**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 lie 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.

We will not further address points related to grammar / style / English in this reviewer response. In all cases the reviewer was correct, the problems have been fixed, and we ran it through a third party for editing for good measure.

**Addressed in paper**: Throughout.

**Reviewer Comment**: “You could, and should, do a lot better with respect to referencing appropriate concepts.”.

**Response**: We have added references to materials on HMM (the Durbin book, and the class 1986 Rabiner overview paper), as well as a few words of explanation when first introducing Viterbi and Baum-Welch.

**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 Dubin as appropriate. Note that we opted to emphasize the Durbin book, as opposed to original papers, as we feel that while difficult, it is still easier for a more biologically oriented research to understand than most of the technical papers on the subject.

**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. We want to compare tools to each other, but in order to do that we need a set where we know the correct answers: we cannot differentiate a true positive from a false positive if we don’t already 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 about our model – incorporating any structural characteristics of the sequence that might be relevant (whether or not we were aware of them). To get this we did the following:

1. Take a set of actual sequences and hand-annotate them for poly(A) tails. We had six sets covering three organisms, and randomly picked 500 from each of the six sets. (Not fun; thank goodness for the existence of undergraduates.)
2. We next too each sequence and “purified” the annotated tail – meaning that we replaced each non-A character with an A.[[1]](#footnote-1) What we know have is a “semi-synthetic” sequence – a real sequence embedded with a probably synthetic tail embedded in it. Whether or not the humans made any errors, this is now essentially a poly(A) tail. It may be biologically meaningless, but there should be no doubt that a search tool should identify it. Further, any characteristics of the surrounding sequence that might screw a tool up are still present, and that portion is unchanged from the original.
3. Now we can introduce into our tails simulated error. For example: we can introduce a controlled base-call error by changing each base in the (artificial) tail with some fixed probability *p*.

As a result: we can generate simulated sequence sets with known “tail” locations and subject to different models of tail obfuscation, and be confident that we have preserved unknown sequence characteristics potentially relevant to tool performance. This is clearly not perfect, and there is more we could do to increase the accuracy of the model. But we argue 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” subjection 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:** We have addressed this in what is now the last paragraph of the HMM Topology section. Certain potential errors are likely unavoidable for any method based solely on sequence analysis. (Another example: if a non-tail A base happens to fall next to the 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 save through wet-lab techniques.) We argue that

**Addressed in paper**:

**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 a description of the results have been added to the paper (paragraph 2 of Results – we included a text description only, as the results plots were so start as to be uninteresting, and not worth extending the length of the paper). This 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. Its our belief that Baum-Welch does not help because our preliminary estimation of the HMM parameters are fairly close to the correct values; spending time on Baum-Welch to improve the estimation is not worth the effort, as there is little improvement to be had. 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)”), and as predicted – it failed. The tool had a sensitivity of below 0.1. We have added a new paragraph explaining this, and drawing the conclusion that the problem is complicated enough to justify the HMM machinery. (We 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 something akin to this to the conclusion. I’m hesitant to say too much without actual study. Attempting to identify complex sequences (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.

**Addressed in paper**: Last paragraph of the conclusion.

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