Reconstruction of The RNA Hyper-Editing Detection Tool

Gili Wolf[1]

[1] *The Mina and Everard Goodman Faculty of Life Sciences, Bar-Ilan University,Ramat-Gan, 52900,Israel*

(Dated: September 22, 2024)

I Introduction

First RNA hyper-editing detection tool was developed by Hagit T. Porath in 2014[1]. The tool was the first to examine unmapped reads from RNA-Seq experiments, in order to find reads with excessive ('hyper') editing, which do not easily align to the genome. This method, discovered highly edited sites that were previously screened. However, the original tool had several limitations, including reliance on *BWA*, which lacked support for splicing, was computationally intensive, and had slower performance with reduced parallel processing capabilities. The current project aims to reconstruct and enhance this tool to overcome these limitations by incorporating more efficient alignment methods, advanced parallel processing capabilities, and improved modularity and portability features.

II Biological Background

1 **dsRNA.** Endogenous dsRNA in human cells comes from several sources, including Alu repeats, endogenous retroviruses (ERVs), long interspersed nuclear elements (LINEs), and natural antisense transcripts. Alu repeats are the most common, making up to 10% of the human genome. These repeats can generate dsRNA, especially when activated by stressors such as viral infections or heat shock. ERVs, which resemble exogenous retroviruses, and LINEs also contribute to dsRNA formation, influencing immune responses and disease mechanisms. Additionally, Natural antisense transcripts can create dsRNA when they overlap with sense transcripts,which can affect gene regulation, for example, by inhibiting gene expression or impacting mRNA stability [2]. dsRNA is recognized by several proteins with distinct roles, including MDA5 and the ADAR protein family.

2 **Cellular Anti-Viral Response.** To prevent viral attacks on the cell, one of the primary cellular antiviral responses is controlled by MDA5. As mentioned above, This protein binds to dsRNA [3], a strong indicator of viral infection. Upon binding, MDA5 forms helical structures along the dsRNA, triggering a signaling pathway that leads to the production of Type I interferons, which are crucial for the antiviral response [4]. However, endogenous dsRNA can also trigger this response, potentially leading to inappropriate activation of the antiviral pathway (as shown in 1). To mitigate this, cells employ adenosine-to-inosine (A-to-I) editing on endogenous dsRNA, reducing its recognition by MDA5. [5].

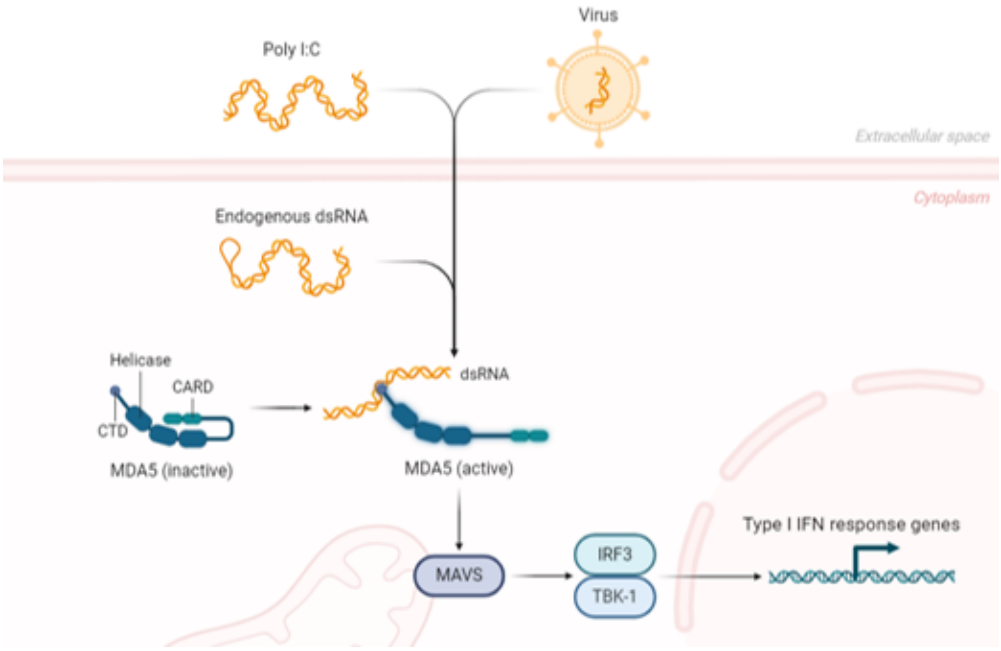

FIG. 1. Illustration depicting the process of MDA5 activation by both viral and endogenous DNA. Created using BioRender.

RNA editing. RNA editing is an evolutionary conserved process which was found in most studied living organism. This process is mediated by several intercellular proteins, each creating distinct modifications in RNA bases. Our focus will be on the most common adenosine-to-inosine (A-to-I) modification, catalyzed by the double-stranded RNA-specific adenosine deaminase (ADAR) protein family [6].The ADAR family includes three main members: ADAR1, ADAR2, and ADAR3. ADAR1 is known for its role in editing dsRNA and is induced by interferons, while ADAR2 primarily targets neurotransmitter receptor transcripts, and ADAR3 is expressed in the brain but lacks catalytic activity, serving mainly to inhibit ADAR2. The ADAR proteins (1 & 2) bind to double-stranded RNA and employe deamination to convert A-to-I (2), later detected as A-to-G, both in the cellular environment and in next generation sequencing (NGS). This editing process aims to weaken the formation of dsRNA, disrupt its secondary structure, and prevent dsRNA from binding to MDA5, thereby avoiding an antiviral response. Notably, It is known that ADAR tends to edit sites in clusters [7], leading to the formation of hyper-edited regions.

III Computational Background

Hyper-Editing Detection. Identifying RNA editing sites ideally involves aligning RNA-seq reads to the reference genome and searching for mismatches that indicate editing sites (A-to-G). However, this process is complicated by several factors. Sequencing errors and genetic polymorphisms can be misinterpreted as editing sites, but more critically, the presence of editing

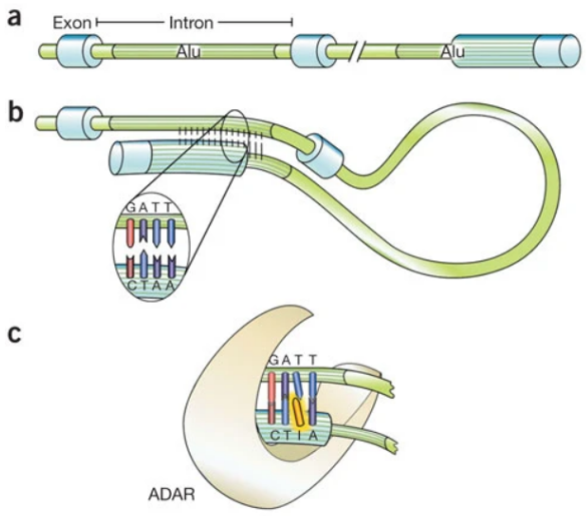

FIG. 2. ADAR protein

sites and other mismatches can hinder accurate alignment of RNA-seq reads to the genome. Current alignment methods typically allow only a limited number of mismatches between reads and the genome. Allowing more mismatches can lead to lower alignment scores, multiple alignments, or complete alignment failure. Consequently, many hyper-edited sites, which result from ADAR's tendency to edit in clusters, remain undetected as these reads are often categorized as unmapped by most conventional aligners.

Original Tool. To address the challenge of detecting hyper-editing in RNA-seq data, a novel approach was published by Hagit T. Porath, Shai Carmi, and Erez Y. Levanon in 2014 [1]. Their method followed a few-step approach (as shown in 3):

1. Initial reads alignment.
2. Collect all unmapped reads from the initial alignment.
3. Transform all As to Gs in both the unmapped reads and the reference genome.
4. Realign the transformed RNA reads to the transformed reference genome.
5. Recover the original sequences and search for dense clusters of A-to-G mismatches.

This pipeline significantly improved the detection of hyper-edited reads. When applied to the Illumina Human BodyMap 2.0 dataset, it identified 390,881 hyper-edited reads, containing 455,014 unique A-to-G-editing sites, providing a more comprehensive detection of editing sites compared to previous methods.

IV Project's Aim

The project aim is the reconstruction of the original tool, which serves three main goals. Firstly, to change the original Burrows-Wheeler Aligner (BWA) to the Spliced Transcripts Alignment to a

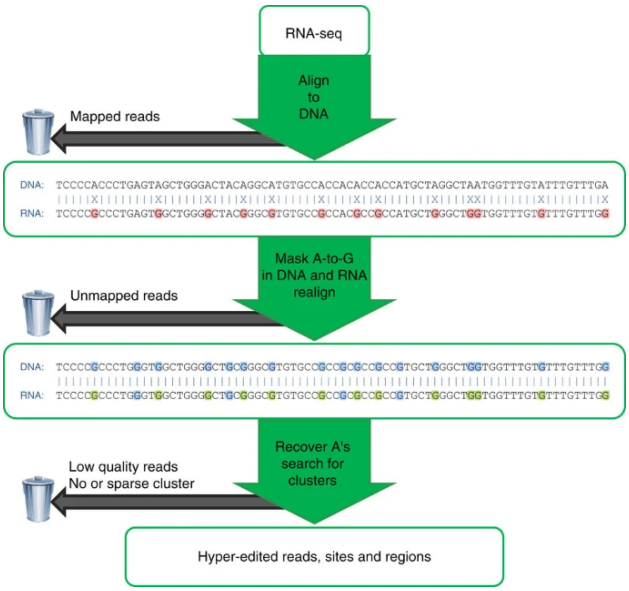

FIG. 3. Original Wf

Reference (*STAR*), which is more suitable for RNA-seq data. Secondly, re-write the tool's code in order to optimize workflow management, parallel processing and modularity of the tool, hopefully to lead for improved performance. Finally, enable mobility and portability across various computing environments, providing easy access to the tool's users across the globe.

Aligner. In the original tool, the *BWA* aligner [8] was used as it was considered the best aligner available at the time. However, *BWA*'s lack of support for splicing presented significant limitations, particularly for RNA-seq data where accurate detection of splicing events is crucial. Splicing is a fundamental process in eukaryotic gene expression where exons are joined together after introns are removed, generating mature mRNA. Accurate alignment of RNA-seq reads must account for these splicing events to correctly interpret gene structure and expression. The *STAR* aligner [9], unlike *BWA*, supports splicing and efficiently handles complex transcript structures, leading to greater potential for identifying hyper-edited reads. Furthermore, *STAR*'s ability to enable memory sharing allows for parallel processing, addressing BWA's slow processing speed and high resource consumption. This shift to *STAR* not only enhances the accuracy of splicing detection but also improves overall alignment efficiency.

Modularity. The original tool for detecting hyper-edited reads operated as a monolithic system, processing data from start to finish without the ability to break down or isolate different components. In contrast, the new tool is built with improved modularity, consisting of three distinct parts (4): *Genome Transformation*, *Pre-Analysis & Detection*. Each part functions independently but integrates smoothly with the others. This modular approach not only enhances flexibility and maintainability but also allows for easier updates and optimization of individual parts, ultimately leading to more effective and adaptable detection of hyper-edited reads. Detailed information about each component will be provided in the Methods section.

FIG. 4. New Pipeline's Main Parts

8 **Parallel Processing.** The new tool is structured using *Nextflow*, a Domain-Specific Language (DSL) designed to optimize the creation of scalable and reproducible scientific workflows. *Nextflow* significantly enhances parallel processing capabilities. Each component of the tool is developed from numerous scripts in different languages, such as Python and Bash. *Nextflow* integrates these scripts, manages inputs and outputs, and ensures efficient execution of parallel processing tasks, even for large datasets. This improvement is especially crucial because the previous BWA-based tool was too resource-intensive to allow for effective parallel processing.

9 **Portabilty.** The new tool significantly improves portability compared to the original, which was limited to the lab's servers. By incorporating *Docker*, the tool is no longer dependent on the local computing environment, ensuring consistent performance across different systems. This containerization makes it easier to deploy and run the tool on various platforms. Additionally, the tool is now available on *GitHub*, facilitating broader access and collaboration. Future plans include making it available on *AWS* Cloud, further enhancing its accessibility and scalability for users.

V The Importance of the Research

10 **Research on RNA editing.** RNA editing is a crucial cellular process involving the post-transcriptional modification of RNA molecules. This modification significantly impacts gene expression and cellular function in various ways. Besides its primary role in inhibiting cellular antiviral responses, as detailed in the *Biological Background* section, RNA editing has other important biological implications. For example, it alters proteins' amino acid sequences, affecting their function and stability, and facilitates the generation of diverse protein isoforms [10].

Additionally, RNA editing plays a major role in RNA regulation within the cell. It influences the 3' untranslated regions (UTRs) of mRNA transcripts, affecting mRNA stability, localization, and translation efficiency, and can modify miRNA binding sites, which are crucial for gene regulation. RNA editing also impacts circular RNAs (circRNAs) by disrupting their formation, thereby influencing the regulatory networks involving these molecules. [11–13]

Identifying additional RNA editing sites is essential for deepening our understanding of this complex phenomenon and its functional implications. This knowledge not only enhances our grasp of cellular processes but also opens avenues for clinical and therapeutic applications. For instance, targeted manipulation of RNA editing mechanisms could correct mutated RNA sequences, offering potential strategies for treating genetic disorders and developing innovative therapies at the molecular level [14].

11 **Relevance to Autoimmune Diseases.** Autoimmune diseases present a significant challenge in medical research due to their complex and largely unknown etiology, which complicates the development of effective treatments. Recent laboratory research has revealed that loss of ADAR activity in pancreatic beta cells leads to phenotypes similar to type 1 diabetes [15]. This finding underscores the critical role of RNA editing in maintaining pancreatic function and highlights potential connections between RNA editing and autoimmune diseases.

Understanding how RNA editing influences immune system regulation and disease pathogenesis is crucial for advancing our knowledge of these conditions. Further investigation into RNA editing could offer valuable insights into their underlying mechanisms and pave the way for new therapeutic approaches.

To support this research, developing improved, faster, and more accessible hyper-editing tools is essential. Such tools would enable more detailed and comprehensive studies, revealing significant insights into RNA editing's role in autoimmune diseases. By enhancing our understanding of these connections, we can advance novel diagnostic and therapeutic strategies targeting the molecular mechanisms underlying autoimmune disorders.

VI Results

We utilized the *Hyper-Editing* tool to analyze RNA-Seq data from the Genotype-Tissue Expression (GTEx) project. The GTEx project is a comprehensive public resource designed to study human gene expression and regulation in relation to genetic variation across diverse tissues and individuals. For this analysis, we selected 10 random samples from four tissues: Artery Aorta, Brain Cerebellum, Whole Blood, and Muscle Skeletal. These tissues were chosen to represent varying levels of RNA editing.

The results section will primarily present a basic comparison between the original and new tools, focusing on hyper-edited regions and processing time. However, the main emphasis of the project was the development of the new tool. A more comprehensive analysis is needed to thoroughly understand the differences in the results. The comparison is complex due to several factors, such as the transition to the STAR aligner, changes in the processing workflow, and modifications to the filtering criteria, including the introduction of fastp for preprocessing.

12 **Number of Hyper-Edited Clusters.** The new *Hyper-Editing* tool produces a significantly smaller number of hyper-edited clusters in the final results. For instance, in the sample GTEX-1212Z_ArteryAorta_GTEX-1212Z-0826-SM-5EQ51, the original tool identified 73,694 clus-

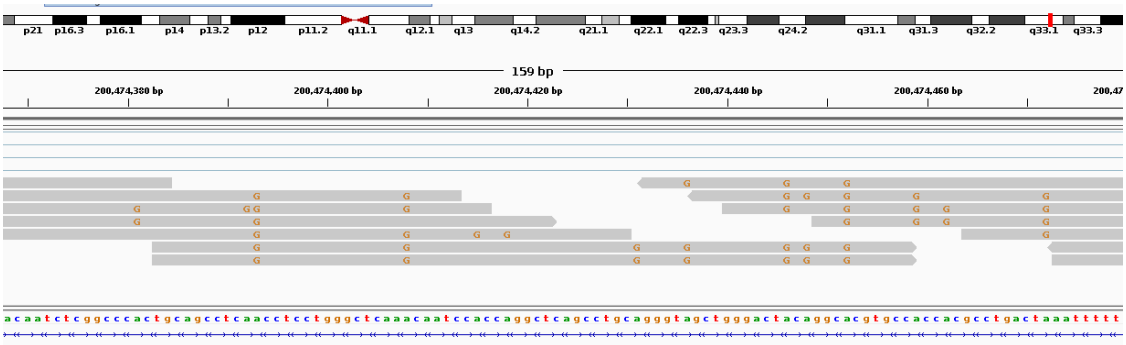

FIG. 5. IGV view of a hyper-edited region in Artery Aorta, identified by both the original and the new tool.

ters, whereas the new tool identified only 56. This reduction is likely due to several factors: (1) reads are discarded early in the process by fastp during the initial quality filtering step, (2) the first alignment yields only about 25% of the previously unmapped reads compared to the original tool, likely because STAR aligner is more accurate and better suited than BWA, which was used in the original tool, and (3) The original tool processed all reads as single-end (SE) throughout the entire workflow, whereas the new tool is designed to handle paired-end (PE) reads. This distinction introduces an additional challenge in aligning both reads during the second (transformed) alignment phase. Notably, when the new tool was tested in SE mode using PE data, we observed a higher abundance of results, further supporting the impact of handling paired-end reads.

13 **Hyper-Edited Regions.** To analyze the regions containing hyper-edited clusters identified by the new tool, we merged the BAM output files from the first and second alignments using samtools. We visualized the genomic positions of the suggested clusters using IGV (Integrative Genomics Viewer). Our goal was to find regions with high coverage where reads exhibit distinct editing patterns, supporting the hypothesis that not all reads undergo editing at all possible editing sites. In the Artery Aorta tissue, we found such a region, which was identified by both the original and the new tool (5).

Utilizing Bedtools intersect with the `-v` option, we observed that most samples exhibited novel hyper-editing clusters identified by the new tool, which were not detected by the original tool. An example of a uniquely identified region found by the new tool in the Brain Cerebellum tissue is shown in Figure 6.

14 **Tissue-Specific Hyper-Editing Levels.** Four distinct tissues were selected for this study to capture a range of RNA editing levels: Brain Cerebellum, Artery Aorta, Whole Blood, and Muscle Skeletal. The variation in editing levels across these tissues can be attributed to several biological factors. For instance, ADAR2, an enzyme responsible for A-to-I RNA editing, is highly expressed in the brain, where it edits transcripts critical for proper neuronal function and activity [16].

The results from the new tool align with these biological expectations, identifying 237 hyper-

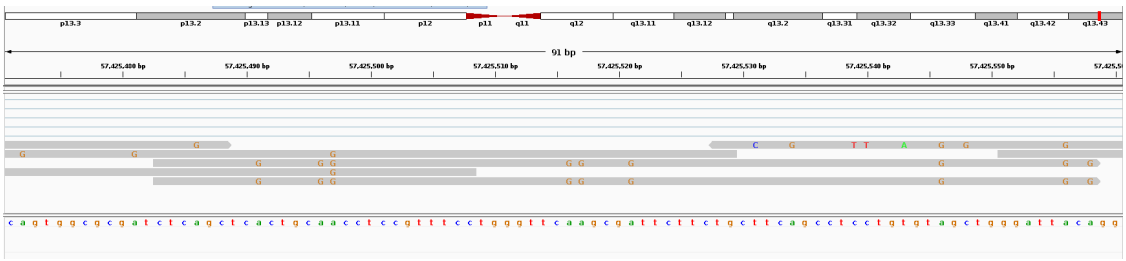

FIG. 6. IGV view of a hyper-edited region in Brain Cerebellum, uniquely identified by the new tool.

edited reads in Brain Cerebellum and 198 in Artery Aorta. In contrast, Whole Blood and Muscle Skeletal tissues, which typically exhibit lower ADAR activity, showed significantly fewer hyper-edited reads, with 58 and 19 respectively. These findings reflect both the tissue-specific activity of RNA-editing enzymes and the sensitivity of the new tool in detecting hyper-editing events.

Run Time. One of the main aims of the new tool is enhance efficiency, particularly by trying to reduced run time. By utilizing Nextflow's built-in reporting capabilities, we analyzed the run time of *Pre-Analysis* across five samples for each tissue (excluding the indexing of the second mapping results). The jobs completed within 6 to 11 hours, depending on various factors such as sample size, editing levels, and CPU load at the time of execution.

This marks a substantial improvement, as the original tool required approximately 20 to 24 hours to process the same number of samples. While these preliminary results demonstrate a clear enhancement in speed, further analysis is required to fully assess the factors influencing run time and to confirm the robustness of these improvements across different environments and datasets.

Editing in Other Organisms. One of the key improvements introduced by the new tool is its modularity, allowing each part to be executed independently. Specifically, this feature enables the independent execution of *Detection*, where users can input outsourced BAM files that were not generated by *Pre-Analysis*. This flexibility has facilitated new research initiatives, including a lab project focused on identifying RNA editing in other organisms.

Previous studies have shown A-to-I RNA editing across different organisms, and even across all the Metazoa [17]. Since not all RNA editing is carried out by ADAR enzymes, and because editing varies depending on the RNA-editing enzymes expressed in each organism, other types of RNA-editing might be performed in other organisms. Thus, the outsourced BAM files are analyzed for potential editing across all 12 base combinations. An ongoing project in the lab is investigating RNA editing in venomous organisms, where preliminary results have already identified approximately 780,000 potentially G-to-C edited clusters. This exploration highlights the broader applicability of the tool in studying RNA editing beyond human systems.

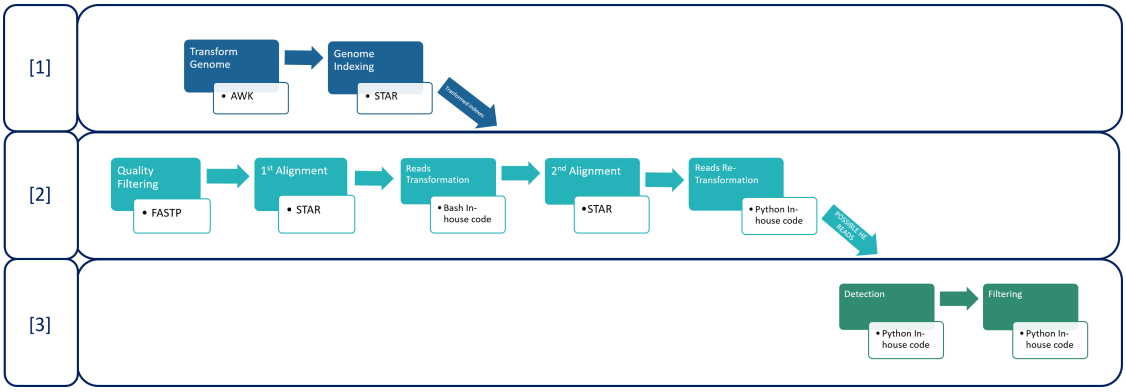

FIG. 7. Pipeline Overview

VII Methods

The new *Hyper-Editing* tool offers a built-in pipeline for identifying hyper-edited reads, supporting both single-end and paired-end data. It is designed to provide maximum modularity, allowing users to adjust each of the pipeline parameters.

17 **Over All design.** We constructed the outline of the pipeline for the *Hyper-Editing* tool using *Nextflow* [18], a Domain Specific Language (DSL) that enables scalable and reproducible scientific workflows.The design consists three main parts: *Genome Transformation*, *Pre-Analysis* & *Detection*. Each part is a *Nextflow* script, comprises internal script processes written in Bash or Python. The overall pipeline is shown in figure 7.

18 **Allignment.** All alignments in the tool were conducted using the *STAR* aligner [9], optimized for RNA-seq data. The original tool utilized *BWA* [8], which is less suitable for RNA-seq as it does not detect and handle splicing.

19 **Transformation** To mask the editing sites, an alignment of the transformed reads to the transformed genome is performed. The tool executes its pipeline 12 times, corresponding to each of the possible single-nucleotide base combinations. Each base combination consists of a reference base (*ref-base*) and an alternative base (*alt-base*). The *Genome Transformation* part transform every *alt-base* to *ref-base* using *AWK* [19], followed by indexing the transformed genomes using *STAR* aligner. The transformed indexes are than transferred to the second alignment in *Pre-Analysis* (the alignment of the transformed reads to the transformed genome).

20 ***Pre-Analysis*** This section executes the principal steps of the original pipeline as described in the *Computational Background* section: aligning the initial reads, isolating unmapped reads, transforming reference bases to alternative bases in both the reads and the reference genome and realigning the modified reads to identify the genomic positions of potential hyper edited clusters. In contrast to the original tool, we first perform quality filtering using *fastp* [20], focusing on parameters that control quality score, N values, and sequence complexity (to avoid same nucleotide sub sequences). Full details of the *fastp* parameters are provided in the supplementary materials.

The quality-filtered reads are then aligned to the original genome using *STAR*, with the human reference genome (hg38) used in our analysis.

Assuming that hyper-edited reads do not successfully align, we extract only the unmapped reads from the initial alignment, using STAR's *–outReadsUnmapped* option. The initial alignment is more stringent, allowing up to 5 multiple alignments per read (reads mapped to more than 5 loci are considered unmapped) to ensure ample data for the subsequent steps. These unmapped reads are transformed using *AWK* and aligned to the corresponding transformed genome (using the same base combination) as provided by *Genome Transformation*.The second alignment is more flexible, permitting up to 20 loci to capture all relevant data due to the reduced complexity of the 3-base transformed alignment. Finally, using the original FASTQ files and utilizing the PySAM module [21] in a Python script, the original (four-letter) sequences of the secondarily mapped reads (after transformation) are recovered and replaced in the sequence field (SEQ) of the SAM [22] output files.

Detection. The input for this component consists of BAM (Binary SAM) files generated in *Pre-Analysis*, containing the original read sequences along with their correct genomic alignments from the second alignment step. We begin by analyzing these files to detect editing events by examining all mismatch occurrences, comparing the read sequences to their corresponding aligned sequences in the reference genome. For each mismatch, we determine whether it represents an editing site—where the original base in the genome matches the *ref-base* and the altered base in the read matches the *alt-base*—or another type of mismatch, and record the count of each type of event. Additionally, each read is annotated with supplementary metadata, including the alignment CIGAR string, SAM flags, and the genomic and read positions of the splicing "blocks" (the internal partition of thre read). All this information is then saved in a CSV file.

Afterwards, we perform a filtering process to extract reads that are potentially hyper-edited. Initially, we filter both editing sites (ES) and other mismatch events (MM) based on their Phred scores [23], which indicate the quality of the sequencing. Only events with scores above a specified threshold are considered in the subsequent filtering process. This process applies several criteria derived from the original tool, such as the number of ES in a read, the editing fraction (number of ES divided by the total number of mismatches), the ratio of ES to the read length, and the ratio of cluster length to read length. For example, if the editing fraction is 1, all mismatch events are considered ES. These criteria help refine the selection of reads that contain hyper-editing events. The filtering process produces five output files:

1. *Condition Analysis CSV:* This file provides a True/False assessment for each condition (passed/not passed) applied to each read.
2. *Passed Reads CSV:* Contains all the metadata provided in the initial detection CSV file, along with additional information post-filtering, such as the number of passed editing sites (ES) and mismatches (MM).
3. *Motifs CSV:* For each read, this file includes the count of each upstream base present in the total editing events (e.g., 3 of the events had T before them) and similarly for downstream

bases. This information is used for later analysis of motifs related to the editing.

4. *Clusters BED:* Contains the genomic positions of each hyper-editing cluster.
5. *Summary JSON:* Provides statistics and summary information for the entire sample, such as the total number of passed reads and the average number of editing sites (ES).

These output files collectively form the final output of the tool, providing comprehensive information about the editing events that can be further analyzed by the user.

Multi-Mappers. In the original tool, reads that mapped to multiple locations were treated by selecting the location with the highest editing fraction (ratio of ES to total MM), provided that this fraction was at least 10% higher than at all other locations. If this condition wasn't met, or if the read mapped to more than 50 locations, the read was discarded. In the new tool, we allow the second alignment to map to a maximum of 20 locations. When multiple locations are detected, the location with the highest editing fraction is selected, without any additional constraints. It is important to note that this fraction is based on the ES and MM before filtering (meaning even low-quality score events can influence the fraction), in contrast to the editing fraction provided in the filtering output.

Paired-End Data. In the original tool, paired-end (PE) reads were treated as two separate single-end reads throughout the pipeline. At the end of the process, a requirement was imposed that the mate read be mapped to a nearby region (within 500 kbp) and in the opposite orientation. In the new tool, paired-end data is treated as such throughout the entire *Pre-Analysis* part, including during *fastp* processing, the first *STAR* alignment, and the second *STAR* alignment. The consequence for the alignments is that if one mate is considered unmapped, the other mate will also be identified as unmapped, thereby preserving the PE properties of the data and utilizing *STAR*'s internal quality mapping criteria for PE data. In the final part of the pipeline (*Detection*), each mate is then treated independently for the detection of hyper-editing events. If a read passes all conditions, the tool checks whether its mate has a complementary editing event (e.g., mate 1 has an A-to-G conversion, and mate 2 has a T-to-C conversion). This information is then added to the passed reads CSV output file.

Stranded Data. In RNA-seq experiments, stranded data refers to sequencing where the strand of origin for each read is known, which is crucial for accurately interpreting sequence modifications. For datasets where the sequenced strand is random, the interpretation of A-to-G editing sites can be ambiguous, as these may also be represented as T-to-C changes on the opposite strand. The original tool accounted for this by verifying whether the editing site corresponded to the strand information, ensuring correct identification of A-to-G or T-to-C changes. However, due to time constraints and prioritization of other significant features, we did not implement support for stranded data in our new tool. Instead, we focused on core functionalities and improvements that had a more immediate impact on finding hyper-edited reads. This feature can be easily integrated in the future, thanks to the modularity and straightforward pipeline structure provided by the tool's Nextflow implementation.

Splice Junctions. A future improvement for the tool should address the handling of

splicing junctions identified by STAR during the alignment process. Since the second alignment is performed on transformed reads and genome sequences, splice junction motifs may be altered, potentially leading to the omission of true splicing junctions or the introduction of false ones. To mitigate this issue, a promising approach would be to utilize STAR's capability to output splice-junction VCF files from the initial alignment. These VCF files could then be transformed in the same manner as the reads and genome sequences, ensuring that the transformed splice junctions are accurately represented in the subsequent alignment. Implementing this strategy would help maintain alignment accuracy and reduce the likelihood of errors related to splicing junction changes, ultimately improving the tool's performance and reliability.

Parameter Adjustments. Most parameters utilized in the new tool are derived from those described in the original article, standard lab practices, or default settings. The entire pipeline's parameters can be customized through configuration files or command-line flags, providing flexibility and modularity. Detailed information about all available parameters is provided in the Supplementary Materials section.

***Docker* Containers.** All internal processes that depend on external programs, beyond basic Bash commands, are executed within *Docker* containers as specified in the configuration files. This approach ensures a consistent and isolated environment for the execution of these processes, minimizing potential conflicts between software dependencies. A detailed description of the chosen *Docker* containers, is on the *Hyper-Editing* manual presented in the *Supplementary Materials* section.

VIII Discussion

RNA editing is a well-studied post-transcriptional modification process with significant biological implications, particularly in gene expression regulation, protein diversity, and immune response modulation. Despite decades of research, many aspects of RNA editing remain unknown, especially its full range of functional impacts across different tissues and species. Uncovering more about these editing events is crucial, as they likely play roles in numerous biological processes and disease mechanisms that are not yet fully understood. Therefore, the development of advanced tools for detecting and analyzing RNA editing is essential for pushing the boundaries of this research field and uncovering novel applications, particularly in disease treatment and cellular biology.

The newly built RNA Hyper-Editing Detection Tool addresses several key limitations of the original tool developed in 2014 by incorporating modern technologies such as the *STAR* aligner, *Nextflow*, and *Docker*. This redesigned system is specifically tailored to handle the unique challenges of RNA-seq data, providing a significant leap forward in accuracy, efficiency, and modularity. *STAR*'s support for spliced alignments and its enhanced memory-sharing capabilities significantly improve detection accuracy compared to *BWA*. Additionally, *Docker* ensures seamless deployment across different computing environments, making the tool more accessible to researchers worldwide.

Moreover, the integration of *Nextflow* for workflow management and parallel processing allows

the tool to handle large datasets more efficiently, resulting in faster processing times compared to the original version. This improvement is particularly crucial for large-scale projects involving RNA-seq data from multiple tissues, as demonstrated by the Genotype-Tissue Expression (GTEx) analysis presented in this study. The tool's modularity, divided into distinct components, further enhances its adaptability for various research settings, allowing users to customize parts of the pipeline or use it with different types of sequencing data.

Preliminary results from the tool show promising outcomes, including tissue-specific editing levels, the detection of editing in different organisms, and a significant reduction in run time. Another key finding is the decrease in the number of hyper-edited clusters identified by the new tool. This reduction can be attributed to several factors, including more stringent quality filtering using fastp, the improved alignment provided by *STAR*, and the handling of the data as paired-end (PE) reads. Although fewer clusters are identified, they may have higher biological relevance, as the tool more effectively excludes low-quality reads and misalignments. However, a comprehensive analysis is still required to fully evaluate the capabilities of the new tool. This should include an in-depth comparison of the results produced by the new tool and the original tool on the same datasets, helping to identify key differences in performance, sensitivity, and accuracy in detecting hyper-editing events.

Despite the main improvements, several limitations remain. The tool currently lacks support for stranded RNA-seq data, which is essential for accurately interpreting A-to-I editing events in these datasets. Additionally, future work should focus on enhancing the handling of splicing junctions during the transformed alignment step to avoid omitting true splicing junctions or introducing false ones. Addressing these issues would further improve the tool's accuracy and expand its applicability across a broader range of RNA-seq experiments.

In conclusion, the RNA Hyper-Editing Detection Tool marks an important step forward in RNA editing analysis, addressing limitations of the previous method. By incorporating newer technologies like *STAR*, *Nextflow*, and *Docker*, the tool aims to improve accuracy and efficiency while offering flexibility for various RNA-seq datasets. Preliminary results are encouraging and suggest potential for deeper insights into RNA editing across different tissues and species.

Future improvements, such as support for stranded RNA-seq data and better handling of splicing junctions, will be important for enhancing the tool's applicability. We hope that this tool will serve as a valuable resource for researchers exploring the complexities of RNA editing.

IX Acknowledgements

I would like to deeply thank Roni Cohen-Fultheim for her support and assistance throughout this process. I also appreciate Erez Levanon for leading the lab and providing guidance on key issues encountered along the way. Thanks to Itamar Twersky for help with technical problems related to Nextflow, and to Hagit Porath for insights on the original tool. I'm grateful to Kobi Shapira, Eli Sinai, and Daniel Hevdeli for their initial use of the tool and their valuable suggestions

for optimization.

References

[1] Hagit T. Porath, Shai Carmi, and Erez Y. Levanon. "A genome-wide map of hyper-edited RNA reveals numerous new sites". In: Nature Communications (2014). URL: https://doi.org/10.1038/ncomms5726.

[2] S. Sadeq et al. "Endogenous Double-Stranded RNA". In: Noncoding RNA (Feb. 2021). DOI: 10.3390/ncrna7010015.

[3] Hiroshi Kato et al. "Length-dependent recognition of double-stranded ribonucleic acids by retinoic acid-inducible gene-I and melanoma differentiation-associated gene 5". In: Journal of Experimental Medicine (July 2008). DOI: 10.1084/jem.20080091.

[4] S. Reikine, J. B. Nguyen, and Y. Modis. "Pattern Recognition and Signaling Mechanisms of RIG-I and MDA5". In: Frontiers in Immunology (July 2014). DOI: 10.3389/fimmu.2014.00342.

[5] B. J. Liddicoat et al. "RNA editing by ADAR1 prevents MDA5 sensing of endogenous dsRNA as nonself". In: Science 349.6252 (2015), pp. 1115–1120. DOI: 10.1126/science.aac7049.

[6] Brenda L. Bass. "RNA Editing by Adenosine Deaminases That Act on RNA". In: Annual Review of Biochemistry (2002). DOI: 10.1146/annurev.biochem.71.110601.135501.

[7] Erez Y. Levanon et al. "Systematic identification of abundant A-to-I editing sites in the human transcriptome". In: Nature Biotechnology (2004). URL: https://doi.org/10.1038/nbt996.

[8] Heng Li and Richard Durbin. "Fast and accurate short read alignment with Burrows-Wheeler transform". In: Bioinformatics (2009). DOI: 10.1093/bioinformatics/btp324. URL: https://doi.org/10.1093/bioinformatics/btp324.

[9] Alexander Dobin et al. "STAR: ultrafast universal RNA-seq aligner". In: Bioinformatics (2013). URL: https://doi.org/10.1093/bioinformatics/bts635.

[10] P.J. McCown et al. "Naturally occurring modified ribonucleosides". In: Wiley Interdisciplinary Reviews: RNA (2020).

[11] C.C. Yang et al. "ADAR1-mediated 3 UTR editing and expression control of antiapoptosis genes fine-tunes cellular apoptosis response". In: Cell Death Dis (2017).

[12] D.E. Dupuis and S. Maas. MiRNA Editing. Methods Mol Biol, 2010.

[13] H. Shen et al. "ADARs act as potent regulators of circular transcriptome in cancer". In: Nat Commun (2022). DOI: 10.1038/s41467-022-29138-2.

[14] Nina Schneider et al. "A pipeline for identifying guide RNA sequences that promote RNA editing of nonsense mutations that cause inherited retinal diseases". In: Molecular Therapy - Nucleic Acids (2024). DOI: 10.1016/j.omtn.2024.102130. URL: https://doi.org/10.1016/j.omtn.2024.102130.

[15] Udi Ehud Knebel et al. "Disrupted RNA editing in beta cells mimics early stage type 1 diabetes". In: Cell Metabolism (2024). DOI: 10.1016/j.cmet.2023.11.011.

[16] Elin Lundin, Chenglin Wu, Axel Widmark, et al. "Spatiotemporal mapping of RNA editing in the developing mouse brain using in situ sequencing reveals regional and cell-type-specific regulation". In: BMC Biology (2020). DOI: 10.1186/s12915-019-0736-3. URL: https://doi.org/10.1186/s12915-019-0736-3.

[17] H.T. Porath, B.A. Knisbacher, E. Eisenberg, et al. "Massive A-to-I RNA editing is common across the Metazoa and correlates with dsRNA abundance". In: Genome Biology (2017). DOI: 10.1186/s13059-017-1315-y. URL: https://doi.org/10.1186/s13059-017-1315-y.

[18] Paolo Di Tommaso et al. "Nextflow enables reproducible computational workflows". In: Nature Biotechnology (2017). URL: https://doi.org/10.1038/nbt.3820.

[19] Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger. The AWK Programming Language. Addison-Wesley Pub. Co., 1988. URL: https://pdf.yt/d/MgN0H1joIoDVoIC7.

[20] Shifu Chen et al. "fastp: an ultra-fast all-in-one FASTQ preprocessor". In: Bioinformatics (2018), pp. i884–i890. URL: https://doi.org/10.1093/bioinformatics/bty560.

[21] PySAM Documentation. URL: https://pysam.readthedocs.io/en/latest/index.html.

[22] Heng Li et al. "The Sequence Alignment/Map format and SAMtools". In: Bioinformatics (2009). DOI: 10.1093/bioinformatics/btp352.

[23] Brent Ewing et al. "Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy Assessment". In: Genome Research (1998). URL: https://genome.cshlp.org/content/8/3/175.

X Supplementary Materials

Hyper-Editing Tool Manual provided in the next page.