150
POLY: adapter length = 200
POLY: adapter length = 250
Tail length
(b)

Figure(s)


Number of identified sequences with post-tail segments of length >= m
SCOPE++ only
TrimPoly only
SCOPE++ + TrimPoly
number identified
0
20000
60000
100000
0
20
40
60
80
100
m
(a)


Percentage of identified sequences with post-tail segments of length >= m
percent identified
0.00
0.05
0.10
0.15
0
20
40
60
80
100
m
(b)

July 3, 2014

Genomics (Elsevier)
RE: Full paper submission

Dr. Quackenbush:

We are re-submitting the attached manuscript, "SCOPE++: Sequence Classification of homoPolymer Emissions", for consideration in Genomics. Based on reviewer comments we have made the requested revisions (see the submitted **Response for Reviewers** file) and hope this paper is now considered worth publication in your journals.

SCOPE++ is a high-throughput computational tool designed to aid analysis of RNA-Seq or EST sequence reads in the study of the role of polyadenylation in gene regulation and expression. It is becoming apparent that both *alternative polyadenylation* and poly(A) tail length variation have critical roles in gene expression and regulation, but collecting data to study the effect is difficult given the dearth of tools capable of determining the precise location, purity and length of poly(A) tails in raw sequence reads. While there are tools that will clean raw sequence reads of sequencing adapter/linker sequences and poly(A) tails (e.g. yours SeqClean package), those tools serve a different purpose: trimming, as opposed to specific data collection. Existing tools cannot be used to infer the needed information about precise poly(A) tail position, length and purity. Hence they are useless in polyadenylation studies. In particular, polyadenylation, RNA editing, non-templated nucleotide addition and other relevant RNA processing procedures, procedures about which we have no clear understanding yet, might change the purity of poly(A) tails. On the other hand, the sequencing errors will definitely cause mismatches in poly(A) tail detection. Consequently, accurate identification of poly(A) tails presents a challenge in bioinformatics.

The included manuscript briefly discusses the underlying machine learning algorithm, the quality of tool results, and a comparison between SCOPE++ and other popular tools. We have also created a new benchmark set of human-annotated sequence reads, and after testing each tool we conclude that SCOPE++ is better able to collect the necessary information. SCOPE++ also uniquely employs machine-learning techniques that allow it to optimize its own parameters towards a particular dataset or new sequencing technology.

SCOPE++ is an open source C++ project. Currently, its executable, user manuals and benchmark testing datasets are available on:

http://code.google.com/p/scopeplusplus

Once our manuscript is accepted for publication, we will release its source code at this website, and are committed to maintaining it for the for-seeable future.

Potential reviewers for the paper include:

1. Renyi Liu <renyi.liu@ucr.edu> University California, Riverside, USA
2. William C. Ray <ray.29@osu.edu> Nationwide Children's Hospital, Ohio, USA
3. Shin-Han Shiu <shius@msu.edu> Michigan State University, USA
4. Robin Buell <buell@msu.edu> Michigan State University, USA

5. Baohong Zhang <zhangb@ecu.edu> East Carolina University, North Carolina, USA

If needed, I can be contacted at karroje@miamiOH.edu, and otherwise I am looking forward to hearing from you about our paper.

Yours Truly,

John Karro
Miami University
Oxford, OH USA

Table(s)

Click here to download Table(s): table1.docx

Data Set Source:			Illumina Arabid-opsis		454 Chalamy		Sanger Human		Average
			5' Strand	3' Strand	5' Strand	3' Strand	5' Strand	3' Strand	
Simulated Error Rate: 0.01 errors / base	Sensitivity	SCOPE++	0.956	1.000	1.000	1.000	1.000	0.996	**0.99**
		TrimPoly	1.000	1.000	1.000	1.000	1.000	1.000	**1.00**
		TrimEst	0.857	0.833	0.975	0.064	0.915	0.822	**0.74**
	Specificity	SCOPE++	0.999	0.945	1.000	1.000	0.992	0.999	**0.99**
		TrimPoly	0.978	0.955	1.000	1.000	0.982	0.998	**0.99**
		TrimEst	0.629	0.699	0.892	0.910	0.506	0.772	**0.73**
	% Correctly Trimmed	SCOPE++	0.947	0.787	0.942	0.659	0.939	0.949	**0.87**
		TrimPoly	0.425	0.164	0.130	0.002	0.181	0.848	**0.29**
		TrimEst	0.290	0.086	0.110	0.030	0.106	0.420	**0.17**
	Average Sum-of-Squares Error	SCOPE++	5	121	5	32	5	6	**29**
		TrimPoly	32	119	201	1101	200	396	**341**
		TrimEst	1112	673	905	1487	5359	13960	**3916**
Simulated Error Rate: 0.03 errors / base	Sensitivity	SCOPE++	0.913	0.999	1.000	1.000	1.000	0.994	**0.98**
		TrimPoly	1.000	1.000	1.000	1.000	1.000	1.000	**1.00**
		TrimEst	0.856	0.832	0.975	0.064	0.915	0.822	**0.74**
	Specificity	SCOPE++	0.999	0.945	1.000	1.000	0.992	0.999	**0.99**
		TrimPoly	0.977	0.955	0.999	1.000	0.982	0.998	**0.99**
		TrimEst	0.625	0.700	0.891	0.909	0.506	0.772	**0.73**
	% Correctly Trimmed	SCOPE++	0.848	0.693	0.832	0.570	0.831	0.854	**0.77**
		TrimPoly	0.409	0.160	0.128	0.002	0.174	0.812	**0.28**
		TrimEst	0.270	0.081	0.107	0.028	0.099	0.387	**0.16**
	Average Sum-of-Squares Error	SCOPE++	6	100	6	28	7	8	**25**
		TrimPoly	30	117	199	1096	197	320	**326**
		TrimEst	1082	670	899	1475	5309	13208	**3773**

Table(s)

Click here to download Table(s): table2.docx

Trim type	Number identified	Percent identified
[TTTT.....]:	45577831	53.1196%
[.....TTTT]:	679815	0.792303%
[AAAA.....]:	15734	0.0183375%
[.....AAAA]:	83200	0.096967%
[TTTT.....TTTT]:	2864767	3.3388%
[AAAA.....AAAA]:	340	0.000396259%
[TTTT.....AAAA]:	2733703	3.18605%
[AAAA…….....TTTT]:	1059	0.00123423%
No homopolymers	33845914	39.44636558%

SCOPE++: Sequence Classification Of homoPolymer Emissions

James T. Morton[1], Patricia Abrudan[2], Nathanial Figueroa[1], and Chun Liang[1,2] §, John E. Karro[1,3,4] §

[1]Department of [1]Computer Science and Software Engineering, [2]Biology, [3]Microbiology, and [4]Statistics, Miami University, Oxford, Ohio, USA.

[§]Corresponding authors

Email addresses:

JTM: mortonjt@miamiOH.edu

PA: abrudapa@miamiOH.edu

NF: figuernd@miamiOH.edu

CL: liangc@miamiOH.edu

JEK: karroje@miamiOH.edu

Abstract

Background: mRNA polyadenylation, the addition of a poly(A) tail to the 3'-end of pre-mRNA, is a process critical to gene expression and regulation in eukaryotes. To understand the molecular mechanisms governing polyadenylation and other relevant biological processes, it is important to identify these poly(A) tails accurately in transcriptome sequencing data and differentiate them from artificial adapter sequences added in the sequencing process. But the annotation of these tails is complicated by the presence of sequencing errors and post-transcriptional modifications. While determining that a tail is present in a given transcript fragment is straight-forward, these obfuscations make the problem of boundary identification a challenge; conventional seed-and-extend algorithms struggle to accurately identify these poly(A) tail end-points. Further, all existing tools that we are aware of focus exclusively on the trimming of poly(A) tails, failing to provide the detailed information needed for studying the polyadenylation process.

Results: We have created SCOPE++, a tool for finding the precise border of poly(A) tails and other homopolymers in raw mRNA sequence reads. Based on a Hidden Markov Model (HMM) approach, SCOPE++ accurately identifies specific homopolymer sequences in error-prone EST/cDNA data or RNA-Seq data at a speed appropriate for large sequence sets.

Conclusions: We demonstrate that our tool can precisely identify poly(A) tails with near perfect accuracy at the speed required for high-throughput applications, providing a valuable resource for polyadenylation research.

Background

Alternative polyadenylation (APA) and *poly(A) tail length variability* have recently been identified as critical mechanisms in gene expression and regulation [1-6]. But conducting studies on their exact role in the regulation process is complicated by the challenge of collecting precise information on the presence and characteristics of poly(A) tails in transcriptome data. While finding a significantly long stretch of adenine bases in a transcript sequence is not difficult, the challenge deepens when you try to account for sequence modifications that could obscure the tail sequence purity (e.g. base-call errors, the effect of processes such as RNA editing, or sequencing artifacts). Tools such as SeqClean [7], TrimEst [8], or SeqTrim [9] are able to effectively remove poly(A) tails via truncation, but cannot recover the detailed information needed when studying issues related to length variation. Thus a tool is

required that is able to identify poly(A) tail boundaries and length, and is robust to disruptions in the homopolymer sequence.

Polyadenylation is a post-transcriptional process in which the 3'-end of a pre-mRNA is cleaved and replaced with a poly(A) tail to form a mature mRNA. Specifically, the polyadenylation protein complex binds to poly(A) signals, then cleaves the sequence at a *poly(A) site*, and finally collaborates with the poly(A) polymerase to perform non-templated adenine addition a few bases downstream of the appropriate poly(A) signals [1]. The poly(A) tail at the 3' end of the mRNA is the hallmark of mRNA maturation, and also serves as a regulatory signal that is critical for mRNA nucleus-to-cytoplasm transportation, mRNA stability, and protein translation [1,2]. Recent research suggests that many eukaryotic genes employ *alternative polyadenylation* (APA), in which multiple distinctive poly(A) sites are utilized to create different transcript isoforms from the same genes [1-3]. It is clear that APA is an important regulator in eukaryotic gene expression and regulation. For example, 3'-UTR shortening by APA appears to be highly active in cancer cells [4]. To increase our understanding of the underlying molecular and biological mechanisms governing polyadenylation and other relevant processes, RNA-seq data is continually being generated to aid in annotating the junctions of the 3'-UTR and the poly(A) tail [5,10]. Moreover, 3'-end tagging (i.e., addition of non-templated U or C/U-rich tags) and 3'-oligouridylation (i.e., poly(U) tails) have been shown to affect mRNA degradation and are common in many eukaryotic species [6,11]. Studies have also demonstrated that the length of the poly(A) tail has a direct effect on mRNA stability, and that mRNAs with short poly(A) tails can be stored in cytoplasm and reactivated later for translation by a re-polyadenylation process that elongates the tail lengths [12,13].

We would like to extract data on poly(A) tail characteristics from the aforementioned RNA-seq data that is so plentiful. However, the precise identification of poly(A) tails embedded within those sequences is a challenge, as the search for a contiguous sequence of adenine bases is complicated by the potential obfuscation of the sequence pattern by base-call errors and the presence of sequencing-induced artificial sequences (e.g., adapters, linkers and primers) added near the poly(A) tails. Such modifications, as well as changes resulting from poorly understood biological processes (e.g. RNA editing and non-templated nucleotide addition [14,15]), can disguise the characteristic adenine sequence and result in impurity in the poly(A) tails. Certain tools, such as SeqClean [7], TrimEST [8], and SeqTrim [9], are able to reliably remove such tails by identifying one end of the tail and truncating it. But they are not able to provide the information needed to study the polyadenylation process itself (e.g. length, or termination position).

In this paper we introduce SCOPE++, an open-source software tool employing a Hidden Markov Model approach for the precise identification of the boundaries and length of poly(A) tails and other homopolymers in sequence reads. SCOPE++ runs at a speed appropriate for Next Generation Sequencing output sizes, with a capability to self-tailor its computational model to the characteristics of a given dataset through the use of machine learning algorithms. This makes it possible to precisely study poly(A) tail length and boundaries, and their roles in regulating gene expression. In particular, our tool is designed to accurately detect poly(A) tails of lower purity, where tail boundaries will be difficult to identify using conventional algorithms.

Methods

SCOPE++ identifies poly(A) tails through the alignment of sequence reads to a predefined Hidden Markov Model (HMM) topology using the Viterbi algorithm. (For the interested reader unfamiliar with HMMs, the Viterbi algorithm, or Baum-Welch Training, a useful bioinformatics-oriented overview is presented in *Durbin et al*. [16], with a more technical review in *Rabiner* [17]). Employing sliding windows to initialize HMM parameter values tailored to the dataset, SCOPE++ utilizes the Viterbi algorithm to approximate the most likely position of a poly(A) tail within any given fragment. [16] We also note that SCOPE++ can search for both poly(A) tails and poly(T) tails, but as the poly(T) version is the mirror of the poly(A) version we will not discuss it further until Results.

HMM Topology

SCOPE++ identifies poly(A) tails using a variable-state Hidden Markov Model [16] conforming to the topology illustrated in Figure 1(a), allowing for the identification of both perfect and imperfect poly(A) tails in raw sequence reads. Using sliding windows along a random sampling of the input as training data for initial parameter values (optionally coupled with Baum-Welch training for HMM parameter optimization), SCOPE++ utilizes the Viterbi algorithm to approximate the most likely position of a poly(A) tail within any given fragment. In Figure 1(a) we see the HMM topology, and in Figure 1(b) the decomposition of a fragment. In the alignment of the fragment each base will be assigned to either: the poly(A) state (thus indicating it is a member of a poly(A) tail); a background state (thus indicating it is not); or one of the intermediate A states. These middle states model bases on the tail boundary, which we require to be error free. Increasing the number of such these boundary can increase the precision of boundary identification, at the cost of sensitivity as base-call errors

become more likely to appear within the defined end region. Experiments indicate that using four such states (split two and two) achieves a reasonable balance.

The topology of the HMM is fairly simple, but requires the fragment conform to a template defined by the model. Specifically, any embedded tail must have x error-free adenine bases at either end (where $x=2$ in Figure 1(a)), and the tolerance of non-adenine tail bases will be dictated by probabilities assigned in the poly(A) state. There is the possibility of a non-tail adenine sitting by chance adjacent to the tail and thus being incorrectly annotated as part of that tail. However, there is virtually no way to distinguish such an aberrant base using only sequence information (or even establishing whether or not there actually an error), hence any tool based on sequence analysis will likely suffer from this. Similarly, two homopolymers sitting in close proximity could be incorrectly merged (with the intervening bases labeled as errors within the tail). However, the occurrence of several consecutive non-A bases will result in a poor fit to the model, and hence are unlikely to be accepted. In practice, the probability of two homopolymers actually occurring close enough to each other be a problem appears to be very small.

Parameter Estimation

Starting with our fixed HMM topology and a set of fragments, we needed to estimate several parameters for the HMM which may be dependent on the characteristics of the biological data or the sequencing processes. Specifically, we needed transition probabilities (the probability of a given base being of one state given the proceeding base was of the other), and emission probabilities (the base distribution for each fragment type). Note that the probabilities involving the end-segment states (the "A" states in Figure 1) are fixed to ensure the assignment of

sequential As to the sequence of states (as these are intended to model the ends of a tail without impurities).

Given a large set of input data and the fixed HMM topology, we estimate the remaining parameters by taking a sampling of the fragments, quickly determining the approximate location of the poly(A) tails, and using that data for the basis of the estimation. Specifically: we apply a sliding-window scoring filter of user-specified width M to a random sampling of the input data. Each window is scored by calculating the difference in the number of adenine and non-adenine bases. As illustrated in Figure 2(a), we identify a window W_{Center} with the highest such score and, if the score is larger than a pre-determined threshold, we treat this as a potential subsequence of a poly(A) tail. We then find the first window to the left, W_{Left}, whose score is equal to the threshold value, and identify a W_{Right} in a symmetric fashion. Finally, we merge together all windows from W_{Left} to W_{Right} to form one contiguous sequence representing a putative poly(A) tail to be used for parameter training (see Figure 2(b)). We note that this method intentionally tends towards the over-estimation of the actual boundaries, allowing us to compensate for the presence of base-call errors at the ends of the poly(A) tail. From these putative sets we can then estimate the HMM parameters: transition probabilities are based on mean tail length with the assumption a geometrical distribution, while emission probabilities are sampled directly from the contents of the putative fragments.

Following this, estimates can be further refined using the Baum-Welch algorithm [16], which is designed to optimally fit an HMM model to a given training dataset. In practice, we find that use of Baum-Welch increases runtime with a negligible improvement in results. It appears that our initial estimate of the model

parameters are fairly close to the estimates returned by Baum-Welch, hence the algorithm is not worth the extra time.

Once we have HMM parameter estimates, SCOPE++ independently applies the Viterbi algorithm [16] to each sequence, getting back the best fit of the sequence to our model (and thus an assignment of each base to one of the characterizing states). While the Viterbi algorithm generally takes $O(mn^2)$ time (where m is the length of the sequence and n is the number of states in the model), the structure of our model allows us to reduce that to $O(mn)$ time. As the number of states employed is fixed, in practice the runtime is linear with respect to the size of the sequence fragment.

Benchmark Sets

There is a lack of beachmark sets on which to test our tools; there has been some work on experimentally identifying tails [18,19], but the information produced by these wet-lab experiments describes tail length *distribution*, not the individual tail positions needed for testing. Hence we have developed a "semi-synthetic" benchmark set: namely, we have taken a set of real sequences with human-identified tails and artificially cleaned the tails of all impurities (that is: we convert all non-adenine bases to adenine). The result is a set of synthetic sequences with characteristics highly correlated to real data. While the human annotation is subject to human error, by cleaning them of impurities we have essentially turned the human-identified tails into simulated tails that we would expect a tool to identify. The non-tail portions of our synthetic fragments are direct copies of actual biological sequences, hence containing within them sequence characteristics that might effect the performance of the tool. We can then introduce simulated base-call error at a controlled rate, providing large datasets with known tail locations on which to test and compare different tools.

The human-annotated data used to generate these sequences can be found in https://github.com/mortonjt/SCOPE github repository.

Results

In assessing the quality of SCOPE++ results we look at three metrics: its ability to correctly identify poly(A) tails (sensitivity), its ability to avoid incorrect identification of tails (specificity), and its ability to find the tail boundaries (precision). We assess sensitivity with the standard formula, the ratio of true positives to actual positives. To assess a tool's precision, we measure result quality in three ways: *% correct*, *average sum of squares trim error*. *% correct* is the fraction of tails with correctly identified end points. *Trim error* is the signed distance between estimated and actual boundaries. Average *sum of squares time error* is the average of the square of the trim errors over all sequences. We the last, as opposed to straight trim error, as it reflects the fact that the seriousness of boundary error increases super-linearly with the error (e.g. being off by four bases is more than twice as problematic as being off by two bases), given the effect of such error on downstream analysis.

First we test whether the complexity of an HMM approach is warranted (given the apparently simple nature of the problem) by comparing SCOPE++ against a basic string search algorithm. Using our semi-synthetic dataset (see Methods), we compare against an algorithm that searches for maximal substrings s such that: (1) s is no shorter than a fixed value m, and (2) the fraction of non-A bases in s is no greater than a fixed value p. We find that while this basic algorithm can effectively detect the presence of poly(A) tails, it cannot match SCOPE++ in boundary detection accuracy. Experiments indicate that an effective parameter assignment is $m=10$ and $p=0.15$, and with those values we find both tools have near-perfect sensitivity (and specificity) for data subjected to a base-call error rate ranging from 0 to 0.1 errors per base. But

the simple algorithm suffers in boundary precision. With a 0% base-call rate (i.e. perfect tails), the simple tool is able to correctly identify boundaries less than 25% of the time, and over-extends the boundaries by an average of 8 bases; under these conditions SCOPE++ returns perfect results. Bumping the error rate improves the simple tail results slightly, as more errors in the tail prevent over-extension. But even at a 5% error rate the tool can only identify 34% of the boundaries correctly, and it over-extends by an average of 4.5 bases. SCOPE++ correctly identifies end points 73% of the time, and those boundaries it misses are short by an average of 0.6 bases. In short: the simple algorithm is useful if we are merely interested in determining if a poly(A) tail is present, but the complexity of SCOPE++ is required if precision is important.

Next we test whether the HMM is an appropriate structural match for the biologically-dictated structure of the sequences. To test this we compare SCOPE++ as described (using $x=2$, thus requiring each tail start and end with 2 As) against a basic two state HMM (containing a "poly(A)" state and a "non-poly(A)" state). Using the same training procedure for the simplified HMM that we use for SCOPE++ (see Methods), the tool tends to assign all bases to the non-poly(A) state. The simplified HMM is able to identify only about 10% of the fragments having tails (as opposed to SCOPE++'s 100%), even when dealing with perfect tails: while tail identification is simple, it is not that simple. We conclude from this that the complexity of our model is in some way describing the appropriate biologically-dictated sequence structure – at least beyond what the trivial HMM can describe.

Having justified the idea behind the general approach, we now move to a comparison against existing tools. For this we looked at the TrimPoly module of SeqClean [7] and TrimEST [8]; SeqTrim [9] was not used due to its slower runtime.

Table 1 displays the average sensitivity and specificity of SCOPE++, TrimPoly [7], and TrimEST [8] over the six different semi-synthetic data sets (described in Methods). For each simulation we start with our benchmark dataset of 500 "cleaned" sequences, and for each sequence we randomly generated 200 simulated sequences by stochastically introducing error into the poly(A) tail, as well as 200 sequences without tails (formed by randomly sampling from the non-tail portion of the sequence). Altogether, this gives us 10,000 tail-containing simulated fragments derived directly from actual fragments and 10,000 tail-omitted fragments based on similar base distributions. Using this set, we observe a almost perfect sensitivity and specificity in both the SCOPE++ and TrimPoly tools.

However, when looking at the *precision* at which boundaries can be identified, we see a different story. We find that SCOPE++ identifies the correct boundaries in a significantly higher number of tails than the other tools (using the 3% simulated error rate, SCOPE++ correctly identified 77% of the sequence boundaries, as opposed to 28% for TrimPoly), and has a significantly smaller average sum-of-squares error rate. In short, SCOPE++ is considerably more precise.

We also look at the performances of SCOPE++ and TrimPoly as functions of both tail length and the length of adjacent sequencing-adapter sequence (a portion of the fragment added downstream of the tail as an artifact of the sequencing process). We find that, while TrimPoly performs better for identifying very short poly(A) tails (< 20 bp), it is highly sensitive to the length of any adapter remaining on the fragment; SCOPE++ is completely robust to such interference. By augmenting the length of tails or adjacent sequencing-adapter fragments of real data in simulation, we can examine the effects of tail length and adapter length on sensitivity (see Figure 3(a)). We observe a significant decrease in the sensitivity of TrimPoly as the adapter

length grows beyond the length of the tail – a factor having no effect on the sensitivity of SCOPE++. In Figure 3(b) we plot sensitivity as a function of tail length for fixed adapter segment length. Once again, while SCOPE++ can handle any such values, TrimPoly suffers in the presence of adapter segments longer than 250 bp – falling to a sensitivity of less than 0.4. In Figure 4 we verify this assertion with real data [5]. Figure 4(a) shows the number of fragments identified as containing a tail as a function of adapter length (with adapter lengths for TrimPoly identified sequences determined by an ad-hoc post processing scan), while in Figure 4(b) we see the same information as a percentage of the total identified (i.e. the sensitivity). The findings are consistent with the simulation: as the tails shift deeper into the fragment, the relative ability of TrimPoly to identify those tails diminishes significantly.

Large Dataset Validation

In order to provide some validation on actual data, we set SCOPE++ to find poly(T) tails and ran it against a 17 GB Arabidopsis dataset [5]. Developed using poly(T) tag sequencing, it has been estimated that about 60% of the reads within this set contain poly(T) tails; our tool discovered them in 59.6% of all reads; see Table 2 for details n the results. A quick search of the dataset for sequences containing short homopolymers returns only 126 clear false negatives, all of which were present in low quality regions and probably would have been discarded anyway.

Availability and Requirements

Our software can be found at https://github.com/mortonjt/SCOPE, with code to be distributed as open source. SCOPE++ was implemented in C++11, and has been tested using the gnu g++ compiler (v. 4.7) on both OS X and Ubuntu Linux.

Conclusions

Here we have presented SCOPE++, a novel approach for identifying imperfect homopolymers when end-point precision is important. We have compared SCOPE++

to simpler tools, and in doing so established the necessity of addressing a seemly simple problem with a more sophisticated solution than seems natural. Unlike other tools, SCOPE++ is able to identify end boundaries at both ends of a tail (as opposed to simple trimming at the inside end) and is able to train on data-specific model parameters. Our tests have shown that the tool performs on par with existing poly(A) trimming tools in terms of speed, while showing considerably more precision in terms of identifying end-point boundaries and sensitivity in identifying tails buried further away from the fragment ends.

The success of SCOPE++ also indicates some potential for generalizing the approach to related problems, such as the identification of low-complexity repeats or simple artifacts that might need to be weeded out of a sequence in a precise manner. While HMM approaches are frequently time consuming and can be over-kill for this sort of problem, in cases where we can keep the complexity of the model fairly low and the number of states small, the same techniques might work while keeping the runtime within reason.

Authors' contributions

JK and LC designed, coordinated and managed the whole project. JM designed, developed and implemented most of the software algorithms. PA conducted human validation on the tests. NF developed software for finding the optimal parameters of SCOPE++. JK contributed to the algorithm design and conducted all of the simulation tests and performed a statistical analysis of the results. LC contributed to software testing and conducted human validation on the results. All authors read and approved the final manuscript.

Acknowledgements

Funding: This project was funded partially by the National Science Foundation (No. O953215 to JK) and the NIH-AREA (1R15GM94732-1 A1 to CL).

Data deposition: The sequence reported in this paper has been deposited in the National Center for Biotechnology Information Short Reads Archive (accession no. SRA028410).

Figures

Figure 1: (Left) A generalization of the HMM topology used for identification. The number of A states between the background and poly(A) states can vary depending on user priorities, with a larger number of states leading to more precise boundary identification but loss of overall sensitivity. Using a total of four states (split two and two) achieves a reasonable balance in practice. (Right) Example of a sequence read containing an identified poly(A) tail that has been divided into four types of segment: background (bases not part of the poly(A) tail), upstream tail-ends (tail bases within *x* of the upstream tail boundary), interior tail bases (further than at least *x* bases away from either boundary), and down-stream tail-ends.

Figure 2: The sliding window scoring scheme used to provide an estimate of the HMM parameters. (a) Illustration of score calculation. (b) Finding W_{center}.

Figure 3: (a) Plot of tool sensitivity as a function of the ratio of adapter segment length to tail length: while SCOPE++ is robust to this ratio, TrimPoly sensitivity deteriorates as the length of the adapter segment becomes longer than that of the tail.

Figure 4: Application of tools to the Triticum aestivum ERR125556 fragment set **[20]**. With 140,000 tail-containing sequences identified, (a) shows the number of sequences with adapter segments of length ≥ m that are identified by SCOPE++, TrimPoly, or both. In (b) we see the percentage of those sequences that were identified only by SCOPE++ or TrimPoly.

Table 1: Results of tests on data-derived simulated sets using SCOPE++, the TrimPoly component of SeqClean [7], and TrimEst [8]; SeqTrim [9] was too slow for testing at the necessary scale. Starting with a set of 500 human-annotated sequenced transcript fragments (Illumina-sequenced Arabidopsis, 454-sequences Chalamydomons, and Sanger-sequence Human [21]), we remove all sequencing errors within the human identified poly(A) or poly(T) tails, then introduce simulated errors into those tails to allow for the testing of sensitivity and boundary identification at a controlled error rate. For specificity we generate sequences using the base distribution of the non-tail segments of the sequences in the set. Applying each tool to the set, we then have a reference solution that allows us to compute result qualities.

Table 2: With the SCOPE++ trim option, poly(A) tails and poly(T) tails can be simultaneously trimmed. The table below lists all possible arrangements of poly(A) and poly(T) tails within a single read and the frequencies in which they appear in the actual dataset.

[1] N.J. Proudfoot, Ending the message: poly(A) signals then and now, Genes & Development. 25 (2011) 1770–1782. doi:10.1101/gad.17268411.

[2] D. Xing, Q.Q. Li, Alternative polyadenylation and gene expression regulation in plants, WIREs RNA. 2 (2010) 445–458. doi:10.1002/wrna.59.

[3] D.C. Di Giammartino, K. Nishida, J.L. Manley, Mechanisms and consequences of alternative polyadenylation, Mol. Cell. 43 (2011) 853–866. doi:10.1016/j.molcel.2011.08.017.

[4] C. Mayr, D.P. Bartel, Widespread shortening of 3'UTRs by alternative cleavage and polyadenylation activates oncogenes in cancer cells, Cell. 138 (2009) 673–684. doi:10.1016/j.cell.2009.06.016.

[5] X. Wu, M. Liu, B. Downie, C. Liang, G. Ji, Q.Q. Li, et al., Genome-wide landscape of polyadenylation in Arabidopsis provides evidence for extensive alternative polyadenylation, Proceedings of the National Academy of Sciences. 108 (2011) 12533–12538. doi:10.1073/pnas.1019732108.

[6] Y.S. Choi, W. Patena, A.D. Leavitt, M.T. McManus, Rna. 18 (2012). doi:10.1261/rna.029306.111.

[7] DFCI Gene Indices Software Tool, (n.d.).

[8] P. Rice, I. Longden, A. Bleasby, EMBOSS: the European Molecular Biology Open Software Suite, Trends Genet. 16 (2000) 276–277.

[9] J. Falgueras, A.J. Lara, N. Fernández-Pozo, F.R. Cantón, G. Pérez-Trabado, M.G. Claros, SeqTrim: a high-throughput pipeline for preprocessing any type of sequence reads, BMC Bioinformatics. 11 (2010) 38. doi:10.1186/1471-2105-11-38.

[10] F. Ozsolak, P. Kapranov, S. Foissac, S.W. Kim, E. Fishilevich, A.P. Monaghan, et al., Comprehensive Polyadenylation Site Maps in Yeast and Human Reveal Pervasive Alternative Polyadenylation, Cell. 143 (2010) 1018–1029. doi:10.1016/j.cell.2010.11.020.

[11] I.Y. Morozov, M.X. Caddick, Cytoplasmic mRNA 3' tagging in eukaryotes: does it spell the end? Biochem. Soc. Trans. 40 (2012) 810–814. doi:10.1042/BST20120068.

[12] C. Bonaïti, S. Parayre, F. Irlinger, Novel extraction strategy of ribosomal RNA and genomic DNA from cheese for PCR-based investigations, Int. J. Food Microbiol. 107 (2006) 171–179. doi:10.1016/j.ijfoodmicro.2005.08.028.

[13] R. Mendez, J.D. Richter, Translational control by CPEB: a means to the end, Nat. Rev. Mol. Cell Biol. 2 (2001) 521. doi:10.1038/35080081.

[14] Y.W. Cheng, L.M. Visomirski-Robic, J.M. Gott, Non-templated addition of nucleotides to the 3' end of nascent RNA during RNA editing in Physarum, Embo J. 20 (2001) 1405–1414. doi:10.1093/emboj/20.6.1405.

[15] Y. JIN, Nontemplated nucleotide addition prior to polyadenylation: A comparison of Arabidopsis cDNA and genomic sequences, Rna. 10 (2004) 1695–1697. doi:10.1261/rna.7610404.

[16] R. Durbin, S.R. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis, Cambridge University Press, 1998.

[17] L. Rabiner, B.H. Juang, An introduction to hidden Markov models, ASSP

Magazine, IEEE. 3 (1986) 4–16. doi:10.1109/MASSP.1986.1165342.
[18] A.O. Subtelny, S.W. Eichhorn, G.R. Chen, H. Sive, D.P. Bartel, Poly(A)-tail profiling reveals an embryonic switch in translational control, Nature. 508 (2014) 66–71. doi:10.1038/nature13007.
[19] H. Chang, J. Lim, M. Ha, V.N. Kim, TAIL-seq: genome-wide determination of poly(A) tail length and 3' end modifications, Mol. Cell. 53 (2014) 1044–1052. doi:10.1016/j.molcel.2014.02.007.
[20] GenBank, GENBANK SRA: ERX101738 (Triticum aestivum), Www.Ncbi.Nlm.Nih.Gov/Sra/Term=ERX101738. (n.d.).
[21] C. Liang, Miami University Bioinfo Lab, Bioinfolab.Miamioh.Edu. (n.d.).