Tailor: a computational framework for detecting non-templated tailing of small silencing RNAs

Min-Te Chou
National Chiao Tung University

Let us know how access to this document benefits you.
Follow this and additional works at: https://escholarship.umassmed.edu/gsbs_sp

Part of the Biochemistry Commons, Bioinformatics Commons, and the Computational Biology Commons

Repository Citation

Creative Commons License
This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 License
This material is brought to you by eScholarship@UMassChan. It has been accepted for inclusion in Morningside Graduate School of Biomedical Sciences Student Publications by an authorized administrator of eScholarship@UMassChan. For more information, please contact Lisa.Palmer@umassmed.edu.
Tailor: a computational framework for detecting non-templated tailing of small silencing RNAs

Min-Te Chou1,†, Bo W. Han2,†, Cheing-Po Hsiao1, Phillip D. Zamore2, Zhiping Weng3 and Jui-Hung Hung1,*

1 Institute of Bioinformatics and Systems Biology and Department of Biological Science and Technology, National Chiao Tung University, Hsinchu, Taiwan, 300, Republic of China, 2 RNA Therapeutics Institute, Howard Hughes Medical Institute, and Department of Biochemistry and Molecular Pharmacology, University of Massachusetts Medical School, Worcester, MA 01605, USA and 3 Program in Bioinformatics and Integrative Biology, University of Massachusetts Medical School, Worcester, MA 01605, USA

Received March 31, 2015; Revised April 24, 2015; Accepted May 10, 2015

ABSTRACT

Small silencing RNAs, including microRNAs, endogenous small interfering RNAs (endo-siRNAs) and Piwi-interacting RNAs (piRNAs), have been shown to play important roles in fine-tuning gene expression, defending virus and controlling transposons. Loss of small silencing RNAs or components in their pathways often leads to severe developmental defects, including lethality and sterility. Recently, non-templated addition of nucleotides to the 3′ end, namely tailing, was found to associate with the processing and stability of small silencing RNAs. Next Generation Sequencing has made possible to detect such modifications at nucleotide resolution in an unprecedented throughput. Unfortunately, detecting such events from millions of short reads confounded by sequencing errors and RNA editing is still a tricky problem. Here, we developed a computational framework, Tailor, driven by an efficient and accurate aligner specifically designed for capturing the tailing events directly from the alignments without extensive post-processing. The performance of Tailor was fully tested and compared favorably with other general-purpose aligners using both simulated and real datasets for tailing analysis. Moreover, to show the broad utility of Tailor, we used Tailor to reanalyze published datasets and revealed novel findings worth further experimental validation. The source code and the executable binaries are freely available at https://github.com/jhhung/Tailor.

INTRODUCTION

Over the past decade, small silencing RNAs, including microRNAs (miRNAs), endogenous small silencing RNAs (endo-siRNAs) and Piwi-interacting RNAs (piRNAs) have been shown to play indispensable roles in regulating gene expression, protecting against viral infection and preventing mobilization of transposable elements (1–4). Small silencing RNAs exert their silencing function by associating with Argonaute proteins to form RNA-induced silencing complex (RISC), which uses the small RNA guide to nd its regulatory targets and reduce gene expression. Although the studies on the biogenesis of small silencing RNAs have made enormous progress in the past decade, the factors controlling their stability and degradation remain elusive.

Recent studies have suggested that non-templated addition to the 3′ end of small silencing RNAs, namely tailing, could play essential roles in this regard. Non-templated 3′ mono- and oligo-uridylation of the pre-microRNA (pre-miRNA) regulates miRNA processing by either preventing or promoting Dicer cleavage in Drosophila (5–7). The 3′ mono-uridylation on small interfering RNAs in Caenorhabditis elegans is associated with negative regulation (8). Am et al. have demonstrated that highly conserved 3′ terminal targets trigger the tailing of miRNAs and eventually lead to their degradation in mice and mammals (9,10); a similar mechanism has been found on some endo-siRNAs as well (11). Identification of tailing events not only suggests the co-evolution of small silencing RNAs and their targets, but also sheds light on the mechanism of their maturation and degradation.

Despite the fact that Next Generation Sequencing (NGS) has greatly facilitated the understanding of RNA tailing, computational detection of non-templated nucleotides from millions of sequencing reads is challenging. The Ketting group used MegaBLAST to align piRNA sequences
to the genome and resulted in post-processing the reported m i m atches gain insights into tailing (8). However, as a heuristic algorithm, BLAST is not guaranteed to nd all the tailing events in the reference and is signi cantly slower than the NG S aligner like MQ (14), BWA (15), Bowtie (16) and SOAP (17). The Chen group used an accurate method that iterates between Bowtie alignment and clipping of unm atched reads (19) to nd all the perfect alignments of trim m ed reads. A similar approach has been used for re m oving erroneous bases at 3 end to increase the sensitivity of detecting mi RNAs (19). Let alone that this method inevitably multiply the running time by an m atching length of tails, erroneous computational works are still needed to retrieve the identity of each trim m ed tail. The study by Am eres et al. used a specialized a tch x tree data structure to ef ciently nd all the tails without sacri cing the accuracy. However, due to the high m em ory footprint of the a tch x tree data structure, which is about 16 to 20 times of the genome size, the read m apping has to be performed for each chrom osome separately (9,20). Extra processing is still required to m atch all the alignments from all chrom osomes.

Moreover, the task becomes even trickier when technical and biological confounding factors are taken into account for better capturing the true tailing events. For example, it is known that reads from small RNAs in H. sapiens and G. eocome analyzer platform have a preference toward C conversions (21,22) and a high error rate at the 3 end of reads, which frequently leads to uncalled bases, i.e. B-tails (23,24). In addition to these technical artifacts enueyed by the sequencers, RNA editing is another common on post-transcriptional m odif cation in an silencing RNA biology that could perplex the tools with erroneous alignments. There are two m ajor types of RNA editing in mammals, adenosine to inosine (A-to-I) and cytidine to uridine (C-to-U) editing. The major enzyme that catalyzes adenosine to inosine are the adenosine deaminases acting on RNA (ADARs), whose main substrates are RNAs with double-stranded structures (25–27). Since any small RNAs are originated from structural RNAs, they are all likely targets of A-to-I editing (28–30). Recent studies have shown that A-to-I editing can occur on the seed region of the mi RNAs with a high occurrence rate (up to 80% in some cases) and have a direct m pact on the selection of their regulatory targets (31,32). Those unm atched bases degenerate the sensitivity and accuracy of short read alignment and have a negative eect on the detection of tailing.

Most of the current methods simply ignore those confounding factors and rely on adapting existing, less specialized tools with extensive post-processing and as a consequence the perform ance, usefulness and application of tailing analysis is seriously com prom ised. A fast, accurate and straightforward approach to study tailing is still in need. To ease the cost of performing tailing analysis with dram atically increasing sequencing throughput, we here introduce Tailor—a fast m atwork that preprocesses and maps sequences to a reference, distinguishes tails from mis atches or bad alignments with a novel algorithm and reports both perfect and tailied alignments simultaneously without loss of information. Tailor is capable of analyzing the non-tailed alignment form RNA and other types of small RNAs and produce publication-quality summaries. In addition, to better demonstrate the utility of Tailor, we reanalyzed published datasets with Tailor and unearthed several interesting observations (see Applications—case studies in Results). Although the ndings still require thorough experimental validation, it is clear that Tailor would help expand the scope of the study of an silencing RNAs.

MATERIALS AND METHODS

Datasets

Illumina sequencing data of an all RNAs from D. melanogaster hen(1 SRR 029608, SRR 029633), D. anb ser (SRR 363984–5), A. thaliana hen (SRR P010683) and A. gongv associated small RNAs in cytoplasm (SRR 529079) and nuclear fraction (SRR 529100) of H. E. A were obtained from NCBI Sequence Read Archive. The length distribution of the simulated confounded reads was from the D. melanogaster A g03 associated small RNAs extracted from ovaries (SRR 916073). In-house program was used to trim the 3 adaptors and m er the reads with low quality. Randomly distributed reads from fruit y genome was generated by Articulate (33).

Ten million reads were randomly chosen using seqtk (github.com/lh3/seqtk.git) with options 'sampl e -p $00 -l000000'. To remove multiple mapping reads in some simulation datasets, we used Bowtie iteratively before and after the tail appending and seeding mutation to assure each read has only one occurrence in the reference.

Rationale

The principle of detecting non-tailed plated bases at the 3 end of reads is basically to nd the longest common subsequence of G and R between the read and each of the suf xes of the reference and then report the m ender on the read as a tail. G is a read R (M base pairs long) and all the suf xes (S) of a reference sequence G (N bp long), one can nd the LCP between R and S by nding the longest consecutive m atches from the 1st base to the last. Since there are totally N suf xes of G , a trivial solution needs atomega N *M m times of computation to nd the LCP of R and G; however the perf ormance is unacceptable slow when G is as large as a human genome. Using index structures, such as the su x tree or suf x array, nding LCPs between the NGS reads and the reference can be solved much more e ciently (34).

Recently, the Full-text Index in M inute space (FM-index) derived from the Burrows-Wheeler transform (BW T) (35–37) is widely used in many NGS applications (15–17). The FM-index is both time and space ef cient and can be built from a suf x array and requires only 3 to 4 bits per base to store the index. A more detailed introduction of building the FM-index of long biological sequences is given in the Supplementary Materials. However, since the FM-index is originally designed for matching all bases of a read to a substring of the reference, it cannot be used directly for nding tails. One straightforward solution is to align reads without those non-tailed plated bases by repeatedly rem oving one last base in each round of the algorithm until at least one perfect hit is found (18), but the approach scar es the speed greatly and requires extensive post-processing. To bene t from the space and m eiciency of the FM-index,
we further modified its matching procedure and adapted the error-tolerant strategy proposed by Langmead et al. (16) to devise an FM-index-based tail detection algorithm, Tailor, which is specialized in capturing the non-templated bases at the 3' end of reads with confounding factors, such as sequencing errors and RNA editing.

Read mapping algorithm of Tailor

The system flow of the Tailor algorithm is outlined in Figure 1. Since searching within the FM-index initiates from the 3' end of the query string (i.e. the read) (36), where the non-templated nucleotides append, Tailor first makes the reverse-complement ent of the query sequence so that searching starts from the original 5' end to avoid excessive exhaustive search at the early stage. To do so, the reference should be reversed complemented as well, and the coordinates of each alignment should be calculated accordingly. To allow searching against both strands simultaneously and in proves the speed, Tailor concatenates the plus and minus strands of the reference and constructs one index instead of two (Figure 1A and Supplementary Materials). Tailor also stores a part of the suffix array similar to other FM-index based aligners (16,38–40) to achieve fast calculation of the text shift for getting the coordinate of each occurrence. Any alignent whose pre x matching portion exceeds the boundary of the m apped chromosome is ignored. The searching continues until either mismatches all the characters of the query to the reference (i.e., the perfect match) or no more bases can be matched (i.e., the plus x matching). In the latter case, Tailor backtracks to the previous matched position and exhaustively enumerates all the possible pre x matches. The unm matched part remain in the query is reported as a tail (Figure 1B).

Clearly, this strategy is vulnerable to confounding factors, since the first mismatch encountered directly determines the seedinder as the tail, which can be very misleading. To accommodate possible sequencing errors or RNA editing events in a read, we devised specialized selection rules as depicted in Figure 2. For each read, the first S (S = 18 by default) bases at its 5' portion is dened as the seed (Figure 2A). Given the fact that sequencing errors tend to occur at the 3' end (23,24) and RNA editing events in mRNAs are enriched at the other end (i.e., the seed region) (30–32), the selection rules behave according to whether or not the first mismatch appears in the seed (Figure 2B).

If the first mismatch is not in the seed region, it is regarded as either the first base of the tail or a sequencing error. In the case that the first mismatch is at the last base, it is directly deemed as a valid tail (Case 2 in Figure 2B). If the tail is longer than 1 nucleotide (nt), it will be further scanned to make sure that the sequence of the tail consists of multiple non-templated nucleotides (Case 3). If the tail is only one nucleotide different from the reference, no tail but a mismatch will be reported (Case 4). Note that in order to differentiate tails from sequencing errors, a trimming step based on the quality is necessary to avoid type I error and has been included in Tailor's pipeline (see below; Analysis pipeline). Our current algorithm cannot differentiate the circumstances that the tail had sequence is identical to the genome sequence. This problem is unlikely to be solved computationally and experimental solutions are expected to be more effective (e.g., using mutants with a defective tailing pathway).

On the other hand, if the first mismatch is in the seed, where RNA editing events occur frequently, the backtracking search will be eliminated and looks for an LCP started from the succeeding base after the mismatch. If no mismatch is found in the initiated search, no tail but a mismatch is reported (Case 5). If a mismatch is found outside the seed, the remainder is reported as a tail (Cases 6 and 7); otherwise, the read is dropped (Case 8). Note that the scenario that Case 4 with another m ismatch in the seed is not allowed (i.e. two mismatches as in Case 8), since in principle we want to avoid Tailor an error tolerance strategy consistent to that of conventional approaches under the one mismatch setting (e.g. -v 1 in Bowtie).

In pipeline

We implemented the core of the Tailor aligner using C++ with built-in support for multithreading. Since Tailor concatenates both strands of the chromosome into a single reference, whose length could exceed the match number represented by 32 bits, we have to use 64 bits to store the indexes in all the relevant data structures, which require about 2X memory footprint than that of other FM-index based aligners. To backward compatibility with the algorithm introduced in Aman et al. (9), which allow only case 1, 2 and 3 in Figure 2, an option (−v) is needed to turn on the detection of other cases. Tailor has a similar command line interface like other NG S aligners and reports alignment in the SAM (41) format. A tail is described as ‘soft-clipping’ in CIGAR and the sequences are reported under ‘TLZ:’ in the optional fields. A mismatch, if allowed (−v), will be reported in the MD tag (see Supplementary Materials). Tailor is freely available on GitHub (http://jhhung.github.io/Tailor/) under GNU General Public License 2. All the scripts used in preparing this manuscript have also been included in the same GitHub repository. The tailing pipelines were implemented in shell scripting language and R.

Test environment and software

All software tests were performed in the x86_64 Centos environment with 24 cores and 48G of memory. The Bowtie software used in this study is version 1.0.0, 64-bit. The version of BWA used is 0.7.5a-r505. The version of Tailor used is 1.0.0. All comma nds for all the tests are listed in the Supplementary Materials.

RESULTS

Performace without confounding factors

To begin with, we ignored confounding factors in the following tests to compare with conventional approaches. To assess the aligning speed directly, we indiscriminately generated 10 million perfectly genome-matched reads from the D. melanogaster genome (simulated tail-free dataset) (33) and randomly appended 1–4 genome-unique matched nucleotides to the 3' ends (simulated tailled
Figure 1. BW T-based tailing detection algorithm. A) Procedure of constructing the FM-index from a reference sequence. B) Procedure of query searching using the FM-index. Searching starts from the 3' end of a reverse-complemented query. Green letters indicate the non-templated tail. Red letters indicate the positions being matched against the index. When a non-templated letter is spotted as in step 4, the algorithm backtracks to previous steps and reports all the hits and marks the unmatched string as 'tail'.

We compared Tailor with two most popular BW T aligners Bowtie and BWA by applying them on simulated small RNA datasets (Figure 3A). For the simulated tail-free dataset, Tailor outperformed Bowtie and BWA in various thread settings (using 2, 4, 8, 12 and 24 threads; Figure 3A, top). All the running time plotted was the average of the actual running time of repeated experiments. But for the simulated tailed dataset, Bowtie ran slightly faster than Tailor possibly due to the fact that the reported no alignment and did not perform any disk writing (Figure 3A, bottom).

We also performed the speed test with real small RNA sequencing data from hen1+/− and hen1−/− datasets (Figure 3B). hen1 encodes for a methyl-transferase that adds a methyl group to the 3' end of siRNA and piRNA at the 2'-O position and prevents tailing (9,42). For both hen1+/− and hen1−/− libraries, Tailor outperformed Bowtie and BWA and reproduced the published result that siRNAs, but not mRNAs, were subjected to tailing in the absence of hen1 (Supplementary Figure S1). Please note that, Bowtie and BWA in the speed test setting here were not capable of detecting non-templated tails. These tests were just used to compare their execution speed but not functionality.

To prove the accuracy of Tailor when confounding factors were not considered, we then used either Tailor or the Chen method to identify the non-templated tailing events (18). To achieve maximum speed of the Chen method to our best knowledge, we used the '-3k' option of Bowtie to clip k bases off from the 3' end of each read. This strategy avoided calling secondary programs and ensured that in all computational work was done other than Bowtie mapping. We started the alignment by setting k to 0. After the initial mapping, the unaligned reads were realigned with an incremented k (k = 1). This process was repeated four times. In the last iteration, four nucleotides were trimmed off from the 3' end (k = 4) and all the tailed reads should have been mapped at this point. In the simulation test, this method finished in 67 ± 1 s with Bowtie being called 47 times (k =
Not surprisingly, directly mapping by Tailor finished in 22 ± 1 seconds in the same computational environment. Both methods reported the same coordinates. However, in such setting, Chen method was not able to identify the tails, which required considerable computational work and time to retrieve from the raw reads. In contrast, Tailor revealed the length and the identity of the tails in the alignment output directly (see Supplementary Materials).

Performance with error tolerance

It is arguable that some NGS aligners that support local alignment, such as Bowtie2 (38) and BWA, can recover those tails with error tolerance. We simulated two datasets (one normal, one mutated, see below) whose distribution of read length follows that of the real small RNA sequencing dataset (43) (see Datasets in Materials and Methods' section; and also Supplementary Figure S2). For the normal dataset, two million reads were randomly sampled from the reference genome. We intentionally kept reads having just one unique occurrence in the genome and then appended a 1–4 nt non-template added tail on each read. For the mutated dataset, a similar procedure was used to generate another two million reads, but one additional step was added: we introduced one substitution in the nucleotides 2–8 of each read to simulate a RNA editing event as suggested by Vesely et al. (32). Again, this substitution was picked carefully to have only one occurrence in the genome with exactly one mismatch. The simulation guaranteed that there existed only one best alignment to the reference for each read in both datasets (see Datasets in Materials and Methods' section).

Then we examined the mapability of these datasets by Tailor (with -v option), Bowtie2 and BWA (see Figure 3C). Tailor clearly reported more unique mapping reads than others especially in the mutated datasets. When we looked closer to those reads that were mapped to multiple positions, we found Bowtie2 and BWA were more likely to align the tails to the reference than Tailor and create many alternative alignments. Note that the seed region setting was used to allow all three tools for the alignment (S = 20 and -v in Tailor and the equivalences in Bowtie2 and BWA; mismatches in the seed region were allowed) and all tools should try to align the first 20 nt of each read to the genome e, but Bowtie2 and BWA still generated suboptimal alignments. The execution time of these aligners with the error tolerant setting is depicted in Supplementary Figure S3. The complete commands for running all the tests are listed in Supplementary Materials.

We further checked whether the alignments and the tails were correctly reported. As shown in Figure 3D, Tailor was the only tool that gave satisfactory results reporting correct alignments and tails in the mutated dataset. There was no information in the output of BWA to recover the tails, and since most of the reads were aligned to multiple locations, it was expected that extensive post-processing would be needed for extracting the tails. The simulation clearly shows that Tailor is the only practical solution for doing tailing analysis with confounding factors.

Analysis pipeline

In order to provide a thorough and straightforward tailing analysis of deep sequencing libraries to the scientific community, we developed an easy-to-use pipeline (see Supplementary Materials).
Figure 3. Speed comparison between Tailor and other software. (A) Speed comparison between Tailor, BWA and Bowtie using simulated 18–23 nt s small RNA with (top) or without (bottom) non-matured tails. Tailor ran with the default setting, which allows no mis-match in the middle of the query. Tailored alignments were reported if perfect match could not be found. Bowtie ran with ‘-a–best s strata–v 0’ setting to allow no mis-match while report all best alignments. BWA ran with the default setting. Five different CPU settings were used and the running time was plotted. Three replicates were performed. (B) Speed comparison between Tailor, BWA and Bowtie (commands can be found in Supplementary Materials) using published small RNA Illumina NGS libraries from hen 1+/− and hen 1−/− mutants in fruity and zebra sh. Same settings were used as in (A). (C) The mappability of the normal (N) and mutated (M) datasets aligned by Tailor, Bowtie (with local alignment) and BWA. Multiple mapping was deemed as mis-alignments since each read was guaranteed to have only one occurrence in the reference. (D) The unique m apping reads shown in (C) were further examined to make sure they were aligned correctly and with proper tails reported (correct tails). Unique mapping reads that didn’t have correct alignment or tails were categorized another group (wrong tails/wrong alignment). The unmappable and multiple mapping reads were grouped together (undetermined or unmappable).
Applications--case studies

To prove the utility of Tailor, we applied Tailor to re-examine several publicly available small RNA sequencing datasets and revealed new facts about the data that has not been reported yet. In plants, HUA ENHANCER 1 (HEN1) methylates both mRNA and siRNA at their 3' ends to protect them from non-templated uridylation catalyzed by HEN1 SUPPRESSOR1 (HESO1), a terminal nucleotidyltransferase that favors uridine as substrate (18, 45). We applied Tailor on small RNA sequencing libraries from WT, hen1−/− and hen1−/−;heso1−/− cells of Arabidopsis and the results showed that siRNAs were subjected to both non-templated uridylation and cytosylation without HEN1 while mRNAs were mainly subjected to uridylation. Furthermore, the loss of HESO1 only reduced the uridylation but not cytosylation of siRNAs, suggesting the existence of an additional nucleotidyl transferase that prefers cytosine as substrates (Figure 4A).

We then applied Tailor to two NGS libraries that cloned Ago2 associated small RNA from nuclear and cytoplasmic fraction of Hela cells respectively (46). Since RNA were cloned using poly-A polymerase instead of 3' adaptor ligation in the library preparation, A-tails were unable to be re-
covered to a putationally. Although most mRNAs showed a very similar length distribution and tailing frequency between these two samples, one mRNA, mR-15a, exhibited a distinct pattern. In cytoplasm, mR-15a was only 21 nt long and had mOES.U tailing for its 22-mer isoform. Surprisingly, in the nuclear fraction, mR-15a peaked at 22 nt and showed strong U tailing (Figure 4B). In addition, mR-15b, which shares its seed sequence with mR-15a and only has one nucleotide difference from mR-15a in the first 19 nt of its mature sequence, did not exhibit obvious variation between the two samples. This suggests that, either 9-12 nt, also known as the ‘central site’ or the 3' end of guide miRNA play an important role in tailing regulation.

Finally, we applied Tailor to study the possible relationship between RNA editing and tailing in miRNAs. The miRNA libraries were constructed from the whole brain tissue cells dissected from Adar2−/− and wild-type mice (32). A day2 is known for its strongest effects on miRNA abundance and editing among the three isoforms of ADARs (47). One of the highly expressed ADAR substrates, mR-379, was shown to be directly edited at the nucleotide within the seed region and about half of the mature mR-379 were edited by ADAR 2 (32). As expected, the edited form of mR-379 (i.e. mR-379-5G) was greatly reduced in Adar2−/− mice. To evaluate the effects on miR-379 on miRNA stability change, we used Tailor to study the potential role of miRNA-Argonaute or tailing (1,48).

The importance of miRNA in regulating the function, processing and stability of many small RNAs. Consistent with the above results, Tailor by comparing it with the Chen method with simulated reads and found that they generated exactly the same results. We applied Tailor only used a third of the time to align and provided the same information in a very similar length distribution and tailing frequency between the two samples. This suggests that, either 9-12 nt, also known as the ‘central site’ or the 3' end of guide miRNA play an important role in tailing regulation.

Finally, we applied Tailor to study the possible relationship between RNA editing and tailing in miRNAs. The miRNA libraries were constructed from the whole brain tissue cells dissected from Adar2−/− and wild-type mice (32). A day2 is known for its strongest effects on miRNA abundance and editing among the three isoforms of ADARs (47). One of the highly expressed ADAR substrates, mR-379, was shown to be directly edited at the nucleotide within the seed region and about half of the mature mR-379 were edited by ADAR 2 (32). As expected, the edited form of mR-379 (i.e. mR-379-5G) was greatly reduced in Adar2−/− mice. To evaluate the effects on miR-379 on miRNA stability change, we used Tailor to study the potential role of miRNA-Argonaute or tailing (1,48).

ACKNOWLEDGEMENT

We thank the members of the Hung, Weng and Zamore laboratories for helpful discussion and critical testing.

FUNDING


REFERENCES

8. van O 8ewinkelu, C., Barym, J.-M., Batista, P.J., M. Elb, C.C., Bezr...