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

Abstract

The study of functional genomics–particularly in non-model organisms has been dramatically improved over the last few years by use of transcriptomes and RNAseq. While these studies are potentially extremely powerful, a computationally intensive procedure–the *de novo* construction of a reference transcriptome must be completed as a prerequisite to further analyses. The accurate reference is critically important as all downstream steps, including estimating transcript abundance are critically dependent on the construction of an accurate reference. Though a substantial amount of research has been done on assembly, only recently have the pre-assembly procedures been studied in detail. Specifically, several stand-alone error correction modules have been reported on, and while they have shown to be effective in reducing errors at the level of sequencing reads, how error correction impacts assembly accuracy is largely unknown. Here, we show via use of a simulated and empiric dataset, that applying error correction to sequencing reads has significant positive effects on assembly accuracy, and should be applied to all datasets.

Introduction

The popularity of genome enabled biology has increased dramatically, particularly for researchers studying non-model organisms, during the last few years. For many, the primary goal of these works is to better understand the genomic underpinnings of adaptive (Linnen et al., 2013; Narum et al., 2013) or functional (Muñoz Merida et al., 2013; Hsu et al., 2012) traits. While extremely promising, the study of functional genomics in non-model organisms typically requires the generation of a reference transcriptome to which comparisons are made. Although compared to genome assembly (Bradnam et al., 2013; Earl et al., 2011), transcriptome assembly is less challenging, significant hurdles still exist (see Francis et al. (2013); Vijay et al. (2013); Pyrkosz et al. (2013) for examples of the types of challenges).

The process of transcriptome assembly is further complicated by the error-prone nature of high-throughput sequencing reads. With regards to Illumina sequencing, error is distributed non-randomly over the length of the read, with the rate of error increasing from 5' to 3' end (Liu et al., 2012). These errors are overwhelmingly substitution errors (Yang et al., 2013), with the global error rate being between 1% and 3%. While beyond the focus of this paper, the accuracy of *de novo* transcriptome assembly, sequencing errors may have important implications for SNP calling, and the estimation of nucleotide polymorphism and the estimation of transcript abundance.

With regards to assembly, sequencing read error has both technical and 'real-world' importance. Because most transcriptome assemblers use a *de Bruijn* graph representation of sequence connectedness, sequencing error can dramatically increase the size and complexity of the graph, and thus increase both RAM requirements and runtime (Conway and Bromage, 2011; Pell et al., 2012). More important, however, are their effects on assembly accuracy. Before the current work, sequence assemblers were thought to efficiently handle error given sufficient sequence coverage. While this is largely true, sequence error may lead to assembly error at the nucleotide level despite high coverage, and therefore should be corrected, if possible. In addition, there may be technical, biological, or financial reasons why extremely deep coverage may not be possible, therefore, a more general solution is warranted.

While the vast majority of computational genomics research has focused on either assembly (Chaisson et al., 2004; Miller et al., 2010; Earl et al., 2011; Bradnam et al., 2013) or transcript abundance estimation (Soneson and Delorenzi, 2013; Marioni et al., 2008; Mortazavi et al., 2008; Pyrkosz et al., 2013), up until recently, research regarding the dynamics of pre-assembly procedures has largely been missing. However, error correction has become more popular, with several software packages becoming available for error correction– e.g. ALLPATHSLG error correction (Gnerre et al., 2011), QUAKE (Kelley et al., 2010), ECHO (Kao et al., 2011), REPTILE (Yang et al., 2010), SOAP*denovo* (Liu et al., 2011), SGA (Simpson and Durbin, 2010) and SEECER (Le et al., 2013). While these packages have largely focused on the error correction of genomic reads (with exception to SEECER, which was designed for RNAeq reads), they may likely be used as effectively for RNAseq reads.

Recently a review (Yang et al., 2013) evaluating several of these methods in their ability to correct genomic sequence read error was published. However, the application of these techniques to RNAseq reads, as well as an understanding of how error correction influences accuracy of the *de novo* transcriptome assembly has not been evaluated. Here we aim to evaluate several of the available error correction methods. Though an understanding of the error correction process itself, specifically it's interaction with read coverage may be a useful exercise, our initial efforts described here, focus on the the effects of error correction on assembly, the resource which forms the basis of all downstream (e.g. differential expression, SNP calling) steps.

To accomplish this, we simulated 30 million paired-end Illumina reads and assembled uncorrected reads, as well as reads corrected by each of the evaluated correction methods, which were chosen to represent the breadth of computational techniques used for sequence read error correction. Though we focus on the simulated dataset, we corroborate our findings through use of an empirically derived Illumina dataset. For both datasets, we evaluate assembly content, number of errors incorporated into the assembly, and mapping efficiency in an attempt to understand the effects of error correction on assembly. Although Illumina is just one of the available high-throughput sequencing technologies currently available, we chose to limit our investigation to this single, most widely used technology, though similar investigations will become necessary as the sequencing technology evolves.

Because the *de novo* assembly is a key resource for all subsequent studies of gene expression and allelic variation, the production of an error-free reference is absolutely critical. Indeed, error in the reference itself will have potential impacts on the results of downstream analyses. These types of error may be particularly problematic in *de novo* assemblies of non-model organisms, where experimental validation of sequence accuracy may be impossible. Though methods for the correction of sequencing reads have been available for the last few years, their adoption has been limited, seemingly because a demonstration of their effects has been lacking. Here, we show that error correction has a large effect on assembly quality, and therefore argue that it should become a routine part of workflow involved in processing Illumina mRNA sequence data. Though this initial work focuses on the results of error correction; arguably the most logical candidate for study, future work will attempt to gain a deeper understanding of error in the error correction process itself.

Results

Thirty million 100nt paired-end (PE) reads were simulated using the program Flux Simulator (Griebel et al., 2012). Simulated reads were based on the coding portion of the *Mus musculus* genome and included coverage of about 60k transcripts with average depth of 70X. Thirty million reads were simulated as this corresponds to the sequencing effort suggested by (Francis et al., 2013) as an appropriate effort, balancing coverage with the accumulation of errors, particularly in non-model animal transcriptomics. These reads were qualitatively similar to several published datasets (MacManes and Lacey, 2012; Chen et al., 2011). Sequence error was simulated to follow the well-characterized Illumina error profile (Supplementary Figure 1). Similarly, patterns of gene expressions were typical of many mammalian tissues (Supplementary Figure 2), and follows a Poisson distribution with lambda=1 (Auer and Doerge, 2011; Hu et al., 2011; Jiang and Wong, 2009).

In addition to the simulated dataset, error correction was applied to an empirically derived Illumina dataset. This dataset consists of 50 million 76nt paired-end Illumina sequence reads from *Mus musculus* mRNA, and is available as part of the Trinity software package (Haas et al., 2013; Grabherr et al., 2011). Because we were interested in comparing the two datasets, we randomly selected 30 million PE reads from the total 50 million reads for analyses. The simulated read dataset is available at https://www.dropbox.com/s/mp8fu0tijox69ki/simulated.reads.tar.gz, while the empirical dataset is at https://www.dropbox.com/s/rkl0ihqom28smb2/empiric.reads.tar.gz. [Of note, these datasets are to be moved to Dryad upon acceptance for publication.]

Error correction of the simulated and empiric datasets was completed using the Seecer, AllPath-sLG, SGA, and Reptile error correction modules. Details regarding the specific numbers of nucleotide changes and the proportion of reads being affected are detailed in (Table 1). Despite the fact that each software package attempted to solve the same basic problem, runtime considerations and results were quite different. Trinity assembly using the uncorrected simulated reads produced an assembly consisting of 78.43Mb, while the assembly of empirically derived reads was 74.24Mb.

Table 1

Simulated Dataset	Total Reads	Num reads corr	Num nt corr	Runtime
Raw reads	30M PE	n/a	n/a	n/a
AllPathsLG Corr.	30M PE	?	139,592,317	∼ 8hrs
Sga Corr.	30M PE	?	19,826,919	∼ 38 minutes
Reptile Corr.	30M PE	2,047,088	7,782,594	∼ 3 hours
Seecer Corr.	30M PE	8,782,350	14,033,709	∼ 5 hours

Table 1. Number of raw sequencing reads, sequencing reads corrected, nucleotides (nt) corrected, and approximate runtime for each of the datasets. Note that neither AllPaths nor Sga provides information regarding the number of reads affected by the correction process.

Simulated Data

Analyses focused on a high-confidence subset of the data, as defined as being 99% similar to the reference over at least 90% of its length. The high-confidence subset of the simulated uncorrected read assembly (n=38459 contigs) contained approximately 54k nucleotide mismatches (Figure 1), corresponding to an mean error rate of 1.40 mismatches per contig (SD=7.38, max=178). There did not appear to be an observe an obvious relationship between gene expression and the quality of the assembled transcripts (Figure 2). While the rate of error is low, and indeed a testament to the general utility the *de Bruijn* graph approach for sequence assembly, a dramatic improvement in accuracy would be worth pursuing, if possible.

Error correction of simulated reads using Reptile was a laborious process, with multiple (>5) individual executions of the program required for parameter optimization. While each individual run was relatively quick, the total time exceed 12 hours, with manual intervention and decision making required at each execution. Error correction resulted in the correction of 7.8M nucleotides (of a total $\sim$ 5B nucleotides contained in the sequencing read dataset). The resultant assembly contains an average of 1.23 mismatches per contig (SD=6.46, max=152). The absolute number of errors decreased by $\sim$ 12% (Figure 1), which represents substantial improvement, particularly given that the high confidence subset of the Reptile-corrected assembly was the largest (n= 38670 contigs) of any of the methods (Table 2).

Figure 1

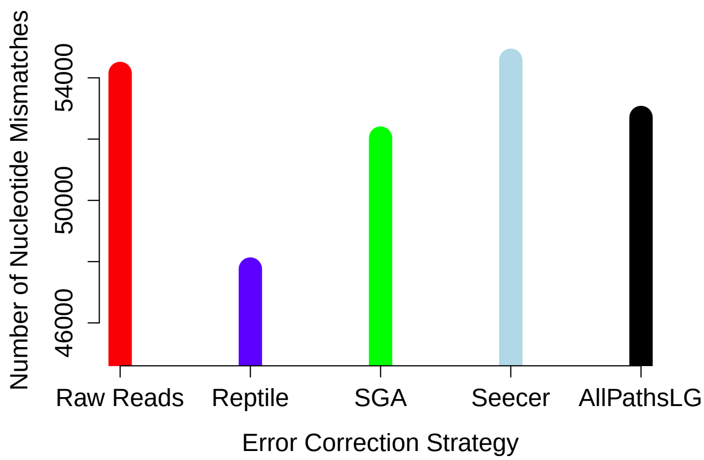

Fig. 1. The global estimate of nucleotide mismatch decreases with error correction. The assembly done with REPTILE corrected reads has approximately 10% fewer errors than does the raw read assembly.

ALLPATHSLG error correction software implemented by far the most aggressive correction, selected optimized parameters in an automated fashion, and did so within a 4 hour runtime. ALLPATHSLG corrected over 140M nucleotides (again, out of a total $\sim$ 5B nucleotides contained in the sequencing reads), which resulted in a final assembly with 52706 nucleotide errors, corresponding to a decrease in error of approximately 2.7%.

SEECER, is the only dedicated error-correction software package dedicated to RNAseq reads. Though

SEECER is expected to handle RNAseq datasets better than the other correction programs, its results were disappointing. More than 14 million nucleotides were changed, affecting approximately 8.8M sequencing reads. Upon assembly 54,574 nucleotide errors remained, which is equivalent to the number of errors contained in the assembly of uncorrected reads.

Figure 2

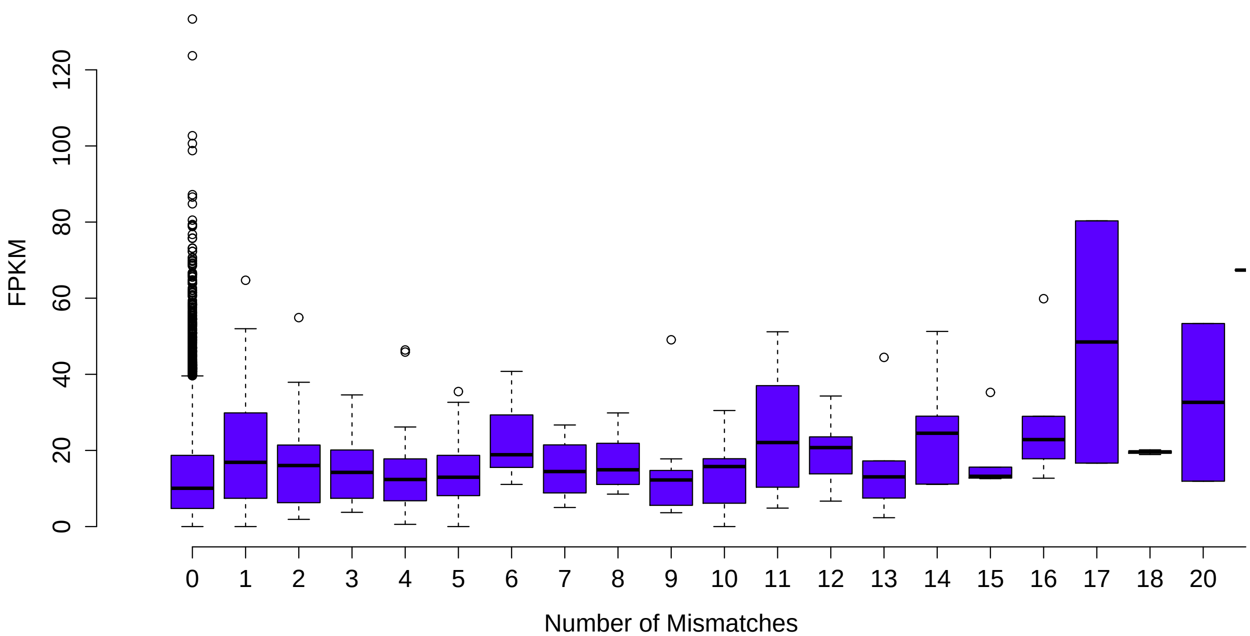

Fig. 2. The number of nucleotide mismatches in a given contig is not related to gene expression. On average, in the assembly of uncorrected simulated reads, poorly expressed transcripts are no more error prone than are highly expressed transcripts.

Lastly, SGA error correction was implemented on the simulated read dataset. SGA, is the fastest of all error correction modules, and finished correcting the simulated dataset in 38 minutes. The software applied corrections to 19.8M nucleotides. It's correction resulted in a modest improvement in error, with a reduction in error of approximately 4% over the assembly of uncorrected errors.

Assembly content, aside from fine–scaled differences at the nucleotide level, as described above, were equivalent. Assemblies consisted of between 63,099 (Reptile) – 65,468 (Seecer) putative transcripts greater than 200nt in length. N50 ranged from 2319 (Reptile) – 2403nt (Sga). The high-confidence portion of the assemblies ranged in size from 38407 contigs (Seecer assembly) to 38670 contigs in the Reptile assembly. Assemblies are detailed in Table 2, and available at http://dx.doi.org/10.6084/m9.figshare.725715.

Table 2

Dataset	Error Corr. Method	Raw Assembly Size	High Conf. Size
Simulated Reads			
	None	64491 (78Mb)	38459 (27Mb)
	AllPathsLG	64682 (78Mb)	38628 (27Mb)
	Sga	65059 (80Mb)	38619 (27Mb)
	Reptile	63099 (73Mb)	38670 (25Mb)
	Seecer	65468 (80Mb)	38407 (27Mb)
Empiric Reads			
	None	57338 (74Mb)	21406 (24Mb)
	AllPathsLG	53884 (66Mb)	21204 (23Mb)
	Sga	56707 (75Mb)	21323 (24Mb)
	Reptile	53780 (60Mb)	21850 (22Mb)
	Seecer	57311 (75Mb)	21268 (24Mb)

Table 2. Assembly details. High confidence datasets included only contigs that matched a single refernce, had sequence similarity $>$99%, and covered $\geq$ 90% of length of reference.

The proportion of reads mapping to each assembled dataset was equivalent as well, ranging from 92.44% using raw reads to 94.89% in SGA corrected reads. Assemblies did not appear to differ in general patterns of contiguity, (Figure 3), though it should be noted that the most successful error corrector, REPTILE had both the smallest assembly size *and* largest number of high confidence contigs. Taken together, these patterns suggest that error correction may have a significant effect on the structure of assembly; though its major effects are in enhancing resolution at the level of the nucleotide. Indeed, while we did not find, nor expect to find large differences in these global metrics, we do expect to see a significant effect on transcriptome based studies of marker development and population genetics, which are endeavors fundamentally linked to polymorphism, estimates of which can easily be confused by sequence error.

Empirical Data

The high-confidence subset of the uncorrected empirical read assembly (n=21406 contigs) contained approximately 14.7k nucleotide mismatches, corresponding to an mean error rate of .68 mismatches per contig (SD=3.60 max=197). Error correction procedures were implemented as described above. Indeed, the resultant pattens of correction were recapitulated. Error correction using REPTILE were most favorable, and resulted in a reduction in the number of nucleotide errors by more than 10%, to approximately 13k. As above, the high-confidence portion of the REPTILE-corrected dataset was the largest, with 21580 contigs, which is slightly larger that the assembly of uncorrected reads. Similar to what was observed in the simulated dataset, the high-confidence portion of the ALLPATHS corrected assembly was the smallest of any of the datasets, and contained the most error. Of interest, the SGA correction performed well, similar to as in simulated reads, decreasing error by more than 9%.

Empirical assemblies contained between 53780 (REPTILE) and 57338 (uncorrected assembly) contigs greater than 200nt in length. N50 ranged from between 2412 (REPTILE) and 2666nt (SEECER) in length. As above, assemblies did not differ widely in their general content or structure; instead effects were limited to differences at nucleotide level. Assemblies are available at http://dx.doi.org/10.

6084/m9.figshare.725715.

Figure 3

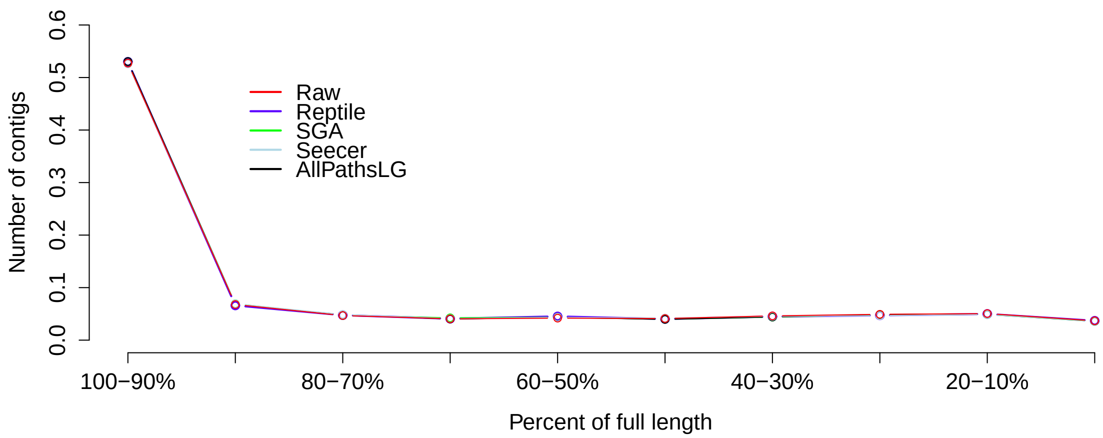

Fig. 3. Assembly contiguity did not vary significantly between assemblies of reads using the different error correction methods. Each error correction methods, as well as assembly of raw reads, produced an assembly that is dominated by full length (both start and stop codon present) or nearly full length assembled transcripts.

Discussion

Though the methods for error correction have become increasingly popular within the last few years, their adoption in general genome or transcriptome assembly pipelines has lagged. One potential reason for this lag has been that their effects on assembly, particularly in RNAseq, has not been demonstrated. Here, we attempt to evaluate the effects of four different error correction algorithms on assembly- arguably the step upon which all downstream steps (e.g. differential expression, functional genomics, SNP discovery, etc.) is based. We use both simulated and empirically derived data to show a significant effect of correction on assembly– especially when using the error corrector Reptile. This particular method, while relatively labor intensive to implement, reduces error by more than 10%, and results in a larger high-confidence subset relative to other methods.

Interesting, SEECER, the only error correction method designed for RNAseq reads, performed relatively poorly. In simulated reads, SEECER slightly increased the number of errors in the assembly, though with applied to empirically derived reads, results were more favorable, decreasing error by $\sim$ 3%. Though the effects of coverage on correction efficiency were not explored in the manuscript describing SEECER (Le et al., 2013), their empirical dataset contained nearly 90 million sequencing reads, a size 3X larger than the dataset we analyze here. Future work investigating the effects of coverage on error correction is necessary.

In addition to this, how error correction interacts with the more complicated reconstructions, splice variants for instance, is an outstanding question. Indeed, reads traversing a splicing junction may be particularly problematic for error correctors, as coverage on opposite sides of the junction may be different owing to differences in isoform expression, which could masquerade as error. Alternative splicing is known to negatively affect both assembly and mapping (Vijay et al., 2013; Sammeth, 2009; Pyrkosz et al., 2013), and given that many computational strategies are shared between these techniques and error correction suggests that similarly, error correction should be affected by splicing. As such, considering this potential source of error in error correction should be considered during error correction. Computational strategies that distinguish these alternative splicing events from real error are currently being developed.

The effects of read coverage on the efficiency of error correction are likely strong. Aside from the suggestion that SEECER'S relatively poor performance owed to low coverage data relative to the dataset tested during the development of that software (Le et al., 2013), other supporting evidence exists. Approximately 5% or reads are miscorrected. When looking at a sample (n=50000) of these reads, the contig to which that read maps is on average more lowly expressed than appropriately corrected reads (Figure 4), which suggests that low coverage may reduce the efficiency of error correction. In addition, miscorrected reads, whose average expression is lower, tends to have more corrections than to the appropriately corrected reads (Figure 5).

Figure 5

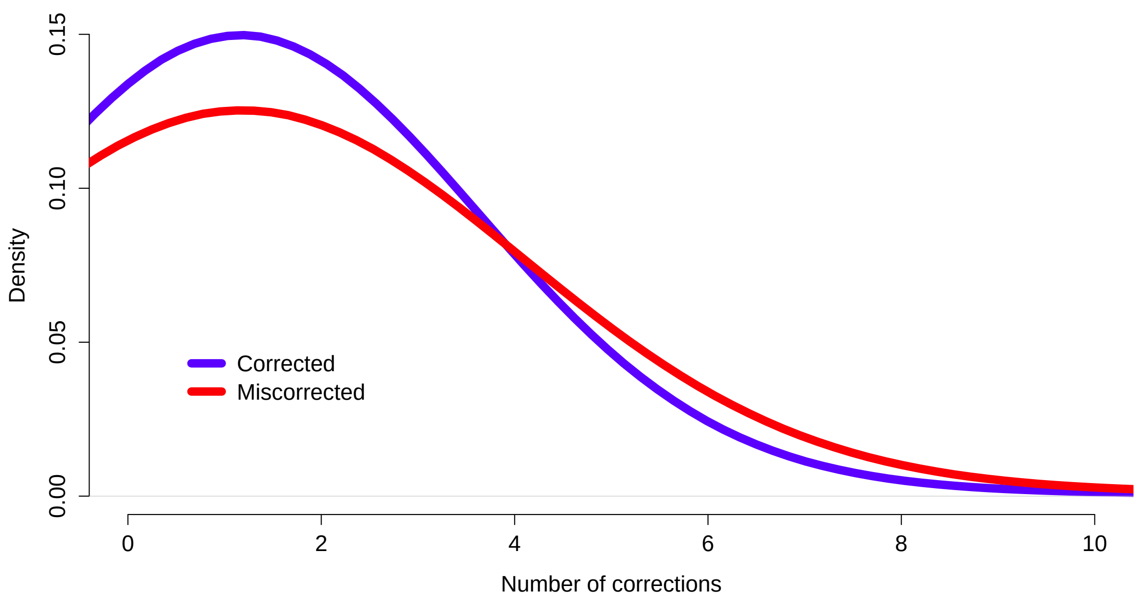

Fig. 5. Reads miscorrected by REPTILE

Methods

Because we were interested in understanding the effects of error correction on the assembly of vertebrate transcriptome assembly, we elected to use coding sequences greater than 200nt in length from the *Mus musculus* reference genome (GRCm37.71), available at `http://uswest.ensembl.org/Mus_musculus/Info/Index`. Thirty million 100nt paired-end Illumina reads were simulated with the program FLUX SIMULATOR (Griebel et al., 2012) which attempts to simulate a realistic Illumina RNAseq dataset, incorporating biases related to library construction and sequencing. Thirty million PE reads were simulated as this sequencing effort was suggested to be optimal for studies of whole-animal non-model transcriptomes (Francis et al., 2013). Sequencing error increased along the length of the read, as per program default. Patterns of gene expression were modeled to follow patterns typically seen in studies of Eukaryotic gene expression. The FLUX SIMULATOR requires the use of a parameter file, which is available at https://gist.github.com/macmanes/5859902.

In addition to analyses conducted on a simulated dataset, we used the well-characterized mouse dataset included with the Trinity software package (http://sourceforge.net/projects/trinityrnaseq/files/misc/MouseRNASEQ/mouse_SS_rnaseq.50M.fastqs.tgz/download) to validate the observed patterns using an empirically derived dataset. To enable comparison between the simulated and empiric dataset, we randomly selected a subset of this dataset consisting of 30 million PE reads.

Quality metrics for simulated and experimental raw reads were generated using the program SOLEXAQA (Cox et al., 2010), and visualized using R (R Core Development Team, 2011). Patterns of gene expression were validated using the software packages BOWTIE2 (Trapnell et al., 2010) and EXPRESS (Roberts and Pachter, 2012). All computational work was performed on a 16-core 36GB RAM Linux Ubuntu workstation.

Error correction was performed on both simulated and empirical datasets using four different error correction software packages. These included SEECER, ALLPATHSLG error correction, REPTILE, and SGA. These specific methods were chosen in an attempt to cover the breadth of analytical methods currently used for error correction. Indeed, each of these programs implements a different computational strategy for error correction, and therefore their success, and ultimate effects on assembly accuracy are expected to vary. In addition, several of these packages have been included in a recent review of error correction methods, with one of these (REPTILE) having been shown to be amongst the most accurate (Yang et al., 2013).

Though error correction has been a part of the ALLPATHSLG genome assembler for the past several versions, only recently has a stand-along version of their python-based error correction module (http://www.broadinstitute.org/software/allpaths-lg/blog/?p=577), which leverages several of the AllPaths subroutines, become available. With exception to the minimum kmer frequency, which was set to 0 (unique kmers retained in the final corrected dataset), the ALLPATHSLG error correction software was run using default settings for correcting errors contained within the raw sequencing reads. Code for running the program is available at https://gist.github.com/macmanes/5859931.

Error correction using the software package REPTILE requires the optimization of several parameters via an included set of scripts, and therefore several runs of the program. To correct errors contained within the raw dataset, we set kmer size to 25 (*KmerLen=25*), and the maximum error rate to 2% (*MaxErrRare=0.02*). Kmer=25 was selected to most closely match the kmer size used by the assembler TRINITY. We empirically determined optimal values for *T_expGoodCnt* and *T_card* using multiple independent program executions. REPTILE requires the use of a parameter file, which is available at https://gist.github.com/macmanes/5859947.

The software package SGA was also used to correct simulated and empiric Illumina reads. This program, like ALLPATHS-LG, allows its error correction module to be applied independent of the rest of the pipeline. These preliminary steps, preprocessing, indexing, and error correction were run with default settings, with exception to the kmer size, which was set to 25.

Lastly, the software package SEECER was used to error correct the raw read dataset. The software package is fundamentally different than the other packages, in that it was designed for with RNAseq reads in mind. We ran SEECER using default settings.

Transcriptome assemblies were generated using the default settings of the program TRINITY (Grabherr et al., 2011). Code for running TRINITY is available at https://gist.github.com/macmanes/5859956. Assemblies were evaluated using a variety of different metrics. First, BLAST+ (Camacho et al., 2009) was used to match assembled transcripts to their reference. TRANSDECODER (http://transdecoder.sourceforge.net/) was used to identify full-length transcripts. For analysis of nucleotide mismatch, we elected to analyze a 'high-confidence' portion of out dataset as multiple hits and low quality BLAT matches could significantly bias results. To subset the data, we chose to include only contigs whose identity was $\geq$ 99% similar to, and covering at least 90% of the reference sequence. The program BLAT (Kent, 2002) was used to identify and count nucleotide mismatches between reconstructed transcripts in the high-confidence datasets and their corresponding reference. Differences were visualized using the program R.

Conclusions

To evaluate the effects of correction of sequencing error on assembly accuracy, we generated a simulated Illumina dataset, which consisted of 30M paired-end reads. In addition, we applied the selected error correction strategy to an empirically derived *Mus musculus* dataset. We attempted error correction using four popular error correction software packages, and evaluated their effect on assembly. Though originally developed with genome sequencing in mind, we found that all tested methods do correct mRNAseq reads, and increase assembly accuracy, though REPTILE appeared to have the most favorable effect. This study demonstrates the utility of error correction, and proposes that it become a routine step in the processing of Illumina sequence data.

Acknowledgments

This paper was greatly improved by suggestions from members of the Eisen Lab, and from two named reviewers, C. Titus Brown and Mick Watson.

References

Auer, P.L., Doerge, R.W., 2011. A Two-Stage Poisson Model for Testing RNA-Seq Data. Statistical Applications in Genetics and Molecular Biology 10, 1–26.

Bradnam, K.R., Fass, J.N., Alexandrov, A., Baranay, P., Bechner, M., Boisvert10, S., Chapman, J.A., Chapuis, G., Chikhi, R., Chitsaz, H., Chou, W.C., Corbeil, J., Del Fabbro, C., Docking, T.R., Durbin, R., Earl, D., Emrich, S., Fedotov, P., Fonseca, N.A., Ganapathy, G., Gibbs, R.A., Gnerre, S., Godzaridis, É., Goldstein, S., Haimel, M., Hall, G., Haussler, D., Hiatt, J.B., Ho, I.Y., Howard, J., Hunt, M., Jackman, S.D., Jaffe, D.B., Jarvis, E., Jiang, H., Kazakov, S., Kersey, P.J., Kitzman, J.O., Knight, J.R., Koren, S., Lam, T.W., Lavenier, D., Laviolette, F., Li, Y., Li, Z., Liu, B., Liu, Y., Luo, R., MacCallum, I., MacManes, M.D., Maillet, N., Melnikov, S., Vieira, B.M., Naquin, D., Ning, Z., Otto, T.D., Paten, B., Paulo, O.S., Phillippy, A.M., Pina-Martins, F., Place, M., Przybylski, D., Qin, X., Qu, C., Ribeiro, F.J., Richards, S., Rokhsar, D.S., Ruby, J.G., Scalabrin, S., Schatz, M.C., Schwartz, D.C., Sergushichev, A., Sharpe, T., Shaw, T.I., Shendure, J., Shi, Y., Simpson, J.T., Song, H., Tsarev, F., Vezzi, F., Vicedomini, R., Wang, J., Worley, K.C., Yin, S., Yiu, S.M., Yuan, J., Zhang, G., Zhang, H., Zhou, S., Korf, I.F., 2013. Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species. arXiv.org arXiv:1301.5406v1.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., Madden, T.L., 2009. BLAST+: architecture and applications. BMC Bioinformatics 10, 421.

Chaisson, M., Pevzner, P., Tang, H.X., 2004. Fragment assembly with short reads. Bioinformatics (Oxford, England) 20, 2067–2074.

Chen, Z., Liu, J., Ng, H.K.T., Nadarajah, S., Kaufman, H.L., Yang, J.Y., Deng, Y., 2011. Statistical methods on detecting differentially expressed genes for RNA-seq data. BMC Systems Biology 5, S1.

Conway, T.C., Bromage, a.J., 2011. Succinct data structures for assembling large genomes. Bioinformatics (Oxford, England) 27, 479–486.

Cox, M.P., Peterson, D.A., Biggs, P.J., 2010. SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. BMC Bioinformatics 11, 485.

Earl, D., Bradnam, K., St John, J., Darling, A., Lin, D., Fass, J., Yu, H.O.K., Buffalo, V., Zerbino, D.R., Diekhans, M., Nguyen, N., Ariyaratne, P.N., Sung, W.K., Ning, Z., Haimel, M., Simpson, J.T., Fonseca, N.A., Birol, I., Docking, T.R., Ho, I.Y., Rokhsar, D.S., Chikhi, R., Lavenier, D., Chapuis, G., Naquin, D., Maillet, N., Schatz, M.C., Kelley, D.R., Phillippy, A.M., Koren, S., Yang, S.P., Wu, W., Chou, W.C., Srivastava, A., Shaw, T.I., Ruby, J.G., Skewes-Cox, P., Betegon, M., Dimon, M.T., Solovyev, V., Seledtsov, I., Kosarev, P., Vorobyev, D., Ramirez-Gonzalez, R., Leggett, R., MacLean, D., Xia, F., Luo, R., Li, Z., Xie, Y., Liu, B., Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F.J., Yin, S., Sharpe, T., Hall, G., Kersey, P.J., Durbin, R., Jackman, S.D., Chapman, J.A., Huang, X., DeRisi, J.L., Caccamo, M., Li, Y., Jaffe, D.B., Green, R.E., Haussler, D., Korf, I., Paten, B., 2011. Assemblathon 1: A competitive assessment of *de novo* short read assembly methods. Genome Research 21, 2224–2241.

Francis, W.R., Christianson, L.M., Kiko, R., Powers, M.L., Shaner, N.C., Haddock, S.H.D., 2013. A comparison across non-model animals suggests an optimal sequencing depth for *de novo* transcriptome assembly. BMC Genomics 14, 167.

Gnerre, S., MacCallum, I., Przybylski, D., Ribeiro, F.J., Burton, J.N., Walker, B.J., Sharpe, T., Hall, G., Shea, T.P., Sykes, S., Berlin, A.M., Aird, D., Costello, M., Daza, R., Williams, L., Nicol, R., Gnirke, a., Nusbaum, C., Lander, E.S., Jaffe, D.B., 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proceedings of the National Academy of Sciences 108, 1513–1518.

Grabherr, M.G., Haas, B.J., Yassour, M., Levin, J.Z., Thompson, D.A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., Chen, Z., Mauceli, E., Hacohen, N., Gnirke, a., Rhind, N., di Palma, F., Birren, B.W., Nusbaum, C., Lindblad-Toh, K., Friedman, N., Regev, A., 2011. Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nature Biotechnology 29, 644–652.

Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigó, R., Sammeth, M., 2012. Modelling and simulating generic RNA-Seq experiments with the flux simulator. Nucleic Acids Research 40, 10073–10083.

Haas, B.J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P., Bowden, J., Couger, M., Eccles, D., Li, B., Lieber, M., MacManes, M.D., Ott, M., Orvis, J., Pochet, N., Strozzi, F., Weeks, N., Westerman, R., William, T., Dewey, C.N., Henschel, R., LeDuc, R.G., Friedman, N., Regev, A., 2013. *De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nature protocols , 1–21.

Hsu, J.C., Chien, T.Y., Hu, C.C., Chen, M.J.M., Wu, W.J., Feng, H.T., Haymer, D.S., Chen, C.Y., 2012. Discovery of Genes Related to Insecticide Resistance in *Bactrocera dorsalis* by Functional Genomic Analysis of a *De Novo* Assembled Transcriptome. PLOS ONE 7, e40950.

Hu, M., Zhu, Y., Taylor, J.M.G., Liu, J.S., Qin, Z.S., 2011. Using Poisson mixed-effects model to quantify transcript-level gene expression in RNA-Seq. Bioinformatics (Oxford, England) 28, 63–68.

Jiang, H., Wong, W.H., 2009. Statistical inferences for isoform expression in RNA-Seq. Bioinformatics (Oxford, England) 25, 1026–1032.

Kao, W.C., Chan, a.H., Song, Y.S., 2011. ECHO: a reference-free short-read error correction algorithm. Genome Research 21, 1181–1192.

Kelley, D.R., Schatz, M.C., Salzberg, S.L., 2010. Quake: quality-aware detection and correction of sequencing errors. Genome Biology 11, R116.

Kent, W.J., 2002. BLAT—The BLAST-Like Alignment Tool. Genome Research 12, 656–664.

Le, H.S., Schulz, M.H., McCauley, B.M., Hinman, V.F., Bar-Joseph, Z., 2013. Probabilistic error correction for RNA sequencing. Nucleic Acids Research .

Linnen, C.R., Poh, Y.P., Peterson, B.K., Barrett, R.D.H., Larson, J.G., Jensen, J.D., Hoekstra, H.E., 2013. Adaptive Evolution of Multiple Traits Through Multiple Mutations at a Single Gene. Science (New York, NY) 339, 1312–1316.

Liu, B., Yuan, J., Yiu, S.M., Li, Z., Xie, Y., Chen, Y., Shi, Y., Zhang, H., Li, Y., Lam, T.W., Luo, R., 2012. COPE: an accurate k-mer-based pair-end reads connection tool to facilitate genome assembly. Bioinformatics (Oxford, England) 28, 2870–2874.

Liu, Y., Schmidt, B., Maskell, D.L., 2011. Parallelized short read assembly of large genomes using *de Bruijn* graphs. BMC Bioinformatics 12, 354.

MacManes, M.D., Lacey, E.A., 2012. The Social Brain: Transcriptome Assembly and Characterization of the Hippocampus from a Social Subterranean Rodent, the Colonial Tuco-Tuco (*Ctenomys sociabilis*). PLOS ONE 7, e45524.

Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M., Gilad, Y., 2008. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. Genome Research 18, 1509–1517.

Muñoz Merida, A., Gonzalez-Plaza, J.J., Canada, A., Blanco, A.M., Garcia-Lopez, M.d.C., Rodriguez, J.M., Pedrola, L., Sicardo, M.D., Hernandez, M.L., De la Rosa, R., Belaj, A., Gil-Borja, M., Luque, F., Martinez-Rivas, J.M., Pisano, D.G., Trelles, O., Valpuesta, V., Beuzon, C.R., 2013. De Novo Assembly and Functional Annotation of the Olive (*Olea europaea*) Transcriptome. DNA Research 20, 93–108.

Miller, J.R., Koren, S., Sutton, G., 2010. Assembly algorithms for next-generation sequencing data. Genomics 95, 315–327.

Mortazavi, A., Williams, B.A., Mccue, K., Schaeffer, L., Wold, B., 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nature Methods 5, 621–628.

Narum, S.R., Campbell, N.R., Meyer, K.A., Miller, M.R., Hardy, R.W., 2013. Thermal adaptation and acclimation of ectotherms from differing aquatic climates. Molecular Ecology , 1–8.

Pell, J., Hintze, A., Canino-Koning, R., Howe, A., Tiedje, J.M., Brown, C.T., 2012. Scaling metagenome sequence assembly with probabilistic *de Bruijn* graphs. Proceedings of the National Academy of Sciences 109, 13272–13277.

Pyrkosz, A.B., Cheng, H., Brown, C.T., 2013. RNA-Seq Mapping Errors When Using Incomplete Reference Transcriptomes of Vertebrates. http://arxiv.org/abs/1303.2411v1 arXiv:1303.2411v1.

R Core Development Team, F., 2011. R: A Language and Environment for Statistical Computing .

Roberts, A., Pachter, L., 2012. Streaming fragment assignment for real-time analysis of sequencing experiments. Nature Methods , 1–7.

Sammeth, M., 2009. Complete alternative splicing events are bubbles in splicing graphs. Journal of computational biology : a journal of computational molecular cell biology 16, 1117–1140.

Simpson, J.T., Durbin, R., 2010. Efficient construction of an assembly string graph using the FM-index. Bioinformatics (Oxford, England) 26, i367–i373.

Soneson, C., Delorenzi, M., 2013. A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics 14, 91.

Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J., Pachter, L., 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nature Biotechnology 28, 511–U174.

Vijay, N., Poelstra, J.W., Künstner, A., Wolf, J.B.W., 2013. Challenges and strategies in transcriptome assembly and differential gene expression quantification. A comprehensive *in silico* assessment of RNA-seq experiments. Molecular Ecology 22, 620–634.

Yang, X., Chockalingam, S.P., Aluru, S., 2013. A survey of error-correction methods for next-generation sequencing. Briefings In Bioinformatics 14, 56–66.

Yang, X., Dorman, K.S., Aluru, S., 2010. Reptile: representative tiling for short read error correction. Bioinformatics (Oxford, England) 26, 2526–2533.