5-24-2015

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

Min-Te Chou  
National Chiao Tung University

Bo W. Han  
University of Massachusetts Medical School

Chiung-Po Hsiao  
National Chiao Tung University

See next page for additional authors

Follow this and additional works at: http://escholarship.umassmed.edu/gsbs_sp

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

Repository Citation
Chou, Min-Te; Han, Bo W.; Hsiao, Chiung-Po; Zamore, Phillip D.; Weng, Zhiping; and Hung, Jui-Hung, “Tailor: a computational framework for detecting non-templated tailing of small silencing RNAs” (2015). GSBS Student Publications. 1882.  
http://escholarship.umassmed.edu/gsbs_sp/1882

This material is brought to you by eScholarship@UMMS. It has been accepted for inclusion in GSBS Student Publications by an authorized administrator of eScholarship@UMMS. For more information, please contact Lisa.Palmer@umassmed.edu.
Tailor: a computational framework for detecting non-templated tailing of small silencing RNAs

Authors
Min-Te Chou, Bo W. Han, Chiung-Po Hsiao, Phillip D. Zamore, Zhiping Weng, and Jui-Hung Hung

Comments
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

Creative Commons License
This work is licensed under a Creative Commons Attribution-Noncommercial 4.0 License

Rights and Permissions
Citation: Nucleic Acids Res. 2015 May 24. pii: gkv537. Link to article on publisher’s site

This article is available at eScholarship@UMMS: http://escholarship.umassmed.edu/gsbs_sp/1882
Tailor: a computational framework for detecting non-tem plated tailing of small silencing RNAs

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

1Institute of Bioinformatics and Systems Biology and Department of Biological Science and Technology, National Chiao Tung University, HsinChu, Taiwan, 300, Republic of China, 2RNA Therapeutics Institute, Howard Hughes Medical Institute, and Department of Biochemistry and Molecular Pharmacology, University of Massachusetts Medical School, Worcester, MA 01605, USA and 3Program 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-tem plated 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 it 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/jh Hung/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 complexes (RISC), which uses the small RNA guide to find its regulatory targets and reduce gene expression. Although the studies on the biogenesis of all 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-tem plated addition to the 3' end of small silencing RNAs, namely tailing, could play essential roles in this regard. Non-tem plated 3' mono- and oligo-uridylation of the pre-microRNA (pre-miRNA) regulates miRNA processing by either preventing or promoting Dicer cleavage in invertebrates (5–7). The 3' mono-uridylation on small interfering RNAs in Caenorhabditis elegans is associated with negative regulation (8). Ame et al. have demonstrated that highly complementary targets trigger the tailing of miRNAs and eventually lead to their degradation in flies and mammals (9,10); a similar mechanism has been found in some endo-siRNAs as well (11). Identification of tailing events not only suggests the co-evolution of all 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-tem plated nucleotides from millions of sequencing reads is challenging. The teaming group used MegaBLAST to align pRNA sequences...
to the genome and relaed on post-processing the reported
mismatches to gain insights into tailing (8). However, as a
heuristic algorithm, BLAST is not guaranteed to nd all
the tailing events (22,13) and it is signi cantly slower than
the NGSA aligners, like MAQ (14), BWA (15), Bowtie (16)
and SOAP2 (17). The Chen group used an accurate method
that iterates between Bowtie alignment and 3 clipping of
unmatched reads (18) to nd all the perfect alignments of
trimmed reads. A similar approach has been used for re-
moving erroneous bases at 3 end to increase the sensitivity
of detecting miRNAs (19). Let alone that this method in-
evitably multiples the running time by them axial length of
tails, extra computational works are still needed to retrieve
the identity of each trimmed tail. The study by Amers et al.
used a specialized siRNA tree data structure to ef ciently
nd all the tails without sacri cing the accuracy. However, dueto
the high memory footprint of the siRNA tree data structure,
which is about 16 to 20x of the genome size, the read
mapping has to be performed for each chromosom e sepa-
rate(9,20). Extra processing is still required to nalize the
alignment errors from all chromosom es.

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 tHm ina HSeq and G enec on anal-
alyzer platform s have preferential A C conversions (21,22)
and a high error rate at the 3 end of reads, which frequently
leads to uncodable bases, i.e. B tails (23,24). In addition to
these technical artifacts endued by the sequencers, RNA
tailing is another common post-transcriptional modi-
cation in all silencing RNA biology that could perplex the
tools with erroneous alignment entries. There are two major
types of RNA editing in mammals, adenosine to inosine (A
B I) and cytidine to uridine (C U) editing. The major en-
zyme that catalyzes adenosine to inosine is the adenosine
deaminases acting on RNA (ADARs), whose main sub-
strates are RNA s with double-stranded structures (25–27).
Since any small silencing RNA s are originated from struc-
tural RNAs, they are all likely targets of A C editing (28–30).
Recent studies have shown that A C editing can occur
on the seed region of the miRNAs with fairly high
occurrence rate (up to 80% in some cases) and have a direct
impact on the selection of their regulatory targets (31,32).
Those unmatched bases degenerate the sensitivity and ac-
curacy of short read alignment and have a negative effect
on the detection of tailing.

Most of the current methods simply ignore those con-
 founding factors and rely on adapting existing, less special-
ized tools with extensive post-processing and as a con-
sequence the perform ance, usefulness and application of tail-
ning analysis is seriously compromised. A fast, accurate and
straightforward approach to study tailing is still in need. To
ease the cost of performing tailing analysis with dram ati-
cally increasing sequencing throughput, we here introduce
Tailor—a fast tool that preprocesses and maps sequences
to a reference, distinguishes tails from mismatches or bad
alignments with a novel algorithm and reports both perfect
and trimmed alignments simultaneously without loss of infor-
mation. Tailor is capable of analyzing the non-tem plated
tailing forms of siRNAs without tailed reads using a novel algo-
rithm and reports both perfect and trimmed alignments
simultaneously without loss of information. Tailor is capable
of analyzing the non-tem plated tailing forms of siRNAs and pro-
duces publication-quality summary files. In addition, to
better demonstrate the utility of Tailor, we reanalyzed pub-
lished datasets with Tailor and unearthed several interesting
observations (see Applications—case studies in Results). Al-
though the ndings still require thorough experimental val-
idation, it is clear that Tailor would help expand the scope
of the study of all silencing RNAs.

MATERIALS AND METHODS

Datasets

T. h m ina sequencing data of all miRNAs from Drosophila
melanogaster (SRR029630, SRR029633), D. an is rib Hem (SRR363984–5), A. rabi topsis hemis and (SRP010683) and Ago2 associated small RNAs in cyto-
plasm (SRR529097) and nuclear fraction (SRR529100) of
HeLa were obtained from NCBI Sequence Read Archive.
The length distribution of the simulated confounded reads
was from the D. melanogaster Ago3 associated small RNAs
extracted from ovaries (SRR916073). In-house program
was used to trim the 3 adaptors and ler the reads with low quality. Randomly distributed reads from fruit
fly genome was generated by Arti cia lFastG ene rator (33).
Ten millions reads were randomly chosen using seqtk
(github.com/lh3/seqtk.git) with options 'sam pple-s100
10000000'. To remove multiple mapping reads in somesim-
ulation datasets, we used Bowtie iteratively before and after
the trimming and den mutation to assure each read has only one occurrence in the reference.

Rationale

The principle of detecting non-tem plated bases at the 3
end of reads is basically to nd the longest common pre x
(LCP) between the read and each of the suf xes of the refer-
ence and then report the read as a tail. Given a read R (M base pairs [bp] long) and all the suf xes (Si) of a reference sequence G (N bp long), one can nd the
LCP between R and Si by nding the longest consecutive
matches from the 1st base to the last. Since there are totally N suf xes of G, a trivial solution needs at worst M N
times of comparison to nd the LCP of R and G; however
the perform ance is unacceptably slow when G is as large as a human genome. Using index structures, such as the suf x
tree or suf x array, nding LCPs between the NGS reads
and the reference can be solved much more ef ciently (34).

Recently, the Full-text Index in M inutespace (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 bytes per base
to store the index. A more detailed introduction of build-
ing 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
detecting tails. One straightforward solution is to align reads
without those non-tem plated bases by repeatedly rem oting
one last base in each round of the alignment process until
at least one perfect hits is found (18), but the approach sacri ces
the speed greatly and requires extensive post-processing.
To bene t from the space and time ef ciency of the FM Index,
we further developed our 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 workflow 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 coordinate of each alignment should be calculated accordingly. To allow searching against both strands simultaneously and im proves the speed, Tailor concatenates the plus and minus strands of the reference and constructs an index instead of two (Figure 1A and Supplementary Material a), 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 next shift for getting the coordinate of each occurrence. Any alignment whose prefix matching portion exceeds the boundary of the mapped chromosome is deemed. The searching continues until either the mismatch encountered all the characters of the query to the reference (i.e., the perfect matching) or no more bases can be matched (i.e., the prefix-matching). In the latter case, Tailor backtracks to the previous matched position and exhaustively enumerates all the possible prefix matches. The unmapped part remains 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 read as a 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 selection rule (S = 18 by default) bases at its 5' portion is deduced 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 m is match will be reported (Case 4). Note that in order to differentiate tails from sequencing error, a filtering 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 circumsence that the tail 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 m utant with a defective tailing pathway).

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

**In silico detection**

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 one long reference, whose length could exceed the maximum 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 compatible with the algorithm introduced in Aameres 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 Sa ligners and reports alignment in the SAM (41) format. A tail is described as ‘soft-clipping’ in CIGAR and the sequences are reported under ‘TL’ Z:’ in the optional elds. Mismatch is allowed (−v), will be reported in the M D’ tag (see Supplementary Material a), 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 Bowtie used is 0.7.5a-r405. The version of Tailor used is 1.0.0. A ll commands and all the tests are listed in the Supplementary Material a.

**RESULTS**

**Perform ance 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 independently generated 10 million perfectly genome-matching reads from the D. melanogaster genome (simulated tail-free dataset) (33) and randomly appended 1–4 genome-earmarked nucleotides to the 3' ends (simulated tailing
dataset). 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 all 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 it 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−/− libraries (Supplementary Figure S1). Please note that Bowtie and BWA in the speed test setting here were not capable of detecting non-tailed reads. 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-tailed 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 workload 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 mished in 67 ± 1 s with Bowtie being called 24 times (k = 4).
Figure 2. Error tolerance filtering rules. (A) Reads would have to be reverse-complemented before searching. The corresponding seed region is highlighted in green. (B) Eight rules for determining tails. See the main text for more details.

Then we examined the mapping accuracy 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 aid all these tools for the alignment (S = 20 and −v in Tailor and the equal settings in Bowtie2 and BWA; mismatches in the seed region were allowed) and all tools should try to align the next 20 nt of each read to the genome e, but Bowtie2 and BWA still generated suboptimal alignments. The execution time of three 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 a pipeline that allows one to conveniently implement different parameters and settings for the tailing analysis. The pipeline is designed to be flexible and modular, allowing researchers to customize the analysis according to their specific needs and data characteristics. It includes several key steps: 0–4). Not surprisingly, directly mapping by Tailor finished in 22 ± 1 seconds in the same computational environment. Both methods reported the same e coordinates. However, in such setting, Chen method was not able to identify the tails, which requires 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).
**Figure 3.** Speed comparison between Tailor and other software. (A) Speed comparison between Tailor, BWA and Bowtie using simulated 18-23 nt small RNA with (top) or without (bottom) non-templated tails. Tailor ran with the default setting, which allows no mismatches in the middle of the query. Tailed alignments were reported if perfect matches could not be found. Bowtie ran with '-a' and '−best−strata−v 0' setting to allow no mismatches while reporting 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 (command lines can be found in Supplementary Materials) using published small RNA Illumina NGS libraries from hen1+/− and hen1−/− mutants in fruit fly and zebrafish. Same settings were used as in (A). (C) The mappability of the normal (N) and mutated (M) datasets aligned by Tailor, Bowtie2 (with local alignment) and BWA. Multiple mapping was deemed as misalignments since each read was guaranteed to have only one occurrence in the reference. D) The unique mapping 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 into another group (wrong tails/wrong alignment). The unmappable and multiple mapping reads were grouped together (undetermined or unmappable).

In the community, we developed the interface of Tailor to take FastQ files as input and produce publication-ready figures. The overview chart of the pipeline is summarized in Supplementary Figure S4A. In brief, the input reads, with barcodes and adaptors removed, are subject to a quality trimming step based on a PHRED score threshold provided by the user (e.g. to get rid of B-tails). The pipeline then applies Tailor to align the high-quality reads to the reference. The information on the length and identity of tails are then retrieved from the SAM format output and summarized to a tabular text file. Additionally, the alignments are assigned to different genomic features in small RNAs, exons, introns, etc.) using BEDTools (44). Tails from different categories are summarized in the pipeline. Publication-quality figures depicting the length distribution are drawn using R packages ggplot2 (23) (Supplementary Figure S4B). The pipeline also offers isoRNA specific analysis. Balloon plots describing the 5’ and 3’ relative positions and the tails length are provided for a comprehensive overview (Supplementary Figure S4C).
Applications—case studies

To prove the utility of Tailor, we applied Tailor to re-analyze 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-template uridylation catalyzed by HEN1 SUPPRESSOR1 (HESO1), a terminal nucleotidyl transferase 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-template 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 additional nucleotidyl transferase that prefers cytosine as substrates (Figure 4A).

We then applied Tailor to two NGS libraries that cloned Ago2 associated smallRNA from nuclear and cytoplasmic fraction of HeLa cells respectively (46). Since RNAs were cloned using poly-A polymerase instead of 3’ adaptor ligation in the library preparation, A-tails were unable to be re-
coved by miR-379. Although most miRNAs showed a similar length distribution and tailing frequency between these two samples, one miRNA, miR-15a, exhibited a distinct pattern. In cytoplasm, miR-15a was mostly 21 nt long and had a distinct U tailing for its 22-nt isoform. Surprisingly, in the nuclear fraction, miR-15a peaked at 22 nt and showed strong U tailing (Figure 4B). In addition, miR-15b, which shares its seed sequence with miR-15a and only has one nucleotide different from miR-15a in the 5'-9 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 3-day-old Aדר mice (22). Aдр is known for its strongest effects on miRNA abundance and editing and among the three isoforms of Aдрs (22), one of the highly expressed Aдр substrates, miR-379, was shown to be directly edited at the nucleotide within the seed region and about half of the mature miR-379 were edited by Aдр2 (22). As expected, the edited form of miR-379 (i.e. miR-379–5G) was greatly reduced in Adar2–/– mice. Surprisingly, we found that the normal miR-379 has much more tailing than miR-379–5G (see Figure 4C). Monon- and poly-A tails (the bluish portion) were depleted in miR-379–5G, which makes the probability that Aдрs and the A-to-I editing could affect the affinity between the miRNAs and the unknown enzymes responsible for adenylating the 3' end. Since the proportion of different types of tails was unchanged upon Aдр2 knockout, the tailing machinery is less likely modulated by Aдр2 directly but by the subsequent factors after editing in the seed, such as differential targeting, RNA stability change or miRNA-Agonante sorting (48).

DISCUSSION

Tailing is a molecular phenomenon that associates with the function, processing and stability of many small RNAs. Computational identification of the tailing sequences from the millions of NGS reads has been proven to be challenging and time-consuming. We herein present a tailing analysis framework, Tailor, which aligns reads to the reference genome, reports tailing events simultaneously and visualizes analysis results. We assessed the accuracy of Tailor by comparing it with the Chen method with simulated reads and found that they generated exactly the same results. While Tailor only used a third of the time to align and provided more information compared to the alternative.

When confounding factors were ignored, Tailor was not slower than other well-known fast general-purpose mapping tools in our tests. We demonstrated that Tailor executed in a speed that was very close to the regular Bowtie and BWA. While providing more functionalities for detecting tailing events, when confounding factors were presented in the reads, it was argued that advanced NGS aligners that support the local alignment mode (e.g. Bowtie2) could be competent in nailing tails, but we tested them with simulated reads and showed that Tailor performed significantly better in both accuracy and efficiency.

Tailor’s shell-based framework takes raw reads as input and produces comprehensive tailing analysis results and publication-quality output. We reproduced known conclusions drawn from the published tailing study by the pipeline with little extra scripting and post-processing. We also applied the pipeline to other datasets and shed light on other possibilities of the functional roles of tailing, such as involving in RNA processing, transport, decay and storage by interacting with other RNA binding proteins (49).

Our aim is to design Tailor to reduce the cost of doing tailing analysis and reinforce or replace the conventional computational procedure in analyzing all short non-coding RNAs. We expect that Tailor could be applied to a broader scope and subsequently facilitate the understanding of biological processes related to tailing.

AVAILABILITY

Source code as an Open Source project: http://jhung.github.io/Tailor.

SUPPLEMENTARY DATA

Supplementary data are available at NAR Online.

ACKNOWLEDGEMENT

We thank themembers of the Hung, Wang and Zamore laboratories for helpful discussions and critical testing.

FUNDING


REFERENCES
