Improving transcriptome assembly through error correction of high-throughput sequence reads

Matthew D MacManes[1*] and Michael B. Eisen[1,2]

1 UC Berkeley. California Institute of Quantitative Biology, Berkeley, CA, USA
2 Howard Hughes Medical Institute

∗ Corresponding author: macmanes@gmail.com, Twitter: @PeroMHC

Response to Reviewers Comments

Please find the attached paper, which is a revision of manuscript number 2013:04:462:0:0. The manuscript has undergone substantial change in response to comments from Titus, Mick, and Gerard (Editor). My reading of these comments suggested several major deficits, all of which have been addressed. We now use paired-end data, compare our results using simulated data to 'real' data, and make our data publicly available. In addition to these major changes (which required that we redo correction, assembly, and analysis), we also address the other concerns, including justifying our choice of correction methods and data types.

The paper is much stronger as a result of this review process, and I hope you will find the changes satisfactory.

Best, Matt MacManes

1. The use of real versus fake data.

   This is an excellent suggestion. In response, I have done a parallel analysis using a random 30M PE read subset of the Trinity mouse dataset. I choose to use 30M reads rather than the full 50M read dataset to maximize comparability, though moderate differences in coverage still exist.

   With regards to analyses, the paper continues to focus mostly on the simulated data, though I now state that the results of the simulation study have been corroborated by analysis of real data. That the finding are robust to dataset should be interpreted as evidence of general utility of these methods.

2. Use paired end reads.

   I have used 30 million PE reads in this version of the paper. The results are unchanged, though the magnitude of the improvement is reduced. I believe this to be due to the general improvement of the assembly quality secondary to use of PE reads.

3. Deposit the data!.

   The data are now publicly available. The reads are available on my personal Dropbox (https://www.dropbox.com/s/rkl0ihqom28smb2/empiric.reads.tar.gz and https://www.dropbox.com/s/mp8fu0tijox69ki/simulated.reads.tar.gz). They

will be moved to Dryad upon acceptance. The assemblies are available at http://dx.doi.org/10.6084/m9.figshare.725715. The code has been moved to a Github Gist https://gist.github.com/macmanes

4. Justify the use of error correction techniques

   The specific error correction techniques were chosen in an attempt to cover the breadth of computational techniques, as well as to include tools commonly used, and recently benchmarked. I make this explicit in the manuscript. Of note, this version removes the program ECHO from the analysis. Given that it would have taken at least 4 weeks to run with the new PE data and empirically derived dataset, I did not have the time, given the constraints on resubmission.

5. Use more than just Illumina data.

   While I agree that expanding the paper to include different types of sequence data (454 and IonTorrent) would be interesting, it is beyond the scope of this current paper, which focuses on the current (and likely future, given the number of deployed machines) most popular sequencing technology. In addition, that each technology has a unique error model, understanding the results would not be easy, given simulated data not accessible.

6. What's going on with splice variants?

   Good question... As Titus notes in his review, the analysis of splice variants is tricky. I am quite interested in developing these methods, perhaps as part of a larger project (in connection with my work with Trinity). I think that attempting to dig into splicing here would be too much. I do note the issue in the discussion (line 230 –), as a potential issue with error correction.

7. Variable coverage and miscorrections?

   This is a really interesting point, and one I would have expected the developers of the various error correctors to have addressed– They have not. I do see signs of low expressed reads suffering from a less efficient correction (I suspect this is why Seecer did so poorly in my trials). Also, when looking at the number of corrections per read– reads that are miscorrected tended to have more corrections than did appropriately corrected reads. (line 232 –). About 5% of reads are miscorrected. I would expect that number to be reduced with higher coverage.

8. Use other assemblers.

   This is a useful suggestion, and one which I would have loved to do. However, the extremely large amount of work that benchmarking many different assemblers is not feasible given the constraints on time for revisions. I would expect that the general patterns to hold, though the magnitude of difference may vary as some assemblers may be better/worse that Trinity at 'bubble popping'.

9. Do I recommends these methods for genomic data?

   Yes, all (with exception to Seecer) have been developed for genomic reads, and should be even more efficient with those data, given even coverage.

10. Add a pipeline?

    I have included a set of commands that should facilitate correction using Reptile– https://gist.github.com/macmanes/5878728.

11. Why do errors remain?

    Good question– it seems logical to assume that increasing coverage would solve this, but maybe not.. We do see a suggestion that splice isoforms really messes up error correction (and we know their effects on assembly all too well). Also, it may be something to go with low complexity sequence causing spurious matching..

12. Other issues.

    - I have formalized punctuation.
    - changed text to *de Bruijn*
    - Added citations to highlight the effect of errors (line 22)
    - LaTeX tag has been removed.