

Technical Note

Not peer-reviewed version

SeqMate: A Novel Large Language Model Pipeline for Automating RNA Sequencing

[Devam Mondal](#)* and Atharva Inamdar

Posted Date: 2 July 2024

doi: 10.20944/preprints202407.0160.v1

Keywords: bioinformatics; natural language processing; automation



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Technical Note

SeqMate: A Novel Large Language Model Pipeline for Automating RNA Sequencing

Devam Mondal * and Atharva Inamdar

Center for Complex Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ; atharva.inamdar006@gmail.com

* Correspondence: dmondal@stevens.edu

Abstract: RNA sequencing techniques, like bulk RNA-seq and Single Cell (sc) RNA-seq, are critical tools for the biologist looking to analyze the genetic activity/transcriptome of a tissue or cell during an experimental procedure. Platforms like Illumina's next-generation sequencing (NGS) are used to produce the raw data for this experimental procedure. This raw FASTQ data must then be prepared via a complex series of data manipulations by bioinformaticians. This process currently takes place on an unwieldy textual user interface like a terminal/command line that requires the user to install and import multiple program packages, preventing the untrained biologist from initiating data analysis. Open-source platforms like Galaxy have produced a more user-friendly pipeline, yet the visual interface remains cluttered and highly technical, remaining uninviting for the natural scientist. To address this, SeqMate is a user-friendly tool that allows for one-click analytics by utilizing the power of a large language model (LLM) to automate both data preparation and analysis (differential expression, trajectory analysis, etc). Furthermore, by utilizing the power of generative AI, SeqMate is also capable of analyzing such findings and producing written reports of upregulated/downregulated/user-prompted genes with sources cited from known repositories like PubMed, PDB, and Uniprot.

Keywords: bioinformatics; natural language processing; automation

1. Introduction

Processing RNA-seq data is tedious. Converting the FASTQ file output of bulk-seq or chip-seq involves numerous file conversions and data processing involving numerous packages in a process that remains unautomated and inaccessible to the biologist untrained in bioinformatics. Many advances in process automation, especially utilizing the technological boom in generative AI and large language models, have not been implemented in the bioinformatics space. SeqMate is looking to change that by utilizing an accountable (cites sources and actively prompted against hallucinations) LLM that automates bioinformatic analysis like trajectory analysis, differential expression, and more. Currently, the tool is designed and tested for differential expression starting with raw FASTQ data, but SeqMate is envisioned as a suite of multiple bioinformatic processes that provides a one-click option for the untrained biologist to spearhead data analysis.

2. RNA Sequencing: A Brief Biological Context

RNA sequencing involves high-throughput sequencing of RNA nucleic acid sequences and amounts. RNA-seq has led to many scientific advances, such as identifying regulatory regions, biomarkers, and mutations [1]. Within the family of RNA sequencing exist two distinct processes, bulk RNA sequencing and single-cell RNA sequencing. Bulk RNA sequencing allows a researcher to look at the transcriptome of a sample/tissue, commonly by using NGS technology like the Illumina sequencing platform. Single-cell seq differs in that instead of looking at the transcriptome of an entire tissue, one can analyze the transcriptome of an individual cell by using microfluidics to separate individual cells and barcoding the RNA of each cell with unique identifiers, allowing researchers to identify the genetic activity of various cell sub-populations within a sample [2]. Both these technologies allow researchers to take snapshots of the genetic activity of a tissue/cell and are powerful tools to determine changes in expression levels of genes after any experimental treatment. Coupling multiple

scRNA-seqs/bulk-seqs over time allows a powerful time series analysis of the expression levels of an experimental sample [1].

3. Role of Bioinformatics in RNA Sequencing

RNA-seq is a powerful high throughput sequencing tool that can produce large amounts of data, yet there is still a large amount of tedious data processing that remains before meaningful insights can be gleaned. To delineate, the traditional bioinformatics process from raw RNA-seq data looks like the following:

1. Perform quality control.
2. Cut adapter regions and align with reference genome.
3. Perform file conversion and produce count matrices.
4. Normalize counts.
5. Perform differential expression analysis.
6. Relate upregulation/downregulation findings with relevant biological pathways.
7. Create write-ups/visuals.

After such analysis, one can meaningfully interpret and extrapolate the results of RNA-seq to their experimental design [3].

4. Current Limitations of Bioinformatics Suites

Currently, bioinformatics analysis involves knowledgeable professionals running various open-source packages on some form of textual user interface (command line/terminal). Conducting such analysis involves importing/downloading the correct packages, uploading the data, running the correct commands, and downloading the processed data to the right location for every step of a multi-step process that takes place in a lay-person unfriendly environment. This cryptic process obfuscates natural scientists like cellular biologists from conducting their own analysis and bottlenecks the scientific process. Open-source platform Galaxy has made a bioinformatics analysis pipeline, but its usability is limited by a confusing visual interface. While Galaxy has created a powerful tool that streamlines bioinformatics for the trained specialist, it is still too complex and technical to be quickly intuitive for the biologist. Additionally, the pipelines Galaxy provides are generic and do not cater to specific use cases, instead serving as general guides [4].

5. Our Approach

In our goal to automate the process of RNA Sequencing, we aimed to create an autonomous system that was capable of receiving user input, running the appropriate intermediate steps (generating genome indices, creating appropriate metadata for differential expression analysis, filtering genome indices, etc.), and providing an output. For this particular use case, we decided that the user input would be a series of FASTQ files, the appropriate genome, thresholds for log fold change and p-value, specification of control and experimental FASTQ files, and the output would be a series of genes demonstrating greatest differential expression based on user-defined thresholds (log fold change and p-value) and detailed information about them.

Such a system would require a "logical driver" that conducts decision-making (picking the appropriate files either from the user or from generated files in other steps of the intermediate process, running the appropriate code/terminal commands in an environment, generating graphs, providing narration, etc.). We chose to use a large language model to be this component of our system due to their ability to understand data through insight generation.

However, vanilla LLMs have two main challenges. First, they are worse at solving complex reasoning problems that require decision making at intermediate steps compared to simple, single-step tasks [5]. Additionally, they lack additional functionalities necessary within this system, such as the ability to run open-source libraries on native environments as well as search online databases/browsers (such as the NCBI GenBank Database and Ensembl). Therefore, we couple the large language model

with a variety of external tools that enable the aforementioned tasks, thus creating an **LLM agent**. Furthermore, we make this agent using the open-source package LangChain. LangChain enables us to improve an LLM's ability to solve complex, reasoning problems by forcing the LLM to use chain-of-thought reasoning (where it assesses the given prompt/query, breaks it down into intermediate steps that are rationale, and executes them using a variant of the ReAct paradigm) [6–8]. Figure 1 demonstrates this coupling.

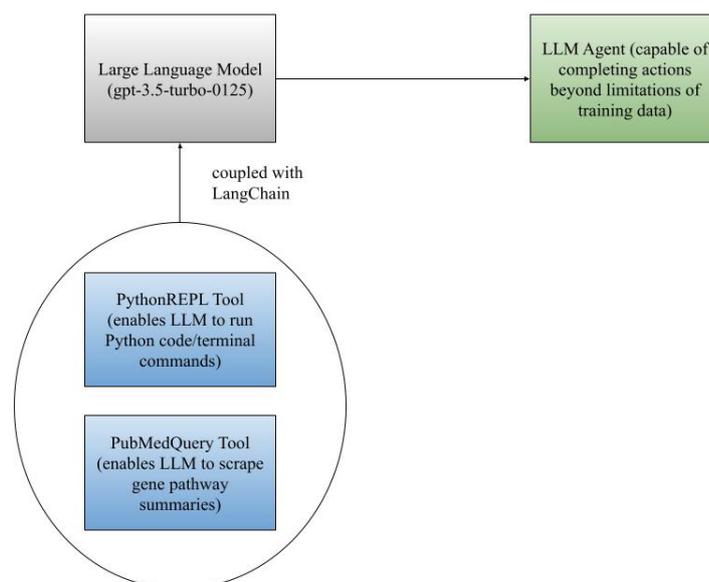


Figure 1. Visualization of LLM agent creation.

With this agent, a combination of an LLM and external tools, we automate each intermediate step through prompt engineering that outlines what files it must use, what package is appropriate, and what files it must output. We also introduce few-shot and one-shot prompting with code snippets in order to increase the success rate of each prompt evocation. Additionally, the LLM's decision-making abilities increase when integrated with LangChain by placing emphasis on chain-of-thought reasoning when approaching each prompt. With these prompts, the agent is able to follow the following pipeline without user interaction:

1. Open the user-provided FASTQ files.
2. Remove low-quality regions and adapters using `cutadapt`.
3. Provide quality control statistics for edited FASTQ files using `bio`, generating graphics and providing narration.
4. Generate a genome index for a user-provided genome using `hisat`.
5. Align edited FASTQ files to the aforementioned genome index using `hisat`, producing a SAM output file.
6. Convert the SAM output file to a BAM file using `pysam`.
7. Download the genome annotation file for the genome.
8. Create a count matrix using the BAM files and the genome annotation file using `featureCounts`.
9. Edit the counts table using appropriate matrix operations.
10. Create metadata needed for differential expression analysis
11. Run differential expression analysis using the count matrix and metadata using `pydeseq2`.
12. Filter outlier genes based on user thresholds mentioned earlier.
13. Generate insights on outlier genes using external databases and provide a detailed, cohesive summary with citations.

It is important to note that the agent utilizes the source machine's environment in order to run the open-source packages. It is thus able to utilize packages present in pip and conda environments that are set up on the source machine, make alterations through installations, and switch to other environments.

6. Limitations and Future Work

Due to intrinsic noise that exists in training data, LLMs are susceptible to hallucinations where they produce factually incorrect results. Though our prompt engineering limits this through one-shot/zero-shot prompts, the pipeline sometimes generates inaccurate results. We hope to generate statistics for hallucinations (frequency, type, etc.) in a follow-up report.

Additionally, generating a genome index through hisat is currently computationally expensive. However, this should not be an issue when the pipeline is run on a desktop or server.

Moreover, for the LLM in the system, we currently rely on OpenAI's gpt-3.5-turbo-0125 model. This may pose a privacy concern, as aspects of the data supplied to the agent are sent to OpenAI through an API. In the future, we would like to experiment with open-source LLMs that can be ran locally.

It is also important to create a graphical user interface for users to interact with the LLM agent, and we also plan on extrapolating our pipeline to other bioinformatics processes.

7. Conclusion

In summary, SeqMate is an automated pipeline that simplifies RNA-seq data processing and analysis built for the natural scientist. Unlike traditional "pipelines" existing bioinformatics suites provide, SeqMate is custom to every use-case by leveraging a large language model agent. In the future, SeqMate looks to expand the pipeline to include other analytical methods following the same user-friendly design policy.

References

1. Deshpande, D.; Chhugani, K.; Chang, Y.; Karlsberg, A.; Loeffler, C.; Zhang, J.; Muszyńska, A.; Munteanu, V.; Yang, H.; Rotman, J.; Tao, L.; Balliu, B.; Tseng, E.; Eskin, E.; Zhao, F.; Mohammadi, P.; P. Łabaj, P.; Mangul, S. RNA-seq data science: From raw data to effective interpretation. *Frontiers in Genetics* **2023**, *14*. doi:10.3389/fgene.2023.997383.
2. Li, X.; Wang, C.Y. From bulk, single-cell to spatial RNA sequencing. *International Journal of Oral Science* **2021**, *13*. doi:10.1038/s41368-021-00146-0.
3. Wu, C.; Bendriem, R.M.; Garamszegi, S.P.; Song, L.; Lee, C. <scp>RNA</scp> sequencing in post-mortem human brains of neuropsychiatric disorders. *Psychiatry and Clinical Neurosciences* **2017**, *71*, 663–672. doi:10.1111/pcn.12550.
4. Afgan, E.; Baker, D.; Batut, B.; van den Beek, M.; Bouvier, D.; Čech, M.; Chilton, J.; Clements, D.; Coraor, N.; Grüning, B.A.; Guerler, A.; Hillman-Jackson, J.; Hiltemann, S.; Jalili, V.; Rasche, H.; Soranzo, N.; Goecks, J.; Taylor, J.; Nekrutenko, A.; Blankenberg, D. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research* **2018**, *46*, W537–W544. doi:10.1093/nar/gky379.
5. Mondorf, P.; Plank, B. Beyond Accuracy: Evaluating the Reasoning Behavior of Large Language Models – A Survey, 2024, [arXiv:cs.CL/2404.01869].
6. Huang, J.; Chang, K.C.C. Towards Reasoning in Large Language Models: A Survey. Findings of the Association for Computational Linguistics: ACL 2023; Rogers, A.; Boyd-Graber, J.; Okazaki, N., Eds.; Association for Computational Linguistics: Toronto, Canada, 2023; pp. 1049–1065. doi:10.18653/v1/2023.findings-acl.67.
7. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2023, [arXiv:cs.CL/2201.11903].
8. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models, 2023, [arXiv:cs.CL/2210.03629].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.