Check for updates

SOFTWARE TOOL ARTICLE

REVISED SCAPE-APA: a package for estimating alternative polyadenylation events from scRNA-seq data

[version 3; peer review: 1 not approved]

Guangzhao Cheng[1*], Tien Le[1*], Ran Zhou[2], Lu Cheng[1,3]

[1]Department of Computer Science, Aalto University, Espoo, Uusimaa, 02150, Finland
[2]State Key Laboratory of Biotherapy and Cancer Center, Sichuan University West China Medical Center, Chengdu, Sichuan, China
[3]Institute of Biomedicine, University of Eastern Finland, Kuopio, 70211, Finland

[*] Equal contributors

v3 **First published:** 16 Oct 2024, **4**:220
https://doi.org/10.12688/openreseurope.18673.1
Second version: 24 Dec 2024, **4**:220
https://doi.org/10.12688/openreseurope.18673.2
Latest published: 07 Mar 2025, **4**:220
https://doi.org/10.12688/openreseurope.18673.3

Abstract

SCAPE was a package that was previously developed to estimate alternative polyadenylation events from single-cell RNA-seq (scRNA-seq) data. It was composed of ad-hoc python scripts and has speed issues when handling large scRNA-seq data. To suit the needs of analyzing large scRNA-seq datasets, we present SCAPE-APA, which is a reimplementation of SCAPE with substantial changes. We made the following updates to the package: (1) we binned similar reads together to accelerate the estimation; (2) we re-derived the mixture model to tailor it for binned reads; (3) we implemented the inference algorithm using Taichi language for acceleration; (4) we re-implemented the untranslated region (UTR) annotation extraction script using the professional package gffutils for better maintenance; (5) we wrote a script to detect spurious alternative polyadenylation sites generated due to junction reads, and (6) we created a formal Python package and uploaded it to the Python Package Index website (Pypi).

Plain language summary

SCAPE-APA is a Python software package designed to detect alternative polyadenylation events from single-cell RNA sequencing data. It uses advanced bioinformatics and computational tools to make the analysis process faster and more efficient. SCAPE-APA is also available on the Python Package Index (PyPi) website, making it easy to install.

Keywords

SCAPE, alternative polyadenylation, scRNA-seq

Open Peer Review

Approval Status ✗

	1
version 3 (revision) 07 Mar 2025	
version 2 (revision) 24 Dec 2024	✗ view
version 1 16 Oct 2024	? view

1. **Wenbin Ye**, University of California Irvine, Irvine, USA

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the Software gateway.



This article is included in the Horizon 2020 gateway.



This article is included in the Marie-Sklodowska-Curie Actions (MSCA) gateway.

Corresponding author: Lu Cheng (lu.cheng.ac@gmail.com)

Author roles: Cheng G: Formal Analysis, Methodology, Writing – Review & Editing; **Le T**: Data Curation, Formal Analysis, Software, Validation, Visualization, Writing – Review & Editing; **Zhou R**: Software, Writing – Review & Editing; **Cheng L**: Conceptualization, Funding Acquisition, Methodology, Software, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 663830-CU122. This project has received funding from the Research Council of Finland under grant agreement Nos 335858 and 358086.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2025 Cheng G *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License , which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Cheng G, Le T, Zhou R and Cheng L. **SCAPE-APA: a package for estimating alternative polyadenylation events from scRNA-seq data [version 3; peer review: 1 not approved]** Open Research Europe 2025, **4**:220 https://doi.org/10.12688/openreseurope.18673.3

First published: 16 Oct 2024, **4**:220 https://doi.org/10.12688/openreseurope.18673.1

REVISED **Amendments from Version 2**

Supplementary Material:

- Supplementary Note 1 is now uploaded to Zenodo and properly referenced, addressing the initial omission.

Figures and Legends:

- Revised Figure 1A to integrate and clearly annotate the improvements made over the original SCAPE model.
- Updated the legend of Figure 1A to explain the meanings of SCAPE's parameters
- Updated Figures 1B and 1C with improved legends and color annotations to clarify the detection and merging of spurious pA sites (e.g., merging PA1 into PA2) and to better represent the relationships among isoforms and pA sites.
- Enhanced figure legends by defining abbreviations (e.g., "std" for standard deviation) and explaining Gaussian parameters (mean, standard deviation, weight) with reference to Supplementary Note 1.

Textual and Methodological Clarifications:

- Added a new paragraph discussing SCAPE's limitations, especially with high-read-count 3'-UTRs, to provide the rationale behind the improvements in SCAPE-APA.
- Added new references to the introduction paragraph
- Added a new paragraph explaining SCAPE's new spurious pA site detection feature and provided the concrete number of spurious pA sites in the test data
- Provide detailed information about the experiments regarding the number of reads per 3'-UTR
- Adjusted language for clarity (e.g., using "summarize as" instead of "summarize into").
- Revised descriptions of polyadenylation sites to note they can occur in regions beyond the 3'-UTR.

Any further responses from the reviewers can be found at the end of the article

Introduction

The 3' untranslated region (3'-UTR) of a messenger RNA (mRNA) molecule is a critical segment located downstream of the coding sequence and plays a pivotal role in gene regulation. Polyadenylation sites (pA sites) are specific locations, primarily within the 3'-UTR but also found in introns, coding sequences, and intergenic regions, where the mRNA undergoes a process known as polyadenylation, resulting in the addition of a poly(A) tail to the pA site. Alternative polyadenylation (Mitschka & Mayr, 2022) refers to the phenomenon in which a single gene possesses multiple pA sites within its 3'-UTR, leading to the production of distinct mRNA isoforms with varying 3'-UTR lengths.

These alternative polyadenylation isoforms can have profound implications for post-transcriptional regulation, as they may influence mRNA localization, stability, and interaction with regulatory factors (Chen *et al.*, 2024) (Mitschka & Mayr, 2022), thereby affecting the ultimate functional output of the gene. Understanding the mechanisms and consequences of alternative polyadenylation is essential to unravel the complexities of gene expression and its role in diverse biological processes.

We previously developed the software package SCAPE (Zhou *et al.*, 2022) that can estimate alternative polyadenylation events from single-cell RNA-seq (scRNA-seq) data. Because the reads in scRNA-seq are generally not long enough to cover the pA sites, we used a Bayesian mixture model (Figure 1A) to infer the pA sites using the insert length distribution. The mixture model contained k ($k > 1$) Gaussian components and one uniform component. Each pA site is represented by a Gaussian component, which has three parameters: mean μ, standard deviation δ and weight π. Detailed meanings of these parameters and model specification are provided in Supplmentary Note 1. The full SCAPE analysis workflow can be summarized as the following steps: (1) extract the 3'-UTR annotation from the General Transfer Format (GTF) or General Feature Format (GFF) file; (2) map the reads to the reference genome and calculate the relative positions of reads on the corresponding 3'-UTR regions; (3) perform parameter inference for each 3'-UTR region and output the result; and (4) perform downstream tasks such as differential pA site usage analysis.

However, computational speed remains a challenge for SCAPE in real data analysis. Some 3'-UTRs contain a large number of reads (>200,000), significantly slowing down the inference algorithm. SCAPE often processes most 3'-UTRs efficiently but struggles with a small subset of high-read-count 3'-UTRs. Simple parallelization strategies, such as processing chromosomes independently or using multithreading, are ineffective because SCAPE's inference algorithm is inherently serial. Additionally, SCAPE is written in Python, which is slower than just-in-time compiled languages like Taichi (Hu *et al.*, 2019). Code maintainability is also an issue, particularly in the inference algorithm and 3'-UTR annotation extraction scripts. To address these challenges, we developed a new version of SCAPE.

Methods

Improvements

We made significant updates to the previous implementation, as shown in Figure 1A. The biggest challenge in real data analysis is that the read number can be very large ($n > 200{,}000$) for some 3'-UTRs. This makes the original inference algorithm extremely slow. As the 3'-UTR is relatively short (~1500bp), there are many redundant reads. To solve this problem, we binned reads in the vicinity, that is, reads with similar x, l, r values in the following likelihood.

$$p(Z, \boldsymbol{x}, \boldsymbol{l}, \boldsymbol{r} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, M_1) = \prod_{n=1}^{N} \prod_{k=1}^{K} \{\pi_k \, p(x_n, l_n, r_n \mid \alpha_n, \beta_n, M_1)\}^{Z_{nk}}$$

Note that the bin is 3-dimensional. The nth bin was denoted by the unique combination (x_n, l_n, r_n) and its count by c_n, where the bin sizes were set to 5bp, 10bp, 10bp, respectively. The joint likelihood of the mixture model can be rewritten as:

$$p(Z, \boldsymbol{x}, \boldsymbol{l}, \boldsymbol{r} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}, M_1) = \prod_{n=1}^{N} \prod_{k=1}^{K} \{\pi_k \, p(x_n, l_n, r_n \mid \alpha_n, \beta_n, M_1)\}^{C_n Z_{nk}}$$

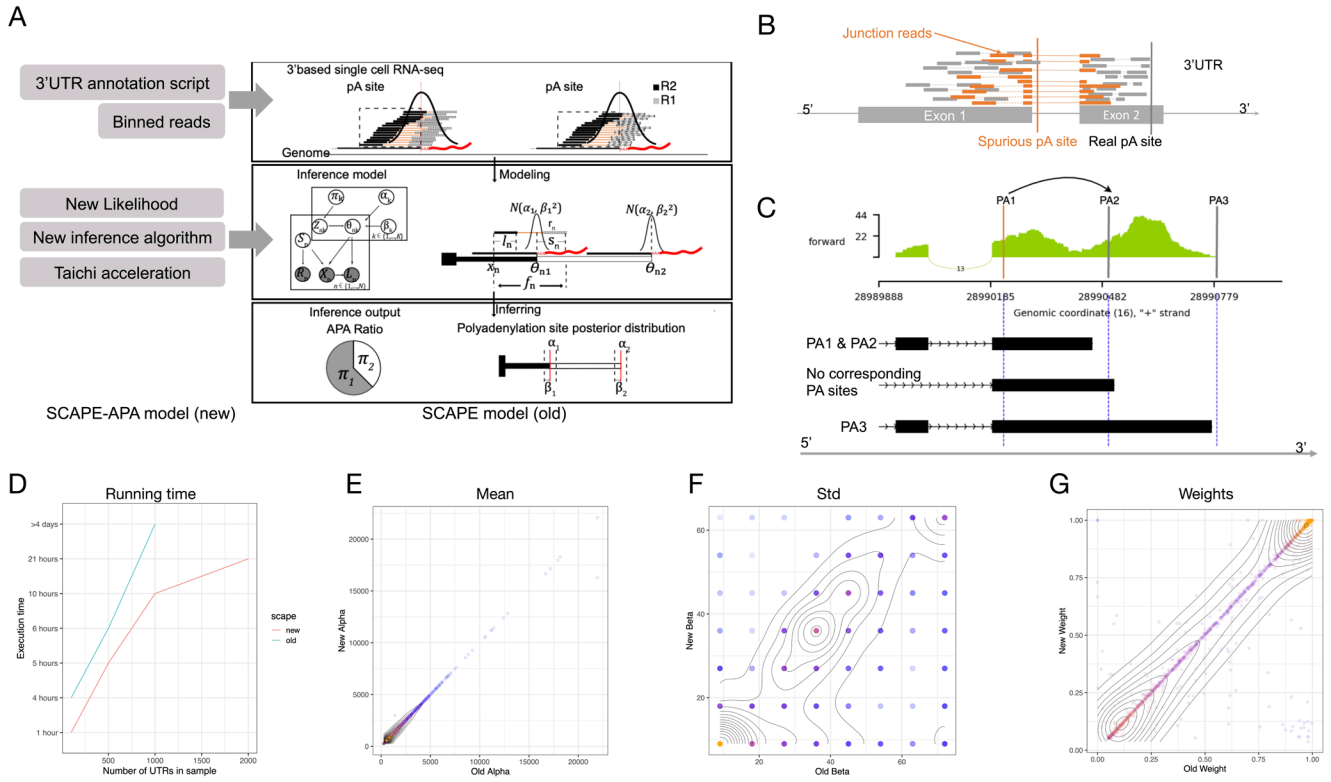


Figure 1. (**A**) Original SCAPE model and improvements of SCAPE-APA. In data preprocessing, we developed a new script to extract 3′-UTR annotations from GTF files and implemented read binning. In modeling, we introduced a new likelihood function for binned reads and derived an updated inference algorithm, as detailed in Supplementary Note 1. In implementation, we rewrote SCAPE using Taichi to enhance computational efficiency. For both SCAPE-APA and SCAPE, the outputs include the estimated mean and standard deviation of pA sites (represented as α and β in the bottom panel). In the SCAPE model, pA site locations are modeled as Gaussian components. (**B**) Schematic illustration of a spurious pA site due to splicing of 3′-UTR. The SCAPE model assumes that each 3′-UTR is a single continuous segment; however, in reality, 3′-UTRs can be multi-segmented. This results in reads that are close together on the transcript appearing far apart on the genome, violating the SCAPE model's assumption and potentially leading to spurious pA sites. In the figure, orange represents junction reads, while gray indicates normal reads. (**C**) A real example of a spurious pA site. PA1 is the spurious pA site and should be merged into PA2. The bottom panel displays 3′-UTR annotations from the GTF file, showing one single-segment 3′-UTR transcript and two multi-segment 3′-UTR transcripts. PA1 and PA2 are most likely derived from the first transcript, necessitating the merging of PA1 into PA2. (**D**) Comparison of running time across different datasets on the same computer. Each 3′-UTR in these datasets contains more than 20,000 reads. All datasets were analyzed using a single CPU core with 10 GB of memory. (**E**–**G**) Comparison of estimated mean, standard deviation (std) and weight of estimated pA sites. The curves are the density contour of data points. The means of the Gaussian components indicate the predicted pA site locations. For a detailed explanation of these parameters, please refer to Supplementary Note 1.

It can be seen that we only need to calculate the likelihood once for all c_n reads in the nth bin in the new likelihood, whereas this has to be repeated c_n times in the original likelihood. As the likelihood changes, the inference algorithm must be modified accordingly. The detailed mathematical derivations of the inference algorithm for the new likelihood are provided in Supplementary Note 1 (see *Extended data*). To further accelerate the inference, we used the Taichi language (Hu *et al.*, 2019) to re-implement the inference algorithm, where Taichi used the just-in-time (JIT) compilation to accelerate the Python code. Because the inference algorithm is serial, the computation times are similar, regardless of whether the GPU is used.

In addition to the new likelihood and inference algorithm, we re-implemented and added new bioinformatics functionalities to streamline the workflow. We re-wrote the Python script to extract 3'-UTR annotations from GTF or GFF files based on the package "gffutils" (https://github.com/daer/gffutils), which was easier to maintain than the previous ad hoc implementation. We added a post-processing script to handle spurious pA sites owing to the junction reads. As shown in Figure 1B, some 3'-UTR may contain multiple exons/fragments, while the SCAPE model assumes the 3'-UTR to be a consecutive region on the genome, that is, we take the start of the first exon/fragment and the end of the last exon/fragment as the full 3'-UTR region. Because of this multi-segment property of the given 3'-UTR in Figure 1B, two pA sites were estimated from the data, where the orange pA site was the spurious pA site. To resolve this, we merged a pA site with the next downstream pA site if there were > 40% junction reads in all reads assigned to this

pA site. Figure 1C shows a spurious pA site case in the real data, where the first estimated pA site should be merged with the second. Finally, we made a python package "scape-apa" and uploaded it to the Python Package Index (PyPI) to facilitate the installation.

We compared the speed and inference results of the old and new SCAPE software by sampling 100, 500, 1000, and 2000 3'-UTRs from public scRNA-seq data (Notaras *et al.*, 2022). Each 3'-UTR analyzed contains at least 20,000 reads to effectively illustrate the speed difference between the new and old versions of SCAPE. Figure 1D shows the running times for the analyses of these datasets on a CPU-only computer (Dell PowerEdge C4130), with the maximum memory set to 10GB. It can be seen that the speedup is 9-fold for the dataset with 1000 3'-UTRs. Figure 1E–G shows the estimated mean, std, and weights given by the old and new SCAPE packages of the dataset with 1000 3'-UTR. We fixed the number of pA sites in each UTR based on the results of the new package. It can be seen that the majority of the data points are scattered around the $y = x$ line for the mean and weight, whereas a larger variation is observed for the standard deviation. These differences are due to the binning operation; however, the overall comparisons suggest a highly consistent result between the old and new packages. Additionally, this benchmark focuses specifically on high-read-count 3'-UTRs, as the performance of both versions remains similar for low-read-count cases.

Spurious pA site detection is a new feature of SCAPE-APA, as the original SCAPE package does not detect spurious pA sites. In our test dataset (Notaras *et al.*, 2022), SCAPE-APA identified 191 spurious pA sites out of 1009 total pA sites across 1000 3'-UTRs, each containing at least 20,000 reads in scRNA-seq data. In summary, we revised the statistical model for binned reads and utilized Taichi to accelerate the parameter inference of SCAPE, as well as developed useful Python scripts to process GTF files and handle spurious pA sites.

Implementation

The SCAPE-APA package has five main modules:

- **utils.py**: This module processes GFF files from the Ensembl database into 3'-UTR annotations using the dedicated APIs of **gffutils**. If no 3'-UTR annotation is available for a gene, the last exon is used as its 3'-UTR.
- **input_processor.py**: This module extracts relevant information from the input BAM files for APA event estimation. This includes variables such as **x, l, r** in the SCAPE model, as well as cell barcode, UMI, and junction read information.
- **apa_core.py**: This module contains the core inference algorithm of the SCAPE model. The ApaModel class implements the core functionality, holding the parameters and results for the model. Its run() function gathers all relevant components for the inference algorithm. Details of the statistical model and inference algorithm are provided in Supplementary Note 1.
- **taichi_core.py**: This module provides equivalent implementations of relevant functions from the inference algorithm using the Taichi language. Since Taichi is a static language, all memory for inputs and outputs must be pre-allocated before calling a Taichi function. These functions are imported by the apa_core.py module to perform the actual inference.
- **junction_handler.py**: This module merges spurious pA sites that arise due to junction reads. If more than 40% of the reads assigned to a given pA site are junction reads, the site is labeled as spurious and merged with the next pA site.

Operation

SCAPE-APA can be installed on Linux, Mac, and Windows systems with Python 3.11 or higher. A GPU is not required to run SCAPE-APA. The basic workflow consists of five steps:

1. Generate the 3'-UTR annotation file using the "scape gen_utr_annotation" command. The input file of this step is GFF3 file of suitable genomes downloaded from Ensembl database.
2. Prepare the input files using the "scape prepare_input" command. The 3'-UTR annotation file from step 1 and the BAM file from the scRNA-seq experiment are the inputs for this step.
3. Perform pA site inference using the "scape infer_pa" command.
4. Remove spurious pA sites caused by junction reads using the "scape merge_pa" command.
5. Extract the pA count matrix using the "scape ex_pa_cnt_mat" command. The rows represent cells, and the columns represent the estimated pA sites. The pA count matrix can be fed into Seurat for differential analysis.

Discussion

Since 2019, the scale of scRNA-seq data has expanded dramatically, growing from datasets with thousands of cells to those containing millions in 2024. This surge in data volume has highlighted the limitations of the original SCAPE tool, which was not equipped to handle such large datasets. In fact, the original SCAPE was extremely slow, sometimes taking weeks to analyze a single dataset with just thousands of cells. Moreover, it contained several bugs that compromised its performance, and the reliance on ad-hoc Python scripts made maintenance difficult, especially as databases like Ensembl regularly update their annotations. Using a dedicated API to extract 3'-UTR annotations is a far more robust and reliable solution for long-term sustainability.

SCAPE-APA was developed to address these critical issues. It offers substantial improvements in speed, reliability, and maintainability, making it far better suited for the analysis of large-scale scRNA-seq datasets. By incorporating modern bioinformatic tools and creating a formal Python package,

SCAPE-APA enables faster and more efficient analysis while reducing maintenance challenges. This updated version will be an invaluable tool for researchers, capable of handling the increasingly large and complex datasets in single-cell RNA sequencing, ensuring that SCAPE-APA meets the growing demands of the field.

Ethics and consent

Ethical approval and consent not required

Data availability

Extended data

Zenodo: SCAPE-APA Supplementary Note 1. https://doi.org/10.5281/zenodo.14517303 (Cheng, 2024)

This project contains the following extended data:

- Supplementary Note 1. Statistical model of SCAPE-APA

Data are available under the terms of the Creative Commons Zero "No rights reserved" data waiver (CC0 1.0 Public domain dedication).

Availability and implementation

Scape-apa is freely available at https://github.com/chengl7-lab/scape and can be easily installed using conda and pip.

Software availability

Source code available from: https://github.com/chengl7-lab/scape

Archived software available from: https://doi.org/10.5281/zenodo.13898252

License: OSI approved open license software is under the MIT License.

Author contribution

G. C. developed an inference algorithm for binned reads and implemented it in Taichi. T. L. developed Python scripts for parsing the GTF file, removing spurious pA sites, and ran benchmark experiments. R. Z. provided advice regarding the 3'-UTR annotation extraction. L. C. supervised the project.

Acknowledgements

GC, TL, and LC acknowledge the computational resources provided by the Aalto Science-IT project and the CSC – IT Center for Science, Finland.

References

Cheng L: **SCAPE-APA Supplementary Note 1.** *Zenodo*. 2024.
http://www.doi.org/10.5281/zenodo.14517304

Chen H, Wang Z, Gong L, *et al.*: **A distinct class of pan-cancer susceptibility genes revealed by an alternative polyadenylation transcriptome-wide association study.** *Nat Commun.* 2024; **15**(1): 1729.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Hu Y, Li TM, Anderson L, *et al.*: **Taichi: a language for high-performance computation on spatially sparse data structures.** *ACM Trans Graph.* 2019; **38**(6): 1–16.
Publisher Full Text

Mitschka S, Mayr C: **Context-specific regulation and function of mRNA alternative polyadenylation.** *Nat Rev Mol Cell Biol.* 2022; **23**(12): 779–96.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Notaras M, Lodhi A, Dündar F, *et al.*: **Schizophrenia is defined by cell-specific neuropathology and multiple neurodevelopmental mechanisms in patient-derived cerebral organoids.** *Mol Psychiatry.* 2022; **27**(3): 1416–34.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Zhou R, Xiao X, He P, *et al.*: **SCAPE: a mixture model revealing single-cell polyadenylation diversity and cellular dynamics during cell differentiation and reprogramming.** *Nucleic Acids Res.* 2022; **50**(11): e66.
PubMed Abstract | **Publisher Full Text** | **Free Full Text**

Open Peer Review

Current Peer Review Status: ✗

Version 2

Reviewer Report 02 January 2025

https://doi.org/10.21956/openreseurope.20818.r49123

© **2025 Ye W.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✗ **Wenbin Ye**
[1] University of California Irvine, Irvine, California, USA
[2] University of California Irvine, Irvine, California, USA

The authors have not sufficiently addressed the specific questions and concerns outlined in my initial review. Regrettably, the current version of the manuscript falls short of the standards necessary for indexing.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: alternative polyadenylation; computational biology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 02 Jan 2025

Lu Cheng

Dear Wenbin Ye, Thanks a lot for your time in reviewing our paper and happy new year! After receiving your comments, we were advised by the editorial team to submit the response after we have got the second batch of review comments, such that we could address comments of both reviewers in one round. I want to mention there are technical issues with the submission of this manuscript. In the original submission, we provided Supplementary Note 1 as a pdf file. However, it was not attached to version 1 of the paper and we were not aware of it until we received your comments. We contacted the editorial team about this and were advised to upload Supplementary Note 1 to a Zenodo repository. After this update, the manuscript turned out to be version 2. Please be assured that we will response to your comments in the next version and revise the manuscript accordingly. We will contact the editorial team about the status of the second review and hope to provide

the responses as soon as possible. It is the first time that we submit our paper to Open Research Europe. We are unfamiliar with the submission process and sorry for the inconveniences and confusions.
Best regards, Lu Cheng

Competing Interests: No competing interests were disclosed.

Author Response 05 Mar 2025

Lu Cheng

Dear Dr. Wenbin Ye, Thank you very much for your valuable comments! We sincerely appreciate the time and effort you have taken to review our manuscript. Due to technical reasons, *Supplementary Note 1* was not uploaded in the initial submission, which may have made the manuscript difficult to follow and led to some confusion. We sincerely apologize for this oversight. To ensure accessibility, we have now uploaded *Supplementary Note 1* to the following Zenodo repository: https://doi.org/10.5281/zenodo.14517303 Below are our responses to your concerns: Major concerns:

1. Although the authors claim that SCAPE-APA can address the challenges of processing large datasets, the adoption of the SCAPE tool remains limited. Additionally, users can mitigate performance issues by employing alternative strategies, such as splitting chromosomes and implementing multithreading, to handle large datasets more efficiently.

Response: Thank you for your comments! In practice, multithreading and chromosome-based splitting were already implemented. Specifically, SCAPE processes each chromosome separately and further divides the data into chunks, each containing 1000 3’-UTRs by default. Each chunk is then processed independently on a single CPU core. However, in real data analysis, certain 3’-UTRs contain an extremely high number of reads (>200,000), creating bottlenecks. Since SCAPE processes each 3’-UTR in a serial manner on a single core, there is no straightforward way to accelerate the process. We are currently analyzing large-scale scRNA-seq data (~10TB, DUOS dataset, Notaras et al., 2024, https://www.nature.com/articles/s41586-024-07109-5) using the new SCAPE version, which significantly improves performance. In the old version, the average processing time per 3’-UTR was approximately 5 seconds, whereas in the new version, these UTRs are processed in just 0.1 seconds. To address this concern, we have added the following paragraph: *"However, computational speed remains a challenge for SCAPE in real data analysis...."*

1. This manuscript does not fully demonstrate the improved performance of SCAPE-APA in handling large datasets, nor does it provide clear evidence that its performance in predicting pA sites is comparable to or improved over SCAPE. The authors only use 100, 500, 1000, and 2000 3'UTRs to illustrate performance improvements, which may not be convincing given the scale of real-world datasets.

Response: SCAPE processes each UTR independently, so comparison on a given number of UTRs could equally show the speed difference between the old and new version of SCAPE. When the number of UTRs increases, the total running time increases linearly. As mentioned in the response to the previous point, the bottlenecks are UTRs with large number of reads, which are easier to show the speedup of the new version. We have analyzed the aforementioned DUOS scRNA-seq (~10TB). The running time of new SCAPE is

12 hours per 10000 UTRs on a single CPU with 100Gb memory, while the old SCAPE could not finish in 5 days (120 hours). Without the new version of SCAPE, we would not be able to analyze very large scRNA-seq datasets. We add a sentence in the manuscript to discuss this issue. “Additionally, this benchmark focuses specifically on high-read-count 3’-UTRs, as the performance of both versions remains similar for low-read-count cases.”

1. This article proposes an improvement to the SCAPE model (SCAPE-APA), but the introduction only briefly describes the SCAPE method and does not mention its limitations. Please add a section discussing these issues to provide background for SCAPE-APA.

Response: Thank you for your comments. We recognize the need for a more comprehensive background on SCAPE, including its limitations, to better contextualize the improvements introduced in SCAPE-APA. To address this, we have expanded the introduction to discuss SCAPE’s key limitations. Specifically, we highlight its challenges in handling high-read-count 3’-UTRs due to the serial nature of its inference algorithm, its reliance on Python (which is slower than just-in-time compiled languages), and difficulties in maintaining code efficiency for large-scale datasets. These factors motivated the development of SCAPE-APA, which incorporates optimizations to significantly improve computational efficiency. To clarify this point, we have added the following paragraph to the manuscript:
"However, computational speed remains a challenge for SCAPE in real data analysis….”

1. For Figure 1A, the SCAPE diagram is unclear and too small, making it difficult to interpret. Additionally, the diagram does not fully reflect the proposed improvements. Specifically, the peripheral gaze does not align with the steps in the SCAPE model, and the cycle diagram suggests an infinite loop. Please integrate the improvements directly into the original SCAPE diagram rather than describing them only in text.

Response: We apologize for the confusion. To clarify, our improvements span three key areas:

1. Data Preprocessing: We developed a new script to extract 3’-UTR annotations from GTF files and implemented read binning.
2. Modeling: We introduced a new likelihood function for binned reads and derived an updated inference algorithm, as detailed in *Supplementary Note 1*.
3. Implementation: We rewrote SCAPE using *Taichi* to significantly accelerate computation.

To better illustrate these improvements, we have explicitly linked each enhancement to its corresponding step in the SCAPE model within Figure 1A. Additionally, we have revised the diagram to improve clarity, adjusted the peripheral elements for better alignment, and refined the legend to provide more detailed explanations. The updated figure is shown as follows:

https://s3-eu-west-1.amazonaws.com/openreseurope/linked/248099.18673-Lu_Cheng.pdf

This is the revised legend of Figure 1A: “*(A) Original SCAPE model and improvements of SCAPE-APA. In data preprocessing, we developed a new script to extract 3’-UTR annotations from GTF files and implemented read binning. In modeling, we introduced a new likelihood function for binned reads and derived an updated inference algorithm, as detailed in Supplementary Note 1. In implementation, we rewrote SCAPE using Taichi to enhance computational efficiency. For both SCAPE-APA and SCAPE, the outputs include the estimated mean and standard deviation of pA sites*

(represented asa and b in the bottom panel). In the SCAPE model, pA site locations are modeled as Gaussian components."

1. Figure C is confusing. Why do the pA sites (e.g., PA1, PA2, PA3) not correspond to the ends of the isoforms? What is the relationship between the three isoforms and the pA sites (PA1, PA2, PA3)? Furthermore, this figure doesn't clearly explain the rationale for merging PA1 into PA2.

Response: We apologize for the confusion. At the bottom of Figure 1C, the three transcript annotations from the GTF file are shown, with thick black lines indicating 3'-UTRs. PA1 and PA2 primarily correspond to the first transcript, with small probabilities from the second transcript, while PA3 corresponds to the third transcript. The coverage plot at the top shows junction reads, indicating that the 3'UTR of the first transcript is split into two segments. Since SCAPE assumes that each 3'UTR is a continuous segment, multi-segment 3'UTRs pose challenges for the inference algorithm. In this case, PA1 arises from the first segment and represents a spurious pA site, whereas the actual pA site is near PA2, marking the true end of the first transcript. Therefore, PA1 was merged into PA2 to ensure accurate estimation. To improve clarity, we have updated the figure to better illustrate this rationale and revised the legend text accordingly.

https://s3-eu-west-1.amazonaws.com/openreseurope/linked/248099.18673-Lu_Cheng.pdf

This is the revised legend for Figure 1C: "*(B) Schematic illustration of a spurious pA site due to splicing of 3'-UTR. The SCAPE model assumes that each 3'-UTR is a single continuous segment; however, in reality, 3'-UTRs can be multi-segmented. This results in reads that are close together on the transcript appearing far apart on the genome, violating the SCAPE model's assumption and potentially leading to spurious pA sites. In the figure, orange represents junction reads, while gray indicates normal reads.*"

1. More importantly, the authors should give an overall statistical result rather than just showing one example, showing how many spurious pA sites can be detected using SCAPE-APA compared with SCAPE

Response: We apologize for the confusion. The original SCAPE does not handle spurious pA sites, making this a new feature in the updated version of SCAPE-APA. In our test dataset, SCAPE-APA detected 191 spurious pA sites out of 1009 pA sites from 1000 UTRs with at least 20000 reads in scRNA-seq data from (Notaras et al., 2022). To provide a more comprehensive evaluation, we have included an overall statistical summary in the revised manuscript, quantifying the effectiveness of SCAPE-APA in identifying spurious pA sites compared to the original SCAPE. We have added the following paragraph to address this point: "*Spurious pA site detection is a new feature of SCAPE-APA, as the original SCAPE package does not detect spurious pA sites. In our test dataset (Notaras et al., 2022), SCAPE-APA identified 191 spurious pA sites out of 1009 total pA sites across 1000 3'-UTRs, each containing at least 20,000 reads in scRNA-seq data.*"

1. The manuscript states that "The biggest challenge in real data analysis is that the read number can be very large (n > 200,000) for some 3'-UTRs". However, several points need to be addressed: 1) Are all the selected 3'UTR deeply covered (Fig. 1D)? 2) the number of 3'UTR (100, 500, 1000, 2000) tested is relatively small. For example, there are estimated to be around 20,000 3'UTR in the human genome. 3) Although the improved method increases speed, processing 2000 3'UTRs in 21 hours is still unacceptable, especially for the human genome.

Response: Thank you for your comments! Below, we address each of your concerns:

1. Depth of Coverage (Figure 1D): The pA sites shown in Figure 1D correspond to 3'-UTRs with at least 20,000 reads. We have updated the figure legend, as well as the main text, to clarify this in the revised manuscript.

"Each 3'-UTR analyzed contains at least 20,000 reads to effectively illustrate the speed difference between the new and old versions of SCAPE."

1. Number of 3'UTRs Tested: The majority of 3'-UTRs in scRNA-seq data have fewer than 20,000 reads. Both the old and new versions of SCAPE run efficiently on these 3'UTRs, meaning they are not the bottleneck in real data analysis. We have added a sentence in the revised manuscript to emphasize this point.

"Additionally, this benchmark focuses specifically on high-read-count 3'-UTRs, as the performance of both versions remains similar for low-read-count cases."

1. Processing Time: The reported running time refers to execution on a single CPU with 10GB RAM, processing all BAM files. Each of the 3'-UTRs analyzed contains at least 20,000 reads, making them computationally intensive. With 10 CPUs, the processing time is reduced to approximately 2 hours. In our analysis of the DUOS dataset (~1.2 million cells), the inference stage for 10,000 3'UTRs takes 12 hours using 1 CPU core with 100 GB memory. We have updated the figure legend to provide these details.

Overall, we believe the new version of SCAPE can efficiently process large-scale scRNA-seq datasets within a reasonable timeframe.

1. Figures 1E-G show consistent results between the old and new packages using the mean, standard deviation, and weight parameters. However, including polyA site prediction accuracy would give users a clearer understanding of SCAPE-APA's performance and how it improves upon SCAPE.

Response: Thank you for your comment. We understand that including polyA site prediction accuracy might offer additional clarity. In Figure 1E, the means of the Gaussian components represent the predicted pA site locations, while Figures 1F and 1G display the corresponding standard deviation and weight parameters. Our primary goal was to demonstrate consistency between the two versions of SCAPE in estimating these parameters. Although SCAPE-APA does not necessarily improve prediction accuracy over the original SCAPE, it offers significant gains in computational efficiency. To help clarify these points, we have updated the legend to emphasize that the focus is on parameter consistency, and we refer readers to Supplementary Note 1 for a detailed explanation of the parameter meanings. This is the revised legend for Figure 1C. *"(C) A real example of a spurious pA site. PA1 is the spurious pA site and should be merged into PA2. The bottom panel displays 3'-UTR annotations from the GTF file, showing one single-segment 3'-UTR transcript and two multi-segment 3'-UTR transcripts. PA1 and PA2 are most likely derived from the first transcript, necessitating the merging of PA1 into PA2."* Minor concerns:

1. The manuscript states that "Polyadenylation sites are specific locations within the 3'-UTR." However, this is not entirely accurate. While pA is commonly associated with the 3'-UTR, it can also occur in other regions, such as introns, coding sequences, and intergenic regions. Please be cautious with your description to avoid similar issues.

Response: Thank you for your valuable feedback. We have revised the text to ensure accuracy and clarity: *"Polyadenylation sites (pA sites) are specific locations, primarily within the 3'-UTR but also found in introns, coding sequences, and intergenic regions, where mRNA undergoes polyadenylation, resulting in the addition of a poly(A) tail."*

1. Please add references to the second paragraph of the description.

Response: Thank you for your suggestion. We have added the following references to the second paragraph of the description in the revised manuscript. *Chen H, Wang Z, Gong L, et al.: A distinct class of pan-cancer susceptibility genes revealed by an alternative polyadenylation transcriptome-wide association study. Nat Commun. 2024;15:46064. 38409266 10.1038/s41467-024-46064-7 10897204 Mitschka S, Mayr C: Context-specific regulation and function of mRNA alternative polyadenylation. Nat Rev Mol Cell Biol. 2022;23(12):779–96. 35798852 10.1038/s41580-022-00507-5 9261900*

1. Please use "summarize as" instead of "summarize into" in the sentence: "*The full SCAPE analysis workflow can be summarized into the following steps* "

Response: Thank you for your suggestion. We have updated the text accordingly in the revised manuscript.

1. The sentence "*Each pA site is represented by a Gaussian component, which has three parameters: mean μ, standard deviation δ and weight π.*" is unclear. It would be helpful to specify what these parameters represent (e.g., do they relate to the distribution of fragment lengths or to the Gaussian mixture model itself?). Additionally, these parameters do not seem to correspond with those shown in Figure 1A. Furthermore, the figure legend lacks sufficient explanation and should be revised for clarity.

Response: We apologize for the confusion. The meanings of these parameters are explained in detail in *Supplementary Note 1*. To improve clarity, we have revised the text as follows: *" Each pA site is represented by a Gaussian component, which has three parameters: mean μ, standard deviation δ and weight π. Detailed meanings of these parameters and model specification are provided in Supplmentary Note 1. "* Additionally, we have revised the figure legend to provide a clearer explanation of the parameters and their relevance to the Gaussian mixture model. *"For both SCAPE-APA and SCAPE, the outputs include the estimated mean and standard deviation of pA sites (represented as \alpha and \beta in the bottom panel). In the SCAPE model, pA site locations are modeled as Gaussian components."*

1. The legend for Figure 1 lacks sufficient detail and does not fully explain the components of the figure.

Response: Due to technical reasons, *Supplementary Note 1* was not uploaded in the initial version. The SCAPE model is described in detail in the original publication and *Supplementary Note 1*. To enhance clarity, we have updated the figure legend as follows: *"(A) Original SCAPE model and improvements of SCAPE-APA. In data preprocessing, we developed a new script to extract 3'-UTR annotations from GTF files and implemented read binning. In modeling, we introduced a new likelihood function for binned reads and derived an updated inference algorithm, as detailed in Supplementary Note 1. In implementation, we rewrote SCAPE using Taichi to enhance computational efficiency. For both SCAPE-APA and SCAPE, the outputs include the estimated mean and standard deviation of pA sites (represented as \alpha and \beta in the bottom panel). In the SCAPE model, pA site locations are modeled as Gaussian components."*

1. In Figure B, please use gray to represent the spurious pA sites and orange to represent the real pA, and annotate it in the figure. For the isoforms in Figure C, please use the colors corresponding to those in Figure B.

Response: Thank you for your suggestion. We have added an annotation in *Figure 1B* to explicitly indicate that orange represents junction reads. To maintain consistency between *Figures 1B* and *1C*, we have used orange for spurious pA sites and gray for normal pA sites. *Figure 1C* is generated from real data using the *trackplot* package. Unfortunately, *trackplot* does not provide a function to customize the color of junction reads, making it technically

challenging to match the colors used in *Figure 1B*. However, we have ensured that the overall visual representation remains clear and informative. We have revised the legend of Figure 1B as follows:
"*(B) Schematic illustration of a spurious pA site due to splicing of 3'-UTR. The SCAPE model assumes that each 3'-UTR is a single continuous segment; however, in reality, 3'-UTRs can be multi-segmented. This results in reads that are close together on the transcript appearing far apart on the genome, violating the SCAPE model's assumption and potentially leading to spurious pA sites. In the figure, orange represents junction reads, while gray indicates normal reads.*"

1. Please add a description for the abbreviation. For example, std, standard deviation.

Response: Thank you for your suggestion. We have added descriptions for abbreviations in the figure legend in the revised manuscript. The updated text is as follows: *"(E–G) Comparison of the estimated mean, standard deviation (std), and weight of predicted pA sites. The curves represent the density contours of data points. The means of the Gaussian components indicate the predicted pA site locations. For a detailed explanation of these parameters, please refer to Supplementary Note 1."*

1. Supplementary file not available.

Response: We apologize for the missing supplementary file in the initial submission. *Supplementary Note 1* was uploaded, but due to technical reasons, it did not appear alongside the main text. We regret any inconvenience this may have caused during the review process. We have now uploaded the file to Zenodo: https://doi.org/10.5281/zenodo.14517303.

Competing Interests: No competing interests were disclosed.

Version 1

Reviewer Report 13 December 2024

https://doi.org/10.21956/openreseurope.20202.r47490

© 2024 Ye W. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Wenbin Ye
[1] University of California Irvine, Irvine, California, USA
[2] University of California Irvine, Irvine, California, USA
[3] University of California Irvine, Irvine, California, USA

In this manuscript, the authors introduce SCAPE-APA, a reimplementation of the previously developed SCAPE package for estimating alternative polyadenylation events from single-cell RNA-seq (scRNA-seq) data. The new version addresses performance issues in handling large datasets

by introducing several key improvements, including read binning, a re-derived mixture model, and the use of Taichi for algorithm acceleration. Additionally, the authors improved the UTR annotation extraction and added a script for detecting spurious polyadenylation sites. Overall, these updates significantly enhance the speed and scalability of SCAPE-APA, making it a valuable tool for large-scale scRNA-seq analyses. However, I have several concerns about this manuscript.

Major concerns:

1. Although the authors claim that SCAPE-APA can address the challenges of processing large datasets, the adoption of the SCAPE tool remains limited. Additionally, users can mitigate performance issues by employing alternative strategies, such as splitting chromosomes and implementing multithreading, to handle large datasets more efficiently.
2. This manuscript does not fully demonstrate the improved performance of SCAPE-APA in handling large datasets, nor does it provide clear evidence that its performance in predicting pA sites is comparable to or improved over SCAPE. The authors only use 100, 500, 1000, and 2000 3'UTRs to illustrate performance improvements, which may not be convincing given the scale of real-world datasets.
3. This article proposes an improvement to the SCAPE model (SCAPE-APA), but the introduction only briefly describes the SCAPE method and does not mention its limitations. Please add a section discussing these issues to provide background for SCAPE-APA.
4. For Figure 1A, the SCAPE diagram is unclear and too small, making it difficult to interpret. Additionally, the diagram does not fully reflect the proposed improvements. Specifically, the peripheral gaze does not align with the steps in the SCAPE model, and the cycle diagram suggests an infinite loop. Please integrate the improvements directly into the original SCAPE diagram rather than describing them only in text.
5. Figure C is confusing. Why do the pA sites (e.g., PA1, PA2, PA3) not correspond to the ends of the isoforms? What is the relationship between the three isoforms and the pA sites (PA1, PA2, PA3)? Furthermore, this figure doesn't clearly explain the rationale for merging PA1 into PA2.
6. More importantly, the authors should give an overall statistical result rather than just showing one example, showing how many spurious pA sites can be detected using SCAPE-APA compared with SCAPE
7. The manuscript states that "The biggest challenge in real data analysis is that the read number can be very large (n > 200,000) for some 3'-UTRs". However, several points need to be addressed: 1) Are all the selected 3'UTR deeply covered (Fig. 1D)? 2) the number of 3'UTR (100, 500, 1000, 2000) tested is relatively small. For example, there are estimated to be around 20,000 3'UTR in the human genome. 3) Although the improved method increases speed, processing 2000 3'UTRs in 21 hours is still unacceptable, especially for the human genome.
8. Figures 1E-G show consistent results between the old and new packages using the mean, standard deviation, and weight parameters. However, including polyA site prediction accuracy would give users a clearer understanding of SCAPE-APA's performance and how it improves upon SCAPE.

Minor concerns:

1. The manuscript states that "Polyadenylation sites are specific locations within the 3'-UTR." However, this is not entirely accurate. While pA is commonly associated with the 3'-UTR, it can also occur in other regions, such as introns, coding sequences, and intergenic regions. Please be cautious with your description to avoid similar issues.
2. Please add references to the second paragraph of the description.

3. Please use "summarize as" instead of "summarize into" in the sentence: "*The full SCAPE analysis workflow can be summarized into the following steps* "
4. The sentence "*Each pA site is represented by a Gaussian component, which has three parameters: mean μ, standard deviation δ and weight π.*" is unclear. It would be helpful to specify what these parameters represent (e.g., do they relate to the distribution of fragment lengths or to the Gaussian mixture model itself?). Additionally, these parameters do not seem to correspond with those shown in Figure 1A. Furthermore, the figure legend lacks sufficient explanation and should be revised for clarity.
5. The legend for Figure 1 lacks sufficient detail and does not fully explain the components of the figure.
6. In Figure B, please use gray to represent the spurious pA sites and orange to represent the real pA, and annotate it in the figure. For the isoforms in Figure C, please use the colors corresponding to those in Figure B.
7. Please add a description for the abbreviation. For example, std, standard deviation.
8. Supplementary file not available.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: alternative polyadenylation; computational biology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.