UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

CENTRO DE CIENCIAS GENÓMICAS
INSTITUTO DE BIOTECNOLOGÍA

NEXT-GENERATION SEQUENCING ANALYSIS FOR NASCENT RNA DATA BASED ON METABOLIC LABELING AND NUCLEOTIDE CONVERSIONS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADA EN CIENCIAS GENÓMICAS

PRESENTA:

PAULINA ROSALES BECERRA

TUTORES:

PROF. DR. ROBERT SCHNEIDER
DR. KEVIN BROCKERS

Cuernavaca, Mor., México
12 de septiembre, 2023

Dedicada a mi familia que,
de lejos pero siempre cerquita, me acompañó.

Abstract

Dynamic gene expression is fundamental for cellular processes. Next-generation sequencing (NGS) based RNA sequencing (RNA-seq) technologies have revolutionized our ability to explore RNA dynamics. Facilitating the direct observation of newly synthesized RNA molecules, nascent RNA-seq offers a unique avenue to dissect transcriptional activity. Among the nascent RNA-seq methods are SLAM-seq and TimeLapse-seq, both harnessing metabolic labeling and nucleotide conversions to simultaneously detect nascent and steady-state RNA within one experiment. However, existing pipelines developed for the processing of nucleotide conversion datasets like SLAM-DUNK are limited to specific RNA-seq library protocols, and sequencing modes, therefore limiting its utility. To address this limitation, I present a reproducible and adaptable analysis pipeline based on the Snakemake framework, designed to accommodate diverse RNA-seq libraries. The pipeline integrates all essential SLAM-DUNK steps and surpasses its capabilities by (i) being applicable to data generated by for diverse protocols, (ii) generating informative quality controls, and (iii) aiding downstream analyses. Using public and in-house datasets for validation, my pipeline proved to accurately quantify nucleotide conversions, revealing comparable results to prior studies. In addition, the pipeline was used to systematically compare SLAM-seq and TimeLapse-seq methodologies, providing a direct comparison of conversion rates generated from each method. This pipeline's adaptability, reproducibility, and informative outputs contribute to a deeper understanding of gene regulation mechanisms through nascent RNA analysis.

Contents

Chapter 1

Introduction

1.1 Gene expression dynamics and RNA sequencing

Gene expression is the biological process by which the information encoded in a gene is transcribed into RNAs, playing a fundamental role in determining an organism's traits and response to its environment (Alberts, 2017). In the case of protein coding genes, the transcribed RNAs are messenger RNAs (mRNAs), a type of single-stranded RNAs directly involved in protein synthesis by encoding protein information and transporting it from the nucleus to the cytoplasm for its translation. In addition there are also non-coding RNA (ncRNA) genes that encode for different species of RNAs such as rRNAs, tRNAs, snoRNAs, lincRNAs; these genes are transcribed into functional RNA molecules rather than encoding for proteins like mRNAs, participating as structural, catalytic or regulatory RNAs (Eddy, 2001; He & Hannon, 2004). These different RNA species originate from distinct polymerases; for instance, RNA polymerase I (Pol I) is responsible for synthesizing ribosomal RNAs (rRNAs) and Pol III synthesizes transfer RNAs (tRNAs) alongside other small structural RNA species. On the other hand, RNA polymerase II (Pol II) governs the production of various RNA types, including protein-coding mRNAs, long non-coding RNAs (lncRNAs), primary microRNAs (pri-miRNAs), and enhancer RNAs (eRNAs; Sims et al., 2004; Hsin & Manley, 2012; Jonkers & Lis, 2015; Wissink et al., 2019).

Within gene expression, dynamic mechanisms govern how genes are selectively activated or silenced within each cell, leading to the unique molecular composition and functional characteristics that distinguish one cell type from another even when they share the same genomic material. Underlying this complex process, transcription and thus RNA abundance within a cell, is highly regulated in response to changing environmental conditions and cellular cues (e.g. temperature, stress, chemical signals, etc.), ensuring precise timing and levels of gene expression (Levine & Tjian, 2003; Davidson, 2010).

RNA sequencing (RNA-seq) is a powerful next-generation sequencing (NSG) technique that enables the comprehensive analysis of gene expression on a genome-wide scale (Stark et al., 2019). By sequencing and quantifying the abundance of RNA molecules within cells, RNA-seq can be used as a method to uncover the dynamic landscape of gene expression under different biological conditions. Total RNA-seq entails the sequencing of the entire RNA transcriptome, including both coding and non-coding RNAs (Wang et al., 2009a). In contrast, mRNA-seq selectively enriches polyadenylated (poly(A)) RNA therefore, capturing only mRNAs, providing a snapshot of protein-coding transcripts and their expression levels (Mortazavi et al., 2008; Wang et al., 2009). Both total RNA-seq and mRNA-seq provide insights into the steady-state levels of RNA molecules reporting the accumulated RNA content within the cells at a given moment. However, it's important to recognize that RNA regulation extends beyond transcription, encompassing various other processes including RNA processing and decay (Dölken et al., 2008; Rabani et al., 2011; Alpert et al., 2017).

1.2 Nascent RNA sequencing for ongoing transcriptional activity analysis

Nascent RNA-based methods have emerged as a promising alternative for identifying alterations in gene expression, along with intricate RNA dynamics encompassing processing and degradation. By specifically detecting newly synthesized RNA within the total RNA pool, these sequencing-based methodologies offer a more immediate and direct insight into the products of ongoing transcriptional activity within cells (Wang et al., 2009; Wissink et al., 2019). The two most used approaches to distinguish between steady-state and nascent RNA are immunoprecipitation (IP)-based methods (Fig. 1a-b) and sequence composition methods (Fig. 1c; Wissink et al., 2019).

While both approaches aim to uncover gene expression changes, they diverge in their strategies and outcomes (Table 1). First, IP-based techniques employ RNA immunoprecipitation with antibody-mediated enrichment of a protein, followed by the isolation, reverse transcription, and sequencing of its interacting RNAs (Wissink et al., 2019). These techniques include metabolic labeling protocols, that rely on the immunoprecipitation of newly synthesized RNA with incorporated nucleotide analogues such as 5-bromouridine 5'-triphosphate (BrU) for Global run-on sequencing (GRO-seq; Fig. 1a; Core et al., 2008) or 4-thiouridine (4sU) for transient transcriptome sequencing (TT-seq; Schwalb et al., 2016). Alternatively, other IP methods targeting Pol II-associated transcripts, like mammalian native elongating transcript sequencing (mNET-seq; Nojima et al., 2015), have also been employed for nascent RNA analysis and do not rely on the incorporation of a nucleotide analogue (Fig. 1b). These techniques exclusively capture newly synthesized RNA, yielding insights into the immediate products of transcription while omitting information about total RNA levels. However, while effective for nascent RNA analysis, these methods often involve more intricate protocols, extensive RNA handling, and a high amount of starting material (Stark et al., 2019; Wissink et al., 2019).

Method	**Strategy**	**Outcome**
GRO-seq	BrU labeling and IP	Nascent RNA
TT-seq	4sU labeling and IP	Nascent RNA
mNET-seq	Pol II IP	Nascent RNA
SLAM-seq and TimeLapse-seq	4sU labeling and sequence composition	Steady-state and nascent RNA

Table 1: Characteristics of nascent RNA analysis methods. GRO-seq, global run-on sequencing; TT-seq, transient transcriptome sequencing; mNET-seq, mammalian native elongating transcript sequencing; SLAM-seq, thiol(SH)-linked alkylation for the metabolic sequencing of RNA.

1.3 SLAM-seq and TimeLapse-seq metabolic labeling methods for nascent RNA quantification

In contrast to IP-based methods, sequence composition methods rely on the nucleotide content of the transcripts after sequencing for the identification of nascent RNAs within the total RNA pool. The general experimental strategy involves the incubation of cells in a medium supplemented with cell-permeable nucleotide analogs (i.e. 4sU), these nucleotide analogs are then incorporated

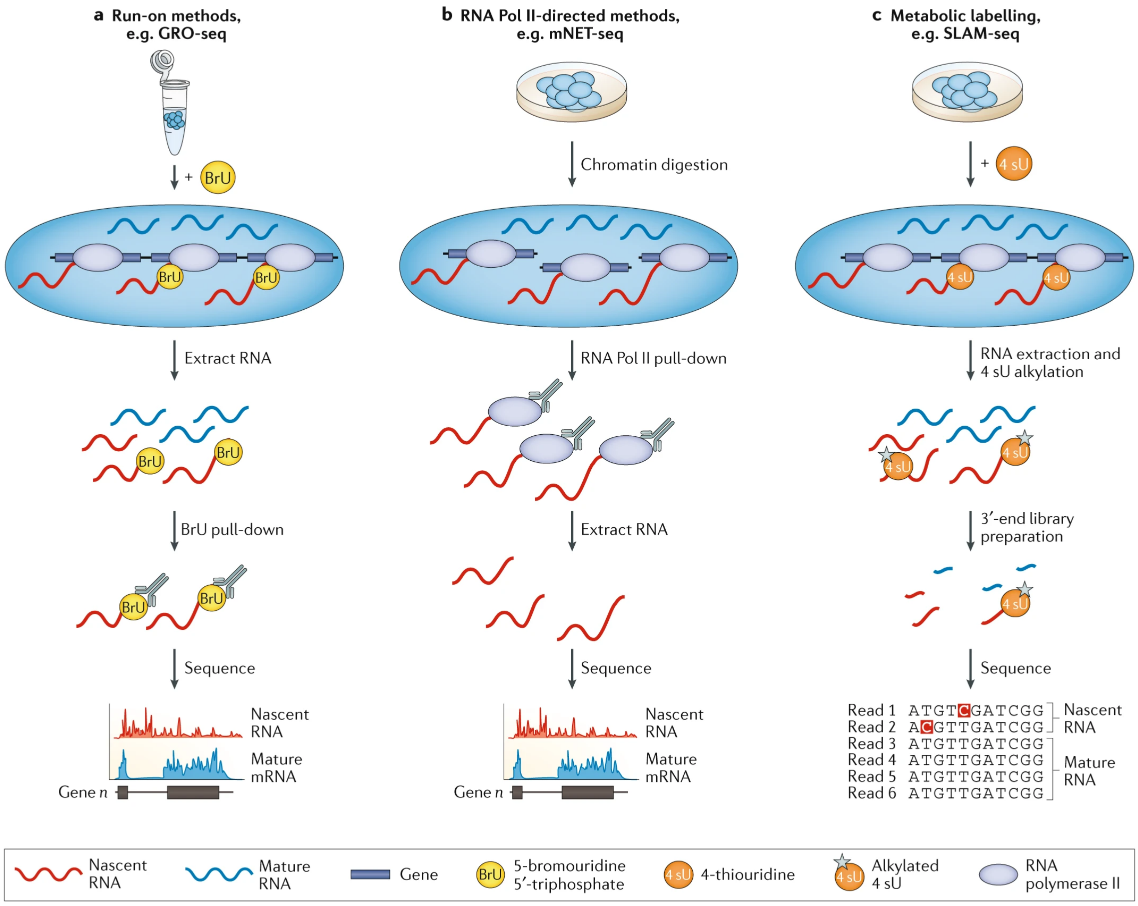

Figure 1: Nascent RNA analysis methods. Nascent RNA analysis methods enrich newly transcribed RNAs fro this to an unenriched (steady-state RNA) control, by different methods for either immunoprecipitation (**a** and **b**) or sequence composition analysis (**c**). (**a**) Run-on methods label RNA by adding a time-limited pulse of modified ribonucleotides into cell media; various modified nucleotides can be used, but global run-on sequencing (GRO-seq) and its corresponding 5-bromouridine 5'-triphosphate (BrU) nucleotide analogue are shown. After incorporation of the modified bases, nascent-RNA strands are enriched by immunoprecipitation (IP) with antibodies specific to the modified nucleotide used and are prepared for RNA-sequencing (RNA-seq) analysis. (**b**) RNA polymerase II (Pol II) IP methods pull down Pol II-associated RNAs after chromatin digestion with micrococcal nuclease. During chromatin digestion, the nascent RNA is protected from nuclease activity by its Pol II footprint. The protected RNA is extracted and processed for RNA-seq analysis. (**c**) Sequence composition methods label RNA similarly to run-on methods, but they use the nucleotide analogue 4-thiouridine (4sU). For the shown SLAM-seq method, alkylation of 4sU after RNA extraction prompts misincorporation of G nucleotides during reverse transcription, allowing 4sU incorporation sites to be directly determined by mutational analysis with base-pair resolution. Preparation of a 3 '-end RNA-seq library increases the signal by reducing the amount of unlabelled RNA carried through to sequencing. Figure taken from Stark et al., 2019.

into newly synthesized RNA during transcription an serve as a marker for distinguishing nascent RNA from steady-state RNA via identification of chemically induced nucleotide-conversions during sequencing data analysis (Fig. 1c; Wissink et al., 2019). Based on this principle, two different methods have been developed: SLAM-seq (thiol(SH)-linked alkylation for the metabolic sequencing of RNA; Herzog et al., 2017) and TimeLapse-seq (Schofield et al., 2018).

By sequencing the total RNA pool, SLAM-seq and TimeLapse-seq offer an alternative that eliminates the need for RNA enrichment and complex experimental procedures. Additionally, by comparing nascent RNA (4sU labeled) with stable RNA (unlabeled) within the same sample, single-nucleotide conversion approaches minimize potential sources of variation and additionally provide information about total RNA levels.

Conceptually similar, both approaches work by detecting the incorporated 4sU as single-nucleotide thymine-to-cytosine mutations (T > C) through reverse transcription and subsequent sequencing (Fig. 2), providing a binary and quantifiable marker to distinguish between nascent and steady-state RNA during data analysis. Notably, SLAM-seq chemistry uses alkylation of the 4sU thione to induce mutations and the uses 3'-end mRNA sequencing (QuantSeq; Saunders et al., 2006), whereas TimeLapse-seq uses an oxidative nucleophilic aromatic substitution reaction for cytidine pattern matching and uses total RNA-seq.

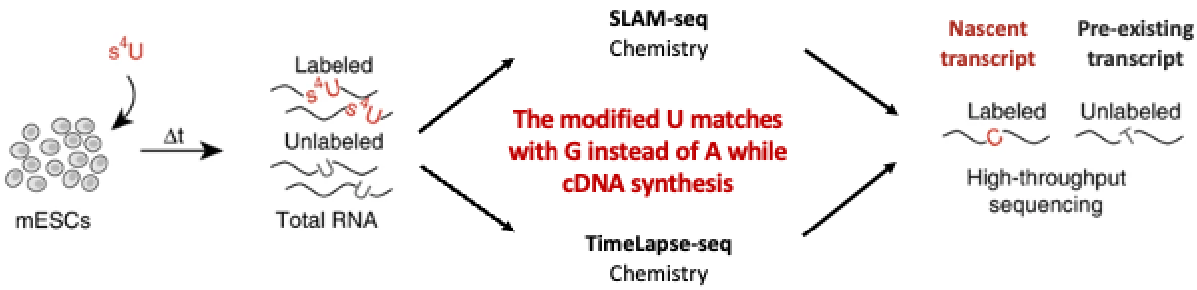

Figure 2: 4sU metabolic labeling for nascent RNA detection methods. Workflow of 4-thiouridine (4sU) metabolic labeling methods SLAM-seq and TimeLapse-seq, differing on the employed chemistry for 4sU incorporation into newly transcribed RNA but both leading to thymine-to-cytosine conversion detection after sequencing. SLAM-seq chemistry involves alkylation of 4sU after RNA extraction, prompting misincorporation of G nucleotides during reverse transcription, allowing 4sU incorporation sites to be directly determined by mutational analysis with base-pair resolution (Herzog et al., 2017). TimeLapse-seq instead uses an oxidative nucleophilic aromatic substitution reaction to fully recode the hydrogen bonding pattern of 4sU to match the native pattern of cytidine (Schofield et al., 2018).

1.4 Data analysis implications for nucleotide-conversion datasets

Relying directly on the data analysis to identify and quantify nucleotides with T > C conversions, SLAM-seq and TimeLapse-seq require a custom nucleotide-conversion centered analysis. Developed as a complementary analysis tool for SLAM-seq data, Digital Unmasking of Nucleotide conversions in K-mers (SLAM-DUNK; Neumann et al., 2019), is an data analysis pipeline that enables the quantification of nucleotide conversions in high-throughput RNA-seq datasets.

The usage of this pipeline offers a range of significant advantages. Firstly, it accounts for technicalities regarding reads containing several single-nucleotide mutations, such as: genomic sequence content, conversion-aware read mapping, exclusion of false-positive T > C conversions (from sequencing error and experimentally-induced) and Single Nucleotide Polymorphism (SNP) correction for accurate quantification. Secondly, the DUNK pipeline is divided into four sequential modules (map, filter, snp and count) allowing custom configuration of parameters and resources used for a given step. More importantly, SLAM-DUNK provides a comprehensive analysis regarding both unconverted and nucleotide-conversion-containing reads, reporting relevant estimations related to read coverage, base-content and nucleotide conversion rates for nascent and stable RNA. Overall, providing a comprehensive approach that covers the analysis of both steady-state and nascent RNA contexts.

However, built around the SLAM-seq original method, DUNK analysis is based on QuantSeq data as input. Designed specifically for the analysis of data from a 3'-end mRNA sequencing protocol, SLAM-DUNK operates under certain technical assumptions about the data characteristics: sequencing in a single-end format, the presence of one read per transcript, uniform length of targeted 3' untranslated regions (UTRs) used and high multimapping rates due to the use of low sequence complexity in 3' UTRs. While these considerations are accurately addressed throughout the pipeline and it has been suggested that 3' RNA-seq can yield similar results as whole transcript sequencing (Tandonnet & Torres, 2017; Ma et al., 2019), it is important that experimental strategies are designed around the objective and resource availability. Consequently, the incompatibility of the original SLAM-DUNK analysis with other RNA-seq data strategies represents a significant limitation, restricting its applicability and utility. Highlighting the necessity of the development of an SLAM-DUNK-like pipeline that can produce comparable results regarding stable and nascent RNA dynamics while remaining compatible with a range of standard RNA-seq library preparation protocols (e.g. total RNA, mRNA).

1.5 Reproducibility and scalability in omics data analysis pipelines

Two fundamental considerations for the development of robust pipelines for omics data analysis are reproducibility and scalability. Workflow managers play a crucial role in omics data analysis by providing systematic and automated frameworks to perform complex data processing pipelines. These tools simplify the computational process, enhance reproducibility, and optimize resource utilization, enabling researchers to effectively handle and interpret the complex datasets generated. Some of the most used workflow manager systems are Nextflow (Di Tommaso et al., 2017) and Snakemake (Köster & Rahmann, 2012).

Nextflow and Snakemake are both robust workflow management systems designed to streamline and automate complex data analysis pipelines in the omics research field. Nextflow outperforms in cross-infrastructure task execution, offering adaptability, portability and nf-core community support, while Snakemake stands out for its human-readable syntax that simplifies the creation of pipelines (Köster & Rahmann, 2012; Di Tommaso et al., 2017; Ewels et al., 2020; Jackson et al., 2021; Mölder et al., 2021). While Nextflow emphasizes scalability and reproducibility, Snakemake prioritizes both these aspects along with efficiency, making them valuable tools for tackling complex data analysis challenges.

Although a version of the SLAM-DUNK pipeline has been integrated into the Nextflow community (nf-core) framework (https://nf-co.re/slamseq/1.0.0) to enhance reproducibility and facilitate computational resource management for extensive analysis, the pipeline's compatibility is exclusive to 3' RNA-seq strategies. Notably, any desired modifications to the workflow's steps are only possible through input arguments, allowing parameter adjustments within a run, but without the flexibility to alter the source code for modifications of the computational step. Hence, to establish an automated pipeline that conducts SLAM-DUNK analysis while ensuring adaptability across diverse RNA-seq protocols, the design of a new analysis pipeline within a workflow manager was imperative.

Chapter 2

Aim

2.1 General

Data analysis pipelines like SLAM-DUNK have enabled the precise quantification of nucleotide conversions within high-throughput RNA-seq datasets in order to determine transcription rates. These approaches offer valuable insights into the intricate dynamics of stable and nascent RNA molecules. However, the existing limitations of the original SLAM-DUNK analysis, particularly its specificity to certain RNA-seq data strategies, underscore the need for the development of more adaptable methodologies.

Given that no pre-existing pipeline for datasets containing T $>$ C conversions covers challenges regarding automation, reproducibility and adaptability. Here we aim to develop a pipeline based on Snakemake that enables nucleotide-conversion-based RNA analysis for multiple RNA-seq library preparation protocols. This project addresses this challenge by proposing an extended framework that accommodates a wider array of RNA-seq library preparation protocols as well as different metabolic labeling methods (SLAM-seq and TimeLapse-seq), aiming to provide a comprehensive understanding of RNA dynamics across diverse experimental designs. By taking advantage of the power of quantitative nucleotide-conversion-based RNA analysis and expanding its applicability, this study contributes to the generation of data analysis tools that can help answer questions regarding our comprehension of gene expression regulation and cellular processes.

The workflow behind the designed pipeline comprehensively incorporates all four SLAM-DUNK steps, each of them adapted to accommodate different RNA-seq libraries. Additionally, the workflow includes supplementary pre-processing, quality control steps, and extra outputs that go beyond what is offered by nf-co.re/slamseq or the SLAM-DUNK tool. Regarding automation, I opted to utilize Snakemake as workflow manager backbone due to its simplified Python-based syntax, easy setup for high-performance computing (HPC) systems, and overall clarity in defining the structure when compared to Nextflow. Overall, this comprehensive approach results in an extensive and reproducible analysis framework for nascent RNA data, based on metabolic labeling for nucleotide conversions (see Results).

In order to test the performance of the pipeline within a range of standard RNA-seq library preparation protocols, I used four different datasets, two of them publicly available and two generated for this project (Table 2). As described, the 3' UTR SLAM and Total RNA TimeLapse datasets are both from the original method publications. I used these datasets to ensure that the results produced from our pipeline were comparable to the previously shown. Within these public datasets, we selected samples that are either metabolically labeled or not with no additional biological modifications to avoid variance coming from experimental design. Additionally, we wanted to compare SLAM-seq and TimeLapse-seq with data generated in the same model system and with the same library preparation and sequencing protocols.. For this, two mESCs mRNA datasets were generated by another lab member, Huiwen Li (see Materials and Methods) and used for the comparative analysis.

	Dataset	**Labeling protocol**	**Sequencing strategy**	**Cell type**	**Total num. of samples**	**Reference**
1	3' UTR SLAM	SLAM-seq	QuantSeq	Haploid mESC	6	Herzog et al., 2017
2	mRNA SLAM	SLAM-seq	mRNA-seq	Diploid mESC	4	Schneider lab (unpublished)
3	mRNA TimeLapse	TimeLapse-seq	mRNA-seq	Diploid mESC	2	Schneider lab (unpublished)
4	Total RNA TimeLapse	TimeLapse-seq	Total RNA-seq	MEF	8	Schofield et al., 2018

Table 2: Datasets used for pipeline prototyping. Different datasets used for testing of the nucleotide-conversion nascent RNA pipeline. Two public (1 and 4) and two generated (2 and 3) were used to compare differences in results according to categories shown on the table. Total number of samples sums untreated (no 4sU) and treated (+4sU) samples from a given reference, each category representing half of the dataset (e.g. total of 6, represents 3 no 4su and 3 +4sU samples). For more information about data obtainment and generation see Materials and Methods.

2.2 Particular

1. Use Snakemake to generate a reproducible automated workflow structure for nascent RNA sequencing data analysis.

2. Allow use of pair-end data as input for SLAM-DUNK analysis.

3. Adapt DUNK steps to non-3' RNA sequencing approaches.

4. Include standard quality controls into the pipeline for automatic generation.

5. Generate additional outputs that can be used for further analysis outside the pipeline.

6. Compare nucleotide-conversion results among test datasets focusing on RNA-seq library preparation protocols (3' mRNA-seq, mRNA-seq and total RNA) and metabolic labeling methods (SLAM-seq and TimeLapse-seq) differences.

Chapter 3

Results

3.1 Automated pipeline for customizable analysis

The pipeline for identification of T > C conversions in paired-end RNA-seq data reported here, addresses two main challenges: (i) analysis reproducibility and scalability, and (ii) accurate identification of nucleotide-conversions for paired-end RNA-seq data. Regarding the first challenge, the complete data analysis was written as a Snakemake pipeline (Köster & Rahmann, 2012; Mölder et al., 2021), achieving automation and high reproducibility of the analysis without compromising its flexibility. In the Snakemake workflow manager every individual computational step, e.g a shell command or a R script, is denoted as a rule with a defined set of input and output files. Snakemake constructs a directed acyclic rule graph (DAG) to determine file dependencies and an optimal computation order to produce the desired output files and jobs are then submitted accordingly.

Once Snakemake was set as the backbone of the pipeline, we focused on the second challenge of this analysis: nucleotide-conversion identification. For this, I first established a general RNA sequencing analysis workflow with the pertinent considerations for a nucleotide-conversion centered analysis at all steps (Fig. 3). These considerations account for related data analysis adaptations concerning mutation tolerance and correct identification, quantification of both total and labeled reads, additional normalization due to sequencing mode change and calculation of nucleotide-conversion rates.

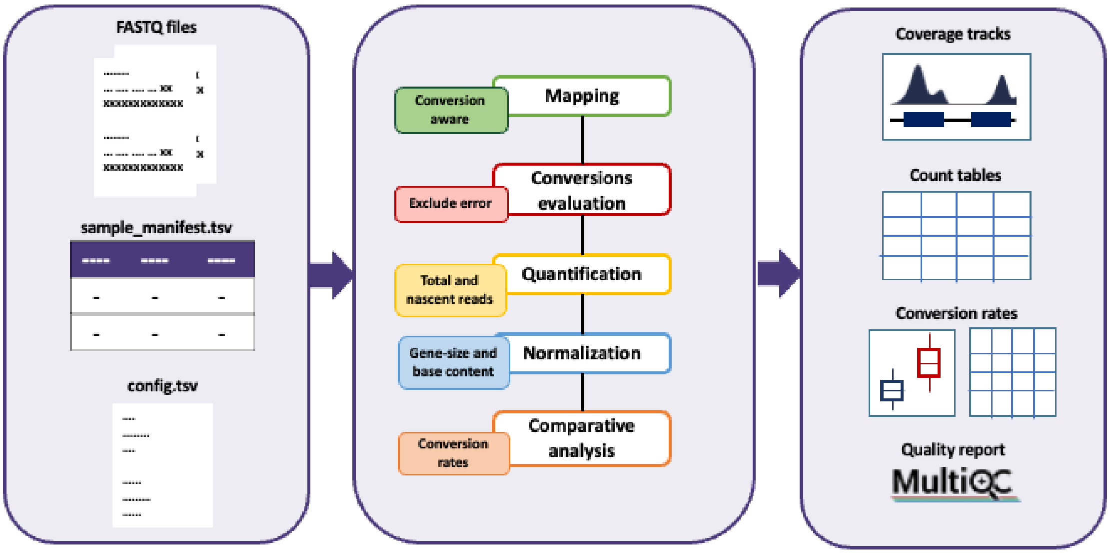

Figure 3: Pipeline for T > C conversions in paired-end RNA-seq data overview. The pipeline is split into input (left), general workflow (middle) and output (right). The workflow shows only the general steps and considerations taken for the development of the pipeline (see text for details).

Once file handling and overall workflow structure were established, I translated the general steps into rules, generating a job-execution system based on output-input interactions, where each rule represents either a command or script executed in order to generate a desired output. This pipeline consists of a total of 25 interconnected rules representing the whole workflow (Supplementary Fig. 1). Along with intermediate processing files, the resulting pipeline generates a set of relevant outputs for nucleotide-conversion centered RNA-seq analysis such as: coverage tracks, count matrices for total and nascent reads, nucleotide conversion rates tables and plots, and quality control reports (Fig. 3).

3.2 Nucleotide-conversion aware mapping for multiple RNA-seq library protocols

Within the mapping step of our pipeline, two additional challenges were taken into consideration: (i) conversion-aware read assignment, and (ii) flexibility regarding the sequencing library preparation used as input. To address the first, NextGenMap (Sedlazeck et al., 2013) was used, setting it to a T > C conversions aware mode implemented by the DUNK pipeline (Neumann et al., 2019). In contrast to a standard scoring, the conversion-aware scoring scheme from DUNK, avoids both mismatch penalty or match score for T > C conversions (see Materials and Methods).

While the map module inside the DUNK pipeline covers the single-nucleotide mutation tolerance needed, the original workflow was built based on a 3'-end mRNA sequencing library preparation (QuantSeq), limiting the data analysis to single-end mode and 3' regions centered. In order to allow the use of different sequencing library preparation protocols, we implemented NextGenMap for paired-end data and tested it for different sequencing methods (Fig. 4a). As shown, conversion-aware mapping was able to identify and assign reads containing T > C nucleotide conversions independently of the sequencing mode and experimental protocol (Fig. 4b).

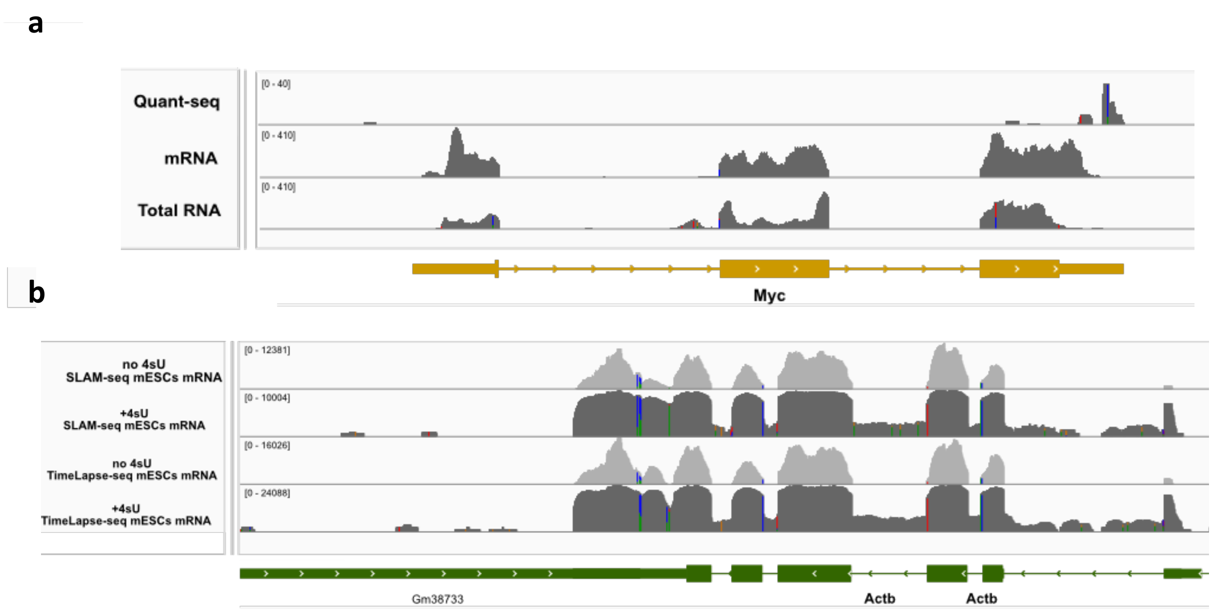

Figure 4: Alignment coverage and T > C reads assignment. Genomic tracks visualized from BAM files. (**a**) Shows difference in coverage for the different library preparation protocols. (**b**) Assignment of T > C reads regardless of the nucleotide changes, as well as difference in point mutations for no 4sU (light gray) and +4sU (dark gray) for SLAM-seq (top) and TimeLapse-seq (bottom). Single nucleotide mismatches from the reference are highlighted in colors over the tracks, mutations to A in green, T in red, G in yellow and C in blue.

3.3 Filtering and multimapper reconciliation

In standard RNA-Seq analysis, after mapping a filtering step is performed evaluating reads by identity, quality, mismatch number and other characteristics depending on the alignment tool used, in our case, this particular characteristic was the assignment of multi-mapping reads. Congruently to the analysis, this step was developed based on -slamdunk filter- (Neumann et al., 2019). In addition to “traditional” alignment filtering, DUNK’s filter includes an annotation-based multimapper reconciliation to reassign reads with multiple location hits. The conservative reassignment strategy behind it, builds around the sequencing method used and a supplied annotation table indicating the mapping regions (see Materials and Methods).

Coming from 3’ UTR regions, sequences used on the original SLAM-DUNK method have a lower sequence complexity, tending to report an increased number of reads mapping equally well to several genomic regions (multimappers). While our approach focuses on RNA-seq protocols from coding regions, implying a higher sequence complexity than 3’ UTRs therefore better unique mapping rates, testing the filtering step of the pipeline on our mRNA datasets showed unexpected multimapper hits within no-coding regions. To address this, we tested two different reference annotation files to filter out non-relevant reads, first, a transcript-level annotation; and second, an exon-level annotation. Exploring coverage tracks for genes that previously showed multimap records on no-coding regions, we noticed that using an exon-level reference allowed us to rescue relevant multimap reads.

3.4 Quantification of T $>$ C conversions and normalization

On top of T $>$ C conversion tolerance during the mapping step, an important concept behind nucleotide-conversion oriented analysis is the accurate identification of false-positives to avoid overestimation. In addition to false-positive T $>$ C conversions coming from standard sequencing error, the chemical treatments used for both SLAM-seq and TimeLapse-seq reported an increase of experimental-induced conversions (Herzog et al., 2017; Schofield et al., 2018). To distinguish between true experimental-induced and false-positive conversions, -slamdunk snp- (Neumann et al., 2019) module was implemented into the pipeline. Briefly, Single Nucleotide Polymorphism (SNP) calling is performed on the filtered mapped reads, and if the fraction of reads carrying an alternative base among all reads exceeds given variant fraction and coverage cut-off, DUNK classifies the SNP position as a true SNP to be masked.

Following the mapping, filtering and SNP calling, the resulting data contains mainly high-quality T > C conversions; these conversions will allow the distinction between total (with and without conversions) and nascent (only with conversions) reads. To be able to quantify both total and nascent reads, we continued using DUNK's quantification method. Similar to the filter module, -slamdunk count- requires an annotation table, but this time indicating counting windows. Deferring to the sequencing-based approach for the definition of the mapping regions, where depending on the genomic feature enriched for a given sequencing strategy, in this step we opted to use a transcription-oriented strategy, using whole transcripts as annotated feature to delimit counting windows, this way all reads mapping to a single transcript independently of the exon are collapsed to the same entry (see Methods).

In addition to the annotation, the count module enclosed in our pipeline requires two user-given thresholds: (i) a minimum quality threshold, and (ii) a minimum number of conversions within a read in order for it to be considered nascent. All together with the filtered reads and masked SNPs, the tool generates a count table per sample containing both total and nascent read counts, along with some other relevant values (e.g. conversion rate, T content, counts per million, etc.; Supplementary Tab. 1) at the count window level that will later be collapsed by gene entry to facilitate further (not included in this pipeline) downstream analysis at gene-level.

After generating this count plots for all of our datasets, we distinguished between total and nascent reads to test the ability of the pipeline to identify nascent reads on previously tested datasets (published data from SLAM-seq and TimeLapse-seq), and to confirm that the metabolic labeling was successful for both protocols in our own datasets. As shown in Figure 5, all datasets independently of the protocol, present a significant difference between untreated and treated samples regarding the nascent counts.

Next, to test if the conversion threshold was enough to account for background error conversions, we first grouped the resulting nascent read counts of the mRNA sequencing datasets by the number of T > C conversions using different thresholds (Fig. 6). Being the most permissive option, employing 2 as threshold resulted in the highest amount of conversion positive reads (n) on untreated (no 4sU) samples (Table 3). While a threshold of 4 yielded a very low number of

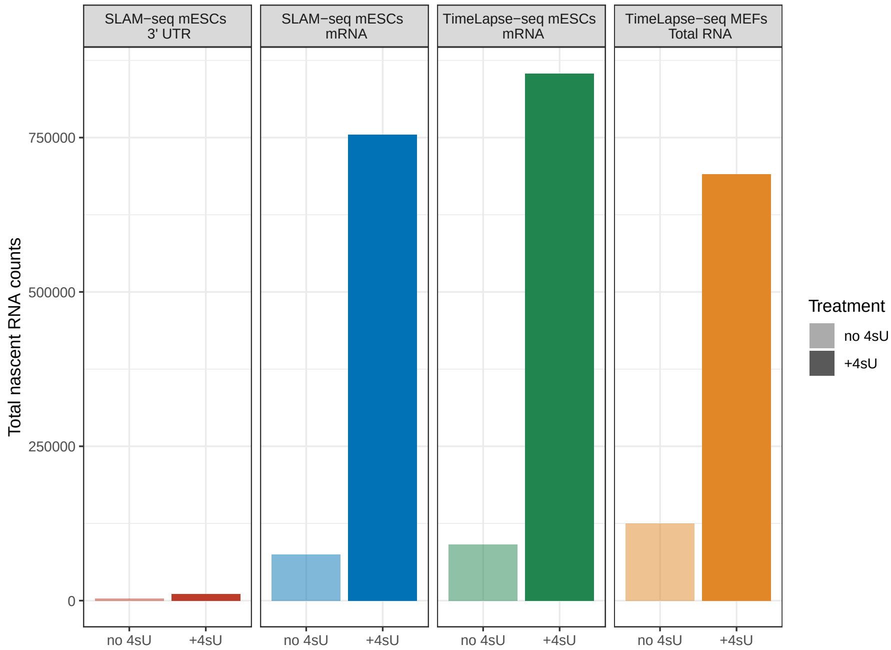

Figure 5: Treatment (+4sU) effect on nascent read counts. Total nascent RNA counts before (no 4sU) and after metabolic labeling (+4sU) for all datasets. Number of reads on the TcReadCount columns for each dataset using a minimum conversion threshold of 2 were summed to get the total number of nascent read counts.

false-positive reads (199 for SLAM-seq and 334 for TimeLapse-seq), we noticed that the loss of 13,000 TC reads compared to a threshold of 2, was too stringent. Additionally, the effect on counts reduction relative to the increase of the threshold showed to be impaired between labeling protocols, SLAM-seq datasets showed a stronger reduction on false-positive T > C reads (T > C reads reported on no 4sU), while maintaining comparable number of T > C reads on +4sU samples (Table 3). Considering this, we opted for a background-subtraction on top of a permissive threshold, avoiding background-error conversions' noise in further analysis while aiming to keep real signals (see Materials and Methods).

After background subtraction, the difference between untreated vs. treated samples is strong enough to identify gene-expression changes for the same genes across different conditions. Nevertheless, there are still two additional variant factors to be taken into consideration before proceeding with further analysis: gene size and transcript abundance. Since our approach is based on genomic features that differ in size and this can later cause biased results, we implemented a normalization step in our pipeline, where a gene-size normalized count entry is added for both total and nascent read counts.

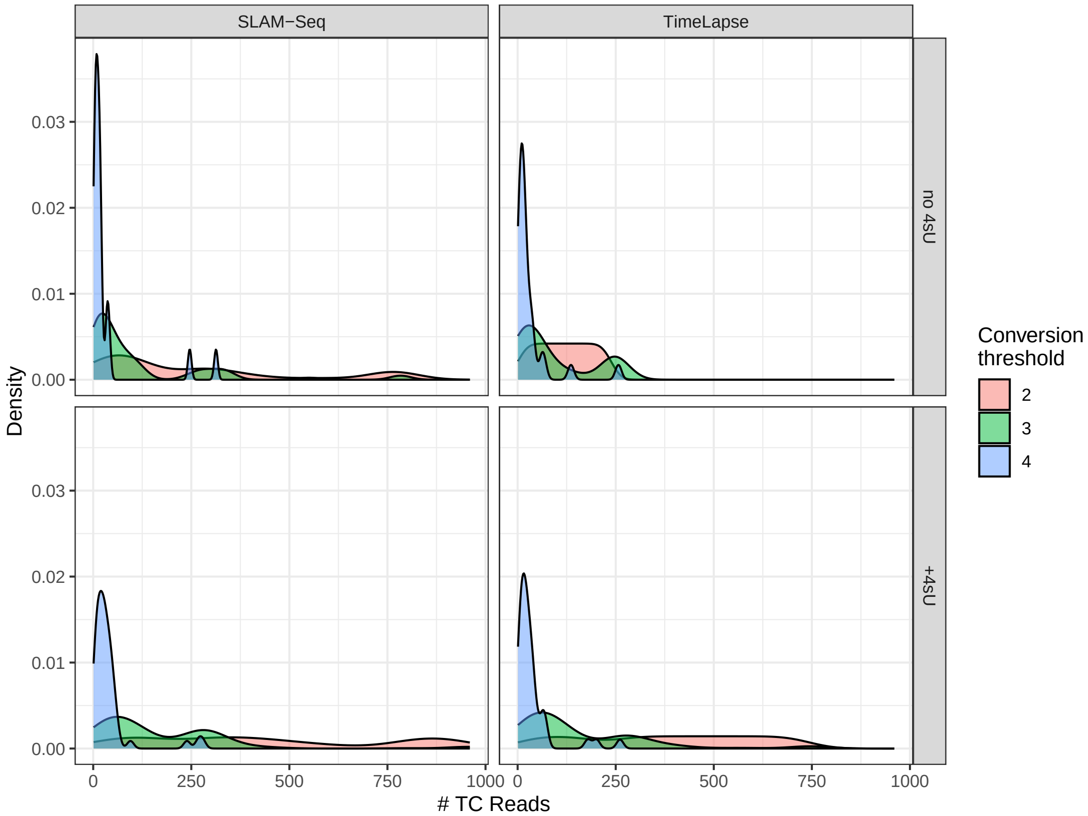

Figure 6: Minimum T > C conversion threshold effect. Density of gene entries (y axis) grouped by number of T > C reads (x axis) reported using different T > C conversion thresholds during the quantification step. Results are divided by metabolic labeling presence: no 4sU (above) or +4sU (below); and method SLAM-seq (left) or TimeLapse-seq (right). Outliers are not shown.

Conversion threshold	**Total T > C reads**			
	SLAM-seq		**TimeLapse-seq**	
	no 4sU	**+4sU**	**no 4sU**	**+4sU**
2	4,045	15,498	4,234	15,996
3	748	9,099	963	8,934
4	199	2,609	334	2,174

Table 3: Total number nascent reads after minimum T > C conversion filtering. Total number of reads identified as nascent after minimum T > C conversion filtering using different thresholds (2,3 and 4), and for SLAM-seq and TimeLapse-seq methods.

Differences on transcript abundances can also be a source of biased results in further analyses. To account for these differences during the same normalization step, we also compute the "fraction of labeled transcript" defined by SLAM-DUNK. This takes not only the ratio of labeled to unlabeled transcripts into consideration, but also if the base composition of those transcripts gives them a higher probability to have conversions, i.e. U-rich transcripts that correspond to T-rich genomic regions. Since a higher chance to present T $>$ C conversions due to nucleotide composition was shown to lead to an overestimation for T-rich and underestimated for T-poor regions (Neumann et al., 2019), the computation of the fraction of labeled transcript is achieved by normalizing to T content and read coverage during the quantification step (see Materials and Methods).

To conclude the quantification steps, we also generate a set of count matrices containing all samples of a given dataset, collapsed by gene entry. As a result different count matrices are generated for each raw and gene-size normalized counts: (i) total read counts, (ii) nascent read counts and (iii) nascent/total read counts. The resulting matrices also contain additional fields useful for downstream analysis (e.g. conversion rates calculation, differential expression analysis, quality controls, etc.); a description of these fields can be found at Supplementary Tab. 2. Altogether with previous steps on the pipeline, quantification outputs are evaluated using -alleyoop tcperreadpos-, -alleyoop snpeval- and standard MultiQC controls, for assessment regarding sequencing reads quality, variant calling and conversion biases, etc.; these quality controls are automatically reported on a QC report generated within the pipeline.

3.5 Comparative analysis

While the count step output calculates fraction of labeled transcript, serving as conversion rates of the transcripts, we also wanted to compare conversion rates at base-level and for different nucleotides. Within our pipeline, -alleyoop utrrates- (Neumann et al., 2019) is used to compute these individual nucleotide conversion rates for all samples. Once again, the original was based on a 3' UTR analysis. In order to use the whole transcript for rate calculation instead of UTRs, we reuse the count windows annotation table as a region defining argument. Computing individual conversion rates per nucleotide combination at the given count window level, is achieved by simply normalizing a given nucleotide conversion over all possible conversions of the reference base and accounting for strandness for correct interpretation of A > G on the minus strand to convert into T > C (see Materials and Methods).

Exploring the conversion rates for our mESCs mRNA datasets we again confirmed that both metabolic labeling techniques were successful, presenting a clear tendency for T > C conversions after treatment, with a slightly stronger effect on SLAM-seq samples (Fig. 7). Moreover, we see similar levels of basal and after treatment rates for non T > C mutations. Comparing our results to the published datasets for SLAM-seq and TimeLapse-seq, we observed different conversion rates between datasets generated using the same labeling protocol (Fig. 8), suggesting a differential detection of conversion rates depending on the sequencing method.

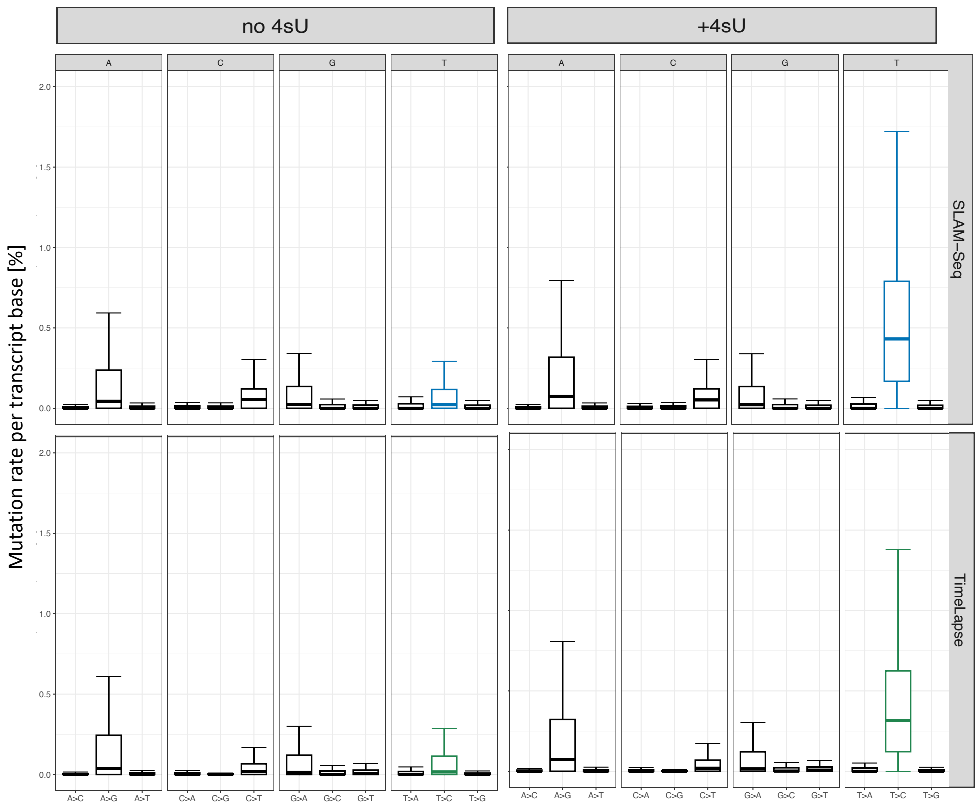

Figure 7: Individual nucleotide conversion rates for mRNA libraries. Conversion rates per base within whole transcript reads of mRNA-seq libraries, prepared from mESCs before (no 4sU, left) and after metabolic labeling (+4sU, right) for SLAM-seq (top) and TimeLapse-seq (below) methods (see Materials and Methods).

Following this, we tested if these differences could be noticeable in a gene expression manner. Using only samples from mESCs with no additional treatment besides metabolic labeling, we examined expression levels for both total reads (steady state) and nascent (T > C reads) counts

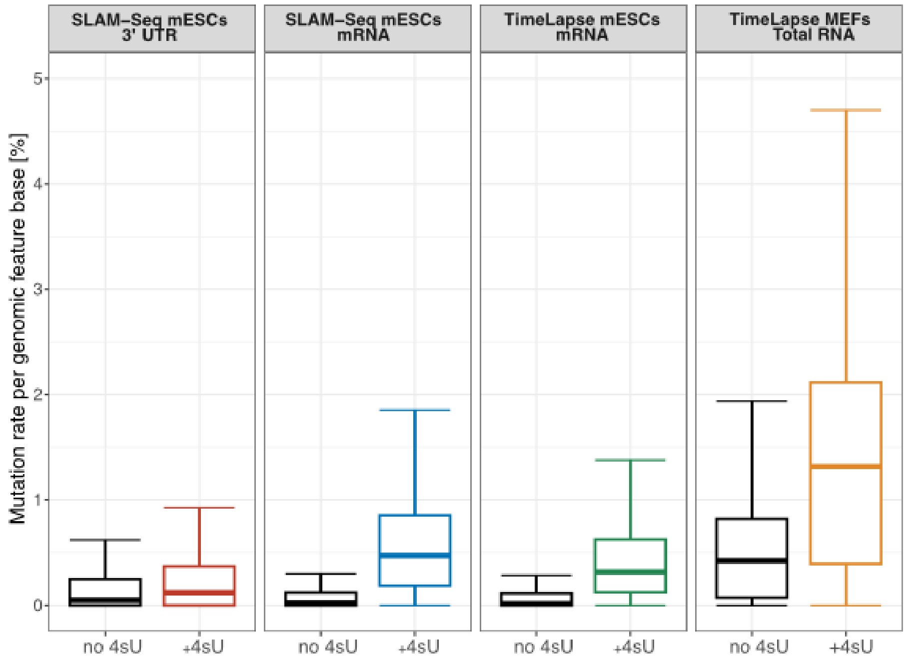

Figure 8: T $>$ C conversion rates for all libraries. Conversion rates per base within defined counting window-mapping reads depending on the library (3' UTR for QuantSeq; whole transcript for mRNA and total RNA-seq), before (no 4sU) and after metabolic labeling (+4sU) for all datasets.

per million (CPM); highlighting pluripotency OSN genes (Oct4, Sox2 and Nanog), as well as some mESCs housekeeping genes. Describe patterns and focus on housekeeping or OSN genes expression matching or not expected levels given the sample type (Fig. 9).

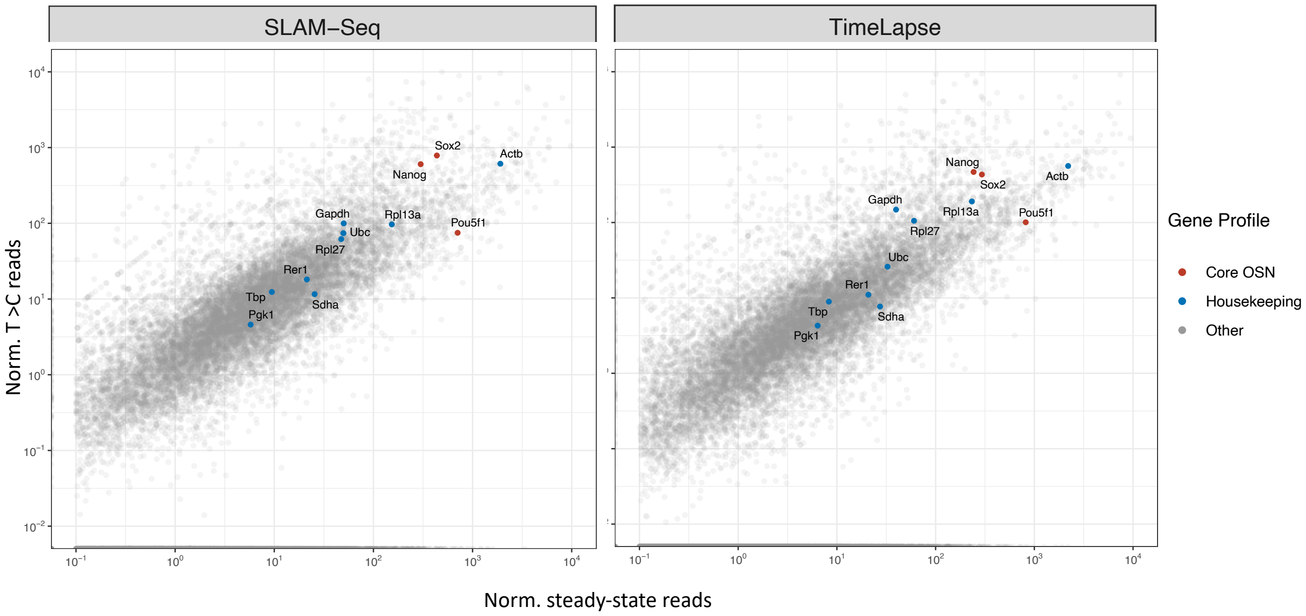

Figure 9: Transcriptional output in mRNA datasets. Transcriptional output for genes in mRNA-seq datasets from SLAM-seq (left) and TimeLapse-seq (right). Norm. T > C reads represent abundance of de novo transcripts in counts per million (cpm) and normalized by gene size; Norm. steady state represents total reads minus-T > C reads in counts per million (cpm) and normalized by gene size. Core OSN (Oct4 (Pou5f1), Sox2, Nanog) pluripotency transcription factors highlighted in red and a gene with housekeeping function in blue.

Chapter 4

Discussion

Discussion The use of nucleoside analogs like 4-thiouridine (4sU) for metabolic labeling of newly synthesized RNA enables the tracking of RNA dynamics (Kawata et al., 2020). By chemical treatment of the 4sU containing nascent RNA prior to sequencing, T-to-C conversions are induced that can be detected by NGS sequencing. This enables the differentiation between total and nascent RNAs through bioinformatic detection, providing valuable insights into the kinetics of RNA synthesis, processing, and decay. In this project, I (i) developed a versatile and automated pipeline to analyze nascent RNA sequencing data from metabolic labeling experiments across multiple RNA-seq library preparation protocols, addressing the accurate quantification of nucleotide conversions within high-throughput RNA-seq datasets generated from non-3' mRNA library preparation protocols (mRNA-seq and total RNA-seq) and (ii) used it to compare different 4sU metabolic labeling based methods (specifically SLAM-seq and TimeLapse-seq)..

The reported pipeline for the identification of T $>$ C conversions allowed us to overcome challenges regarding nucleotide-conversion analysis and pipeline automation. First, built around the Snakemake framework, I achieved automation and high reproducibility of the analysis while maintaining the adaptability via input-based sample characterization. The Snakemake structure allowed us to perform parallel processing of each sample within the dataset compatible with standard and high-performance computing (HPC) softwares, ensuring computational scalability. Moreover, contributing to its reproducibility, the pipeline can be downloaded as a template from GitHub (see Materials and Methods).

Secondly, I optimized the data processing steps in order to establish a versatile pipeline capable of analyzing non-3' sequencing protocols and in particular paired-end sequencing data, coupled with the ability to selectively define filtering and quantification windows depending on the RNA sequencing protocol used. This allows the pipeline to be used for different sequencing strategies such as mRNA-seq or total RNA-seq, irrespective of factors such as sequencing technology or read length. This approach eliminates the necessity for protocol-specific adjustments or data segmentation, as the methodology inherently executes customized computations for each sample within a diverse dataset. This flexibility was validated during our data analysis, where we processed our internally generated mRNA-seq datasets along with the publicly available total RNA-seq TimeLapse-seq data.

Third, I demonstrated the accurate quantification of nucleotide conversions for different nascent RNA-seq technologies through application of my pipeline to various datasets, including re-analysis of the original published datasets from SLAM-seq and TimeLapse-seq methodologies, respectively. As shown on Figure 8, the pipeline was able to detect conversion rates on different datasets and we obtained comparable results to the previously reported conversion rates for SLAM-seq (Herzog et al., 2017) and TimeLapse-seq (Schofield et al., 2018). Thus we can reliably differentiate stable and nascent RNA populations regardless of the sequencing strategy or labeling method.

Additionally, we generated datasets using identical cell types, labeling times, and sequencing protocols to enable a direct comparison between SLMA-seq and TimeLapse-seq (see Materials and Methods). Interestingly, while, as described in chapter 1.3, both methods can in principle be used to detect nascent RNA, we obtained higher T $>$ C read counts and conversion rates for SLAM-seq labeling when compared to TimeLapse-seq (Table 3; Fig. 7-8). This finding holds relevance for guiding the design of forthcoming laboratory experiments. Furthermore, when examining conversion rates across all four datasets, the re-analyzed published datasets showed the lowest and highest conversion rates, with QuantSeq (3' mRNA library preparation protocol) SLAM-seq having the lowest and total RNA-seq TimeLapse-seq having the highest (Fig. 8). However, the fold-changes between the absence of 4sU and its presence (+4sU) remained consistent across datasets relative to the conversion rates in treated (+4sU) samples.

Finally, the pipeline generates informative outputs regarding not only processing steps but also quality controls, while maintaining its compatibility with further downstream analysis. As part of the outputs, standard RNA-seq as well as nucleotide-conversion oriented quality controls are integrated on the MultiQC report output, containing an overview of these quality controls generated through the whole data analysis. Furthermore, output generation such as count matrices for (i) total read counts, (ii) nascent read counts and (iii) nascent/total read counts, were implemented to enhance the utility of the pipeline, providing useful results for downstream analysis (e.g. differential expression, multivariate data analysis).

By offering compatibility with different sequencing library protocols and metabolic labeling methods, this pipeline broadens the scope of research possibilities and supports the exploration of gene expression dynamics across diverse experimental designs. While the initial focus is on accurate processing of metabolically labeled datasets, the pipeline lays the foundation for future downstream analyses related to nucleotide conversion. Additionally, future integration of our pipeline with time course analysis holds the potential for a multi-dimensional approach to decipher the intricate dynamics of gene expression. Through the incorporation of RNA decay measurements, this strategy could uncover the comprehensive life cycle of RNA molecules, spanning from their genesis during transcription to their eventual degradation over time. This comprehensive approach, merging nucleotide labeling, detection, and time course analysis, will enable a deep exploration of transcriptional activity, RNA stability and decay patterns, thereby enhancing our understanding of the regulatory mechanisms governing gene expression changes.

While sequence composition methods like SLAM-seq and TimeLapse-seq exhibit flexibility and applicability, they may introduce biases due to nucleotide analog effects on RNA metabolism and the need for meticulous optimization (Watson et al., 2021). In contrast, computational strategies tailored for conventional RNA-seq datasets, like the exon-intron split analysis (EISA; Gaidatzis et al., 2015), offer an alternative for studying transcriptional and post-transcriptional gene expression regulation. EISA quantifies changes in mature RNA and pre-mRNA reads, providing an alternative methodology to the previously discussed nascent RNA techniques. This computational approach presents an attractive option to explore gene expression dynamics without the constraints of specific experimental manipulations, thus offering a valuable tool for deciphering the intricacies of gene regulation. Nevertheless, this method operates under the assumption of constant RNA processing rates, attributing changes on these rates entirely to RNA degradation; this assumption becomes a strong limitation when studying conditions where RNA is regulated at processing and degradation levels (Furlan et al., 2021).

In conclusion, the presented pipeline contributes to the field of RNA sequencing analysis, providing a powerful tool to uncover the intricacies of gene expression regulation by reporting not only nascent transcription, but also steady-state RNA levels. Its adaptability and comprehensive structure make it a valuable resource for the analysis of nascent RNA sequencing data from metabolic labeling experiments, promoting reproducibility and utility regardless of the experimental design, overall facilitating deeper insights into the molecular mechanisms governing gene expression dynamics.

Chapter 5

Materials and Methods

Sample generation

Mouse embryonic stem cells (mESCs) mRNA datasets were in-house generated by Huiwen Li. For these datasets, 4sU metabolic labeling in mESCs was performed by incubating mESCs in standard medium but adding 4sU to a final concentration of 100 uM. Cells were harvested followed by total RNA extraction using TRIzol, followed by chemically treatment to generate nucleotide conversions accordingly to two labeling protocols (alkylation for SLAM-seq and oxidative-nucleophilic-aromatic-substitution for TimeLapse-seq), subsequently ethanol precipitated and subjected to mRNA library preparation and high throughput sequencing.

Datasets

For validation, we used published datasets from SLAM-seq and TimeLapse-seq original publications (Herzog et al., 2017; Schofield et al., 2018; respectively). SLAM-seq dataset contains 6 samples generated by performing 45 min 4sU-pulse labeling in haploid mESCs at a final concentration of 100 uM and QuantSeq 3' mRNA library preparation (GEO accession: GSE99972). Samples from TimeLapse-seq dataset were supplemented with 4sU (1mM) for 1 hr in MEFs and sequenced after total RNA library preparation (GEO accession: GSE95854, samples: GSM2843697, GSM2843698, GSM2843701, GSM2843702, GSM2843705, GSM2843706, GSM2843709 and GSM2843710).

Snakemake workflow

All data processing steps were implemented inside a Snakemake v7.25.3 (Mölder et al., 2021) pipeline. Snakemake was used with –use-conda option to create the specified software environments via Conda/Bioconda. In addition, particular to the High-performance computing (HPC) cluster used, flags –profiles and -j were added for job submission and resource management requirements via Slurm (Yoo et al., 2003). Total and mRNA datasets were processed at the same time on the same Snakemake pipeline while Quant-seq samples were processed separately on an adapted version of our pipeline to account for the single-end reads and particular annotation files to be used.

For the input, a configuration file was written to specify mandatory parameters and extra arguments for each job comprehending the pipeline. For usage convenience, it follows a tool-based structure, indicating which parameters are necessary for a given tool as well as default values and options when required. Allowing us to track and change the processing parameters without modifying the source code and keeping all processing information inside a single file that can easily be handled and addressed. Additionally, the sample manifest is a tab-separated table containing information about the sample set , in which some of the column names and values are directly linked to the pipeline and used either for conditional command execution or wildcards definition.

Sequencing data pre-processing

Trim Galore! v0.6.6 (https://github.com/FelixKrueger/TrimGalore) was used to trim adapters, removal of short reads ($<$20 nt) and quality filtering of initial raw NGS reads using a Phred score threshold of 20. Quality argument (-q or –2colour) was conditionally determined for all datasets depending on the sequencing technology indicated on the sample manifest table.

Mapping

The overall pipeline was based on the SLAM-DUNK v0.3.2 (Herzog et al., 2017) software tool. Trimmed reads were mapped to GRCm38 reference genome using NextGenMap v0.5.0 (Sedlazeck et al., 2013) directly, instead of slamdunk map, to allow paired-end data as input proper. For proper identification of T > C conversions option –slam-seq 2 setting a conversion-aware mode, using the following scoring scheme and was implemented on the original SLAM-DUNK method to avoid both mismatch penalty or match score for T > C conversions:

		Reference genome			
		A	T	G	C
Read position	A	10	-15	-15	-15
	T	-15	10	-15	-15
	G	-15	-15	10	-15
	C	-15	0	-15	10

Additionally, particular BAM/SAM tags are included indicating the type and number of conversions for proper identification on further analysis.

As for the filtering step, an interval tree is used to identify overlapping multimap reads within the provided mapping windows, removing any reads that align to more than 1 window and reads aligned to non-relevant regions (i.e. not annotated in the supplied reference). As a result, any multimappers with alignments to both single annotated and non-relevant regions will be unequivocally assigned to the single region. Instead, when multiple alignments to a single region are reported, one will be chosen at random; and for cases where a read maps to several mapping windows, the model is unable to reassign the read uniquely and therefore discards it from the analysis.

Annotation tables generation

The pipeline requires two different annotation tables on BED file format to define mapping and counting windows. In order to avoid versioning issues for annotation and genome compatibility and keep the pipeline as reproducible as possible with the minimum amount of inputs, we decided to build our own tables from GTF annotation files. Since the mapping windows can be either exonic or whole transcript regions, depending on the sequencing protocol, the pipeline identifies the sequencing method from the sample manifest and conditionally creates an exon or transcript-level mapping windows annotation. As, for the counting windows, a transcript-level annotation is used in all instances.

The creation of these tables consists of a very basic command line parsing of GTF files by feature, extracting values following the BED file format: chromosome, genomic coordinates, corresponding Ensembl Gene ID and strand. Additionally we create a corresponding gene-level annotation table including biological relevant information (e.g. gene name, biotype, gene size, etc.) that is not used in our pipeline but can be used for downstream analysis.

SNP calling

As described, the VarScan 2.4.1 (Koboldt et al., 2012) based -slamdunk snp- module was used to perform SNP masking. For correct variant fraction consideration we used a threshold of 0.2 for diploid samples (mRNA and total RNA) and 0.8 for haploids (Quant-seq), these values were based on the original method establishment recommendations (Herzog et al., 2017; Neumann et al., 2019). The coverage threshold was kept as default (10).

Background subtraction and normalization

Background error subtraction was performed first by calculating the mean T > C conversions per gene entry over different biological replicates from the same untreated sample, then this mean background signal was directly substrate from the corresponding treated samples. Sample correspondence was established via sample manifest. For gene size normalization, gene length was taken from genomic coordinates and then used to scale both total and nascent counts.

Conversion rates computation

Conversion rates were calculated using -alleyoop utrrates- (Neumann et al., 2019), using the counting windows annotation file to delimit transcripts as feature of interest. Withind alleyoop every the amount of conversions for a given nucleotide conversion is separated by strand so A $>$ G conversions can be identify as T $>$ C and normalized by all the possible conversions for the original nucleotide:

e.g.

$$A \to G = \frac{A \to G}{A \to A + A \to G + A \to C + A \to T}$$

Pipeline availability

A template version of the pipeline reported in this project can be found in GitHub as paurosales/labeled-nascent-rnaseq-snakemake-pipeline.

Bibliography

[1] B. Alberts. *Molecular Biology of the Cell.* Garland Science. Google-Books-ID: jK6UBQAAQBAJ.

[2] T. Alpert, L. Herzel, and K. M. Neugebauer. Perfect timing: splicing and transcription rates in living cells: Splicing and transcription rates in living cells. 8(2):e1401.

[3] F. Baylis. To publish or not to publish. 38(3):271–271.

[4] L. J. Core, J. J. Waterfall, and J. T. Lis. Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. 322(5909):1845–1848.

[5] E. H. Davidson. Emerging properties of animal gene regulatory networks. 468(7326):911–920. Number: 7326 Publisher: Nature Publishing Group.

[6] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame. Nextflow enables reproducible computational workflows. 35(4):316–319.

[7] L. Dölken, Z. Ruzsics, B. Rädle, C. C. Friedel, R. Zimmer, J. Mages, R. Hoffmann, P. Dickinson, T. Forster, P. Ghazal, and U. H. Koszinowski. High-resolution gene expression profiling for simultaneous kinetic parameter analysis of RNA synthesis and decay. 14(9):1959–1972.

[8] S. R. Eddy. Non–coding RNA genes and the modern RNA world. 2(12):919–929.

[9] P. A. Ewels, A. Peltzer, S. Fillinger, Patel, Alneberg, Wilm, Garcia, P. Di Tommaso, and S. Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. 38(3):276–278.

[10] M. Furlan, S. de Pretis, and M. Pelizzola. Dynamics of transcriptional and post-transcriptional regulation. 22(4):bbaa389.

[11] D. Gaidatzis, L. Burger, M. Florescu, and M. B. Stadler. Analysis of intronic and exonic reads in RNA-seq data characterizes transcriptional and post-transcriptional regulation. 33(7):722–729.

[12] D. Gaidatzis, L. Burger, M. Florescu, and M. B. Stadler. Analysis of intronic and exonic reads in RNA-seq data characterizes transcriptional and post-transcriptional regulation. 33(7):722–729.

[13] T. Harada, Y. Heshmati, J. Kalfon, J. X. Ferrucio, M. Perez, J. Ewers, J. M. Ellegast, J. S. Yi, A. Bowker, Q. Zhu, K. Eagle, J. M. Dempster, G. Kugener, J. Wickramasinghe, Z. T. Herbert, C. H. Li, J. V. Koren, V. R. Paralkar, B. Nabet, C. Y. Lin, N. V. Dharia, K. Stegmaier, and M. Pimkin. A distinct core regulatory module enforces oncogene expression in KMT2a-rearranged leukemia.

[14] L. He and G. J. Hannon. MicroRNAs: small RNAs with a big role in gene regulation. 5(7):522–531.

[15] V. A. Herzog, B. Reichholf, T. Neumann, P. Rescheneder, P. Bhat, T. R. Burkard, W. Wlotzka, A. Von Haeseler, J. Zuber, and S. L. Ameres. Thiol-linked alkylation of RNA to assess expression dynamics. 14(12):1198–1204.

[16] J.-P. Hsin and J. L. Manley. The RNA polymerase II CTD coordinates transcription and RNA processing. 26(19):2119–2137.

[17] M. Jackson, K. Kavoussanakis, and E. W. J. Wallace. Using prototyping to choose a bioinformatics workflow management system. 17(2):e1008622.

[18] I. Jonkers and J. T. Lis. Getting up to speed with transcription elongation by RNA polymerase II. 16(3):167–177.

[19] K. Kawata, H. Wakida, T. Yamada, K. Taniue, H. Han, M. Seki, Y. Suzuki, and N. Akimitsu. Metabolic labeling of RNA using multiple ribonucleoside analogs enables the simultaneous evaluation of RNA synthesis and degradation rates. 30(10):1481–1491.

[20] D. C. Koboldt, Q. Zhang, D. E. Larson, D. Shen, M. D. McLellan, L. Lin, C. A. Miller, E. R. Mardis, L. Ding, and R. K. Wilson. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. 22(3):568–576.

[21] J. Köster and S. Rahmann. Snakemake—a scalable bioinformatics workflow engine. 28(19):2520–2522.

[22] M. Levine and R. Tjian. Transcription regulation and animal diversity. 424(6945):147–151. Number: 6945 Publisher: Nature Publishing Group.

[23] F. Ma, B. K. Fuqua, Y. Hasin, C. Yukhtman, C. D. Vulpe, A. J. Lusis, and M. Pellegrini. A comparison between whole transcript and 3' RNA sequencing methods using kapa and lexogen library preparation methods. 20(1):9.

[24] F. Mignone, C. Gissi, S. Liuni, and G. Pesole. Untranslated regions of mRNAs.

[25] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-seq. 5(7):621–628.

[26] F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz, A. Wilm, M. Holtgrewe, S. Rahmann, S. Nahnsen, and J. Köster. Sustainable data analysis with snakemake. 10:33.

[27] T. Neumann, V. A. Herzog, M. Muhar, A. Von Haeseler, J. Zuber, S. L. Ameres, and P. Rescheneder. Quantification of experimentally induced nucleotide conversions in high-throughput sequencing datasets. 20(1):258.

[28] T. Nojima, T. Gomes, A. Grosso, H. Kimura, M. Dye, S. Dhir, M. Carmo-Fonseca, and N. Proudfoot. Mammalian NET-seq reveals genome-wide nascent transcription coupled to RNA processing. 161(3):526–540.

[29] M. Rabani, J. Z. Levin, L. Fan, X. Adiconis, R. Raychowdhury, M. Garber, A. Gnirke, C. Nusbaum, N. Hacohen, N. Friedman, I. Amit, and A. Regev. Metabolic labeling of RNA uncovers principles of RNA production and degradation dynamics in mammalian cells. 29(5):436–442.

[30] J. A. Schofield, E. E. Duffy, L. Kiefer, M. C. Sullivan, and M. D. Simon. TimeLapse-seq: adding a temporal dimension to RNA sequencing through nucleoside recoding. 15(3):221–225.

[31] B. Schwalb, M. Michel, B. Zacher, K. Frühauf, C. Demel, A. Tresch, J. Gagneur, and P. Cramer. TT-seq maps the human transient transcriptome. 352(6290):1225–1228.

[32] F. J. Sedlazeck, P. Rescheneder, and A. Von Haeseler. NextGenMap: fast and accurate read mapping in highly polymorphic genomes. 29(21):2790–2791.

[33] R. J. Sims, R. Belotserkovskaya, and D. Reinberg. Elongation by RNA polymerase II: the short and long of it. 18(20):2437–2468. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

[34] R. Stark, M. Grzelak, and J. Hadfield. RNA sequencing: the teenage years. 20(11):631–656.

[35] S. Tandonnet and T. T. Torres. Traditional versus 3' RNA-seq in a non-model species. 11:9–16.

[36] H. Ura, S. Togi, and Y. Niida. A comparison of mRNA sequencing (RNA-seq) library preparation methods for transcriptome analysis. 23(1):303.

[37] Z. Wang, M. Gerstein, and M. Snyder. RNA-seq: a revolutionary tool for transcriptomics. 10(1):57–63.

[38] M. J. Watson, Y. Park, and C. C. Thoreen. Roadblock-qPCR: a simple and inexpensive strategy for targeted measurements of mRNA stability. 27(3):335–342.

[39] E. M. Wissink, A. Vihervaara, N. D. Tippens, and J. T. Lis. Nascent RNA analyses: tracking transcription and its regulation. 20(12):705–723.

[40] A. B. Yoo, M. A. Jette, and M. Grondona. SLURM: Simple linux utility for resource management. In D. Feitelson, L. Rudolph, and U. Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, volume 2862, pages 44–60. Springer Berlin Heidelberg. Series Title: Lecture Notes in Computer Science.

Supplementary Material

Column	Datatype	Description
Chromosome	*chr*	Chromosome on which the transcript resides
Start	*dbl*	Start position of the transcript (0-based)
End	*dbl*	End position of the transcript (exclusive, 0-based)
Name	*chr*	Ensembl ID of the corresponding gene for the transcript
Length	*dbl*	Length of the transcript
Strand	*chr*	Strand of the transcript
ConversionRate	*dbl*	*ConversionsOnTs* / *CoverageOnTs* for the given transcript
ReadsCPM	*dbl*	Number of reads that mapped to the transcript normalized by library size of retained reads after filtering (counts per million, CPM)
Tcontent	*dbl*	Number of Thymines within the transcript
CoverageOnTs	*dbl*	Cumulative coverage on each Thymine of the transcript
ConversionsOnTs	*dbl*	Cumulative number of T>C conversions in the transcript
ReadCount	*dbl*	Number of reads mapping to the transcript
TcReadCount	*dbl*	Number of reads mapped to the transcript with at least k T>C conversions (T>C reads)
multimapCount	*dbl*	Number of retained reads considered as multimappers mapping to the transcript
ConversionRateLower	*dbl*	Lower bound confidence interval for transcript (not used)
ConversionRateUpper	*dbl*	Upper bound confidence interval for transcript (not used)

Table S1: Output count file content description. Column description for the tab-separated tcount files generated after *slam_count* step. For tables generated after *alley_collapsed* the columns keep the same structure but values are know at gene-level annotation, collapsing all transcripts from a given gene to the same entry.

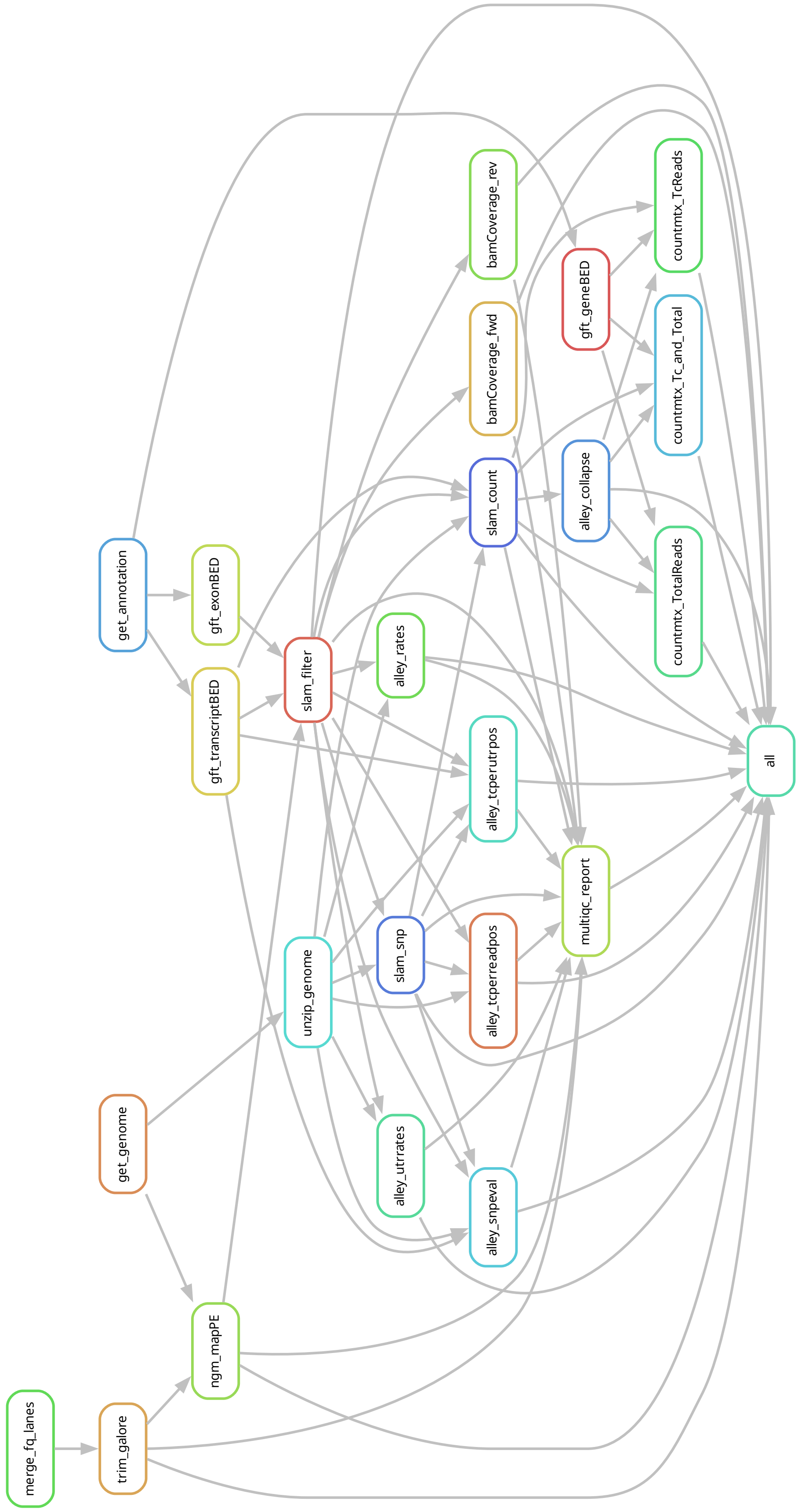

Figure S1: Pipeline for T > C conversions in paired-end RNA-seq data Snakemake rulegraph. Snakemake directed acyclic graph (DAG) indicating file dependencies and an input-output defined computation order. Each rule (box) represents a shell command or R script used to generate the desired output, during the analysis, all rules shown are computed in parallel for each sample within the dataset.

Column	Datatype	Description
Chromosome	*chr*	Chromosome on which the gene resides
Start	*dbl*	Start position of the gene (0-based)
End	*dbl*	End position of the gene (exclusive, 0-based)
Name	*chr*	Gene Ensembl ID
Strand	*chr*	Strand of the gene
Length	*dbl*	Length of the gene
Symbol	*dbl*	Gene Ensembl symbol
Biotype	*dbl*	Gene Ensembl biotype
avgReadsCPM	*dbl*	Average *ReadsCPM* within the transcript compromising the gene
avgTcontent	*dbl*	Average number of Thymines within the transcript compromising the gene
avgCoverageOnTs	*dbl*	Average coverage on each Thymine of the transcripts compromising the gene
avgMultimapper	*dbl*	Average number of retained reads considered as multimappers mapping to the transcript
SAMPLES	*dbl*	Gene counts for all given samples (one column per sample)

Table S2: Output count matrices content description. Column description for the tab-separated count matrices files for counts from (i) *ReadCount*, (ii) *TcReadCount* and (iii) *ReadCount/TcReadCount* entries on the tcount files generated after *alley_merge* steps.