Cray Minor (A15863773, csminor@ucsd.edu)
Adv. Machine Learning Methods (COGS 185, SP23)
Prof. Zhuowen Tu
June 13th, 2023

RNA Sequencing Pipeline: From FASTQ to DEA

https://github.com/crayminor/rna_seq

Abstract:

The main objective of this project is to develop a user-friendly pipeline that can be easily utilized by researchers in our laboratory, even those with limited to no command line experience. By providing a general framework and guidelines, it allows local researchers to conduct RNA-sequencing analysis on their own, enabling them to further explore gene expression data. In the following sections, a comprehensive introduction to RNA sequencing, underlying principles, and its wide-ranging applications is provided. It is important to emphasize the uniqueness of the dataset, and its size (~50 gb of raw reads in fastq.gz reads) as well as its relevance to our research goals. Overall, this project aims to understand expression patterns in the experimental data to unveil novel findings as well as the potential to make a significant impact by providing a valuable tool for researchers in our laboratory. By simplifying the analysis process, we strive to facilitate further discoveries, quicken the process, and make further advancements in the immunological research domain.

Introduction:

RNA sequencing (RNA-seq) has revolutionized the field of genomics by enabling comprehensive analysis of gene expression patterns. It can provide valuable insights into the transcriptome and a powerful tool for studying gene regulation, identifying differentially expressed genes, and exploring biomarkers. In this study, we focused on analyzing a large RNA-seq dataset to investigate the transcriptional dynamics of wild type and transgenic biological systems fed on a High-Fat Diet (HFD) inducing liver stress. The uniqueness of our dataset lies in its

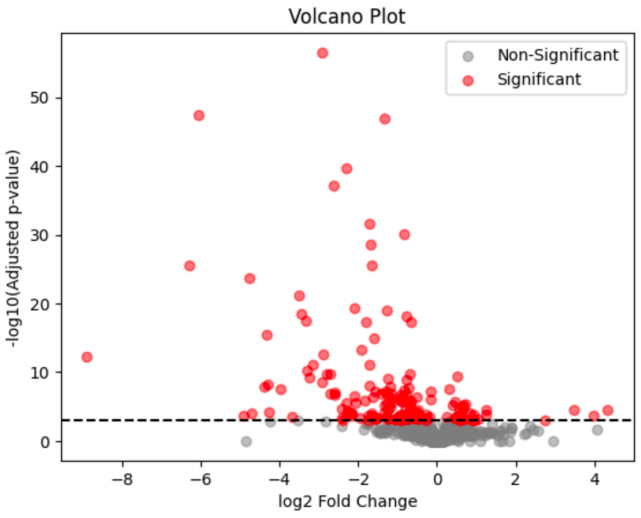

Figure 1: Volcano plot using log2 fold change and adjusted p-val, to visualize differentially expressed genes across conditions.

experimental nature, which allows us to gain a deeper understanding of the regulatory mechanisms underlying gene expression changes in relation to our overall project, examining genes involved in fibrotic tissue build up in Non-alcoholic fatty liver disease. Specifically, we aimed to examine the influence of a transcription factor on its downstream targets, unraveling the intricate network of genes involved in the build up fibrotic tissue in the liver. Analyzing large RNA-sequence datasets presents both challenges and opportunities. One major challenge is the time-consuming nature of processing and manipulating the raw data. For instance, samtools, used for SAM/BAM file manipulation, require substantial computational resources and was particularly time-consuming, given the number of samples per experimental group. However, these challenges are outweighed by the opportunities for uncovering valuable biological insights from these vast datasets. Furthermore, working with large RNA-seq datasets provides an opportunity to leverage advanced computational methods and statistical analyses. The integration of specialized packages, such as PYDeseq2, allows for comprehensive differential expression analysis and the identification of genes exhibiting significant changes across experimental conditions. One of the primary motivations for this study is to shed light on the regulatory mechanisms governed by the transcription factor of interest and its downstream targets to understand the transcriptional network in the development, disease progression, and response to environmental stimuli. In summary, this study aims to leverage the power of RNA-seq technology and advanced analysis tools to investigate the transcriptional dynamics. After careful and time-consuming challenges associated with this RNA-seq dataset, we aim to further unravel the regulatory mechanisms of a transcription factor and shed light on its downstream targets. In addition, this repo will be further built out, to expand the toolset and facilitate further biomedical discovery.

Methods:

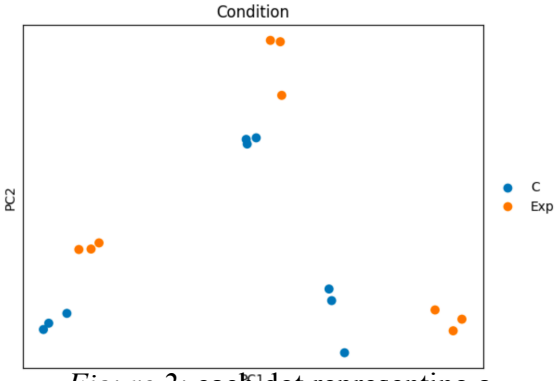

Figure 2: each dot representing a sample, using 3 principal components for dimensionality reduction.

Data collection and preprocessing included obtaining raw fastq files from RNA sequencing experiments conducted on a 9 wild-type and 9 transgenic mice. The raw fastq files opened and processed using fastp for comprehensive quality control and preprocessing . It involved tasks such as adapter trimming, quality filtering, read trimming, and read length filtering. This ensured that the input fastq files are high quality and suitable for downstream analysis. The processed reads were aligned to the mouse reference genome downloaded from the hisat2 page. Then the aligned the reads were mapped to the indexed genome. The output was a SAM file containing the aligned reads, samtools, was used for manipulating SAM/BAM files, and was utilized in multiple conversion steps. Including sorting, indexing, filtering, merging, and converting SAM/BAM files. Samtools facilitated the post-processing of aligned reads and further

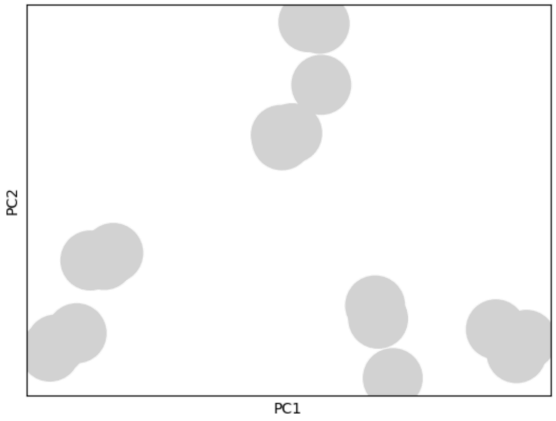

Figure 3: PCA plot blobs.

preparing them for downstream analysis. Read counting and summarization was done with the aligned reads in BAM format and an annotated GTF file and was processed using the featureCounts tool from the subread package. FeatureCounts assigned each read to the corresponding genomic feature, genes/exons, and number of reads per feature were counted and appended to a .txt file. The output was a counts matrix where rows represented the Ensmbl genetic id and columns represented the sample condition and number. For PCA with sklearn, the counts were normalized using transcripts per million method, making use of available gene length and normalizing counts. The counts matrix and sample metadata were used as inputs for differential expression analysis using the pyDeseq2 package, a re-implementation of the original DESeq package in R. The DeseqDataSet object was created, incorporating the counts and metadata, with the design factors were specified. Analysis of counts involved fitting size factors, estimating genewise dispersions, fitting dispersion trend, fitting dispersion prior, and calculating differential expression statistics. Figure 1, was created using a volcano function to find the most differentially expressed genes across experimental conditions and find the transgenic mice have significantly more expression of downstream genes. The volcano was based on the processed data using log2 fold change, and adjusted p-value. To visualize the principal component analysis (PCA), we performed a dimensional reduction

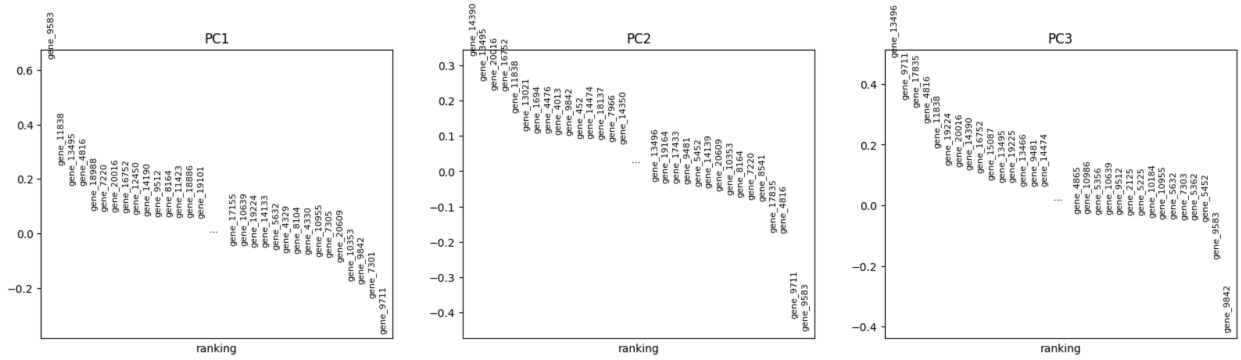

Figure 4: Top 3 principal components and genes in oder of most to least contribution to the principal components.

technique to explore the variation in gene expression across samples. The statistical significance of differential expression was determined using the Wald test, and p-values were adjusted with multiple testings using an appropriate method, such as Cookes outlier detection. Genes with adjusted p-values below a specific threshold 0.01 were considered differentially expressed. Finally, using tensorflow's keras packages I implemented an auto-encoder network to look gene

expression patterns to correctly determine the control (0) or experimental (1) condition mice. Due to the low number of samples to predict the model was able to achieve complete 1.0 accuracy on the test set. Overall, this RNA sequencing project involved preprocessing the raw fastq files, aligning the reads to the reference genome, performing read counting and summarization, conducting differential expression analysis using pydeseq2, and visualizing the results through PCA and volcano plots. These methods allowed for the identification of genes exhibiting significant differential expression between the studied conditions, while maintaining simplicity of syntax.

Experiments:

The experimental design employed in this study was carefully crafted to address our current research objectives and ensure the validity of the findings. We followed a systematic workflow that encompassed several key steps, including data preprocessing, differential expression analysis, and machine learning algorithms. Rigorous experimental design for the collection of the data and parameter tuning were integral to obtaining reliable results. To begin, the raw RNA sequencing data was subjected to quality control and preprocessing steps using fastp, samtools, hisat2, and subread tools in command line. These tools helped remove low-quality reads, mapping, conversion and removal of excess artifacts, ensuring the integrity and accuracy of the downstream analysis. The converted data was then loaded into a JupyterNB where we performed differential expression analysis using the pyDeseq2 package. This analysis allowed us to identify genes that exhibited significant changes in expression levels between different conditions. Statistical tests, such as the negative binomial test, were employed to assess the significance of differential expression, and appropriate adjustments for testing. After breaking down the principal components used for dimensionality reduction we found that the first 3 principal components

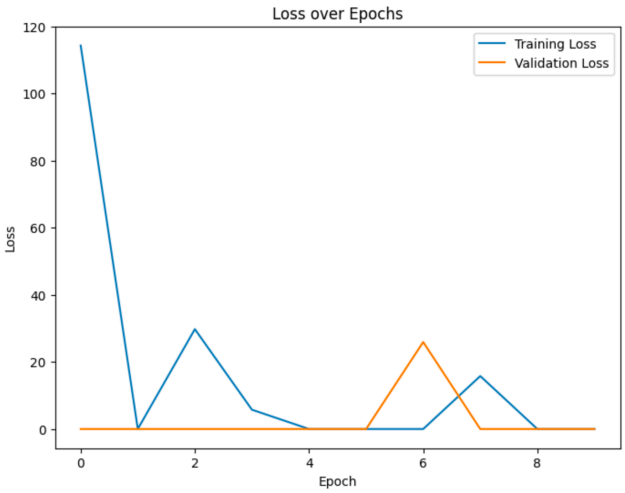

Figure 5: Auto-encoder network results, loss over 10 epochs.

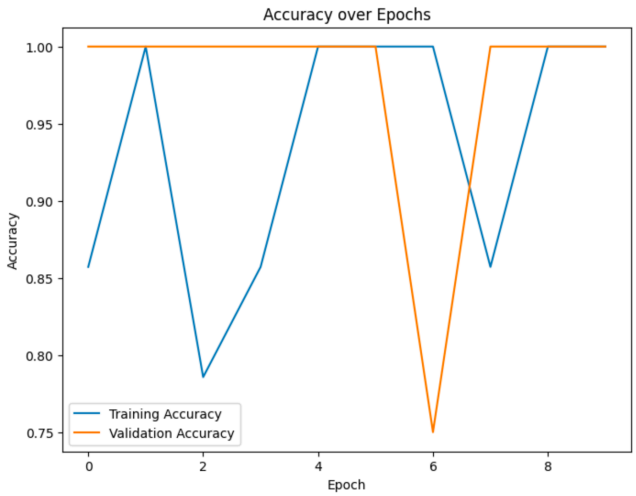

Figure 6: Auto-encoder network, accuracy of model at predicting experimental condition from expression patterns

were important, and were able to identify the genes highly involved in this dimensionality reduction algorithm. To gain insights into the overall structure and patterns within the dataset, we examined the list of genes highly associated. PCA enabled us to visualize the most variable relationships among samples based on their genetic expression profiles. This technique provided a multidimensional representation of the data, allowing for identification of potential clusters based on gene expression patterns. Furthermore, an auto-encoder algorithm was employed to explore predictive modeling and classification. We aimed to build a simple pre-trained model that could classify sample condition based on gene expression data. Throughout the experimental process and examining the data, we played with hyper-params of the algorithms to optimize performance and generalization capabilities. Overall, the experimental design encompassed a comprehensive approach to analyze the RNA sequencing data and extract meaningful insights, while maintaining a simple, comprehensive pipeline. The combination of differential expression analysis with pyDESeq2, PCA, and auto-encoder network allowed us to explore the underlying patterns, identify potential biomarkers, and gain a deeper understanding of the biological processes involved in non-alcoholic fatty liver disease (NAFLD). This notebook helped us identify genes up-regulated in fibrotic build up. The rigorous experimentation and careful parameter tuning ensured the reliability and reproducibility of the results. In summary, the experimental section of this study demonstrates the systematic and comprehensive approach employed to convert and analyze the RNA sequencing data. These results serve as a foundation for further investigations in liver-related disease.

Conclusion:

In this study, we conducted comprehensive analysis on RNA sequencing data to investigate the effects of activating a transcription factor in transgenic mice on a HFD (to induce NAFLD). The experimental design encompassed various steps, including data collection and preprocessing through command line tools, differential expression analysis in python, and tensor flow transfer learning algorithms. By leveraging advanced computational-biology packages, we aimed to uncover novel insights and further application in the field of immunology. The results of the differential expression analysis revealed significant changes in gene expression profiles between wild type and transgenic mouse samples. Several genes exhibited differentially expressed patterns, suggesting their involvement in the build-up of fibrotic tissue. These findings contribute to our understanding of the underlying molecular mechanisms and potential biomarkers related to NAFLD. Validation and quality of our analysis, included employing visualizations such as PCA plots, volcano, and accuracy/loss curves. These plots demonstrated clear separation between the conditions, indicating the robustness of our data and the effectiveness of the analytical pipeline. Additionally, further work can be done to produce more plots, including heatmaps or clustering analyses for genome set enrichment analysis, to better identify gene expression patterns and potential regulatory pathways. While this study presents valuable insights into liver disease and cancer, it is important to acknowledge limitations. First and foremost was the time the data conversion took, limiting time spent on expression analysis in python. Another limitation is the reliance on the available data, future studies should aim to

expand the dataset and incorporate more diverse samples to enhance the generalizability of the findings. Furthermore, additional machine learning algorithms can be applied in this study to represent a subset of available methods to explore additional approaches for further insights and improve prediction accuracy. In conclusion, this research project demonstrates the potential impact of RNA sequencing data analysis and the ability to generalize this pipeline to large transcriptome reads. The combination of the methods and experiments perform has provided valuable insights into the molecular mechanisms underlying regulation of fibrosis in the liver. Moving forward, it is crucial to continue exploring novel methods, expanding datasets, and collaborating with multidisciplinary teams to advance our understanding and potential therapeutic interventions in NAFLD, as well as accelerate scientific advancement within the laboratory.

Privacy & Ethics:

For privacy reasons, the data will not be publicly listed until future publication. Data was collected from local laboratory samples and part of a larger study. This notebook is publicly available on GitHub to provide a pipeline that can be adapted for other FASTQ reads.

Sources:

[1] TPM function calculation: https://github.com/lucynwosu/TPM-Transcripts-Per-Million-Normalization-Python/blob/main/TPM-Transcripts-Per-Million-Normalization.ipynb
pydeseq2 syntax help: https://github.com/mousepixels/sanbomics_scripts/blob/main/PyDeseq2_DE_tutorial.ipynb

[2] fastp: Shifu Chen. 2023. Ultrafast one-pass FASTQ data preprocessing, quality control, and deduplication using fastp. iMeta 2: e107. https://doi.org/10.1002/imt2.107; 2) Shifu Chen, Yanqing Zhou, Yaru Chen, Jia Gu; fastp: an ultra-fast all-in-one FASTQ preprocessor, Bioinformatics, Volume 34, Issue 17, 1 September 2018, Pages i884–i890, https://doi.org/10.1093/bioinformatics/bty560

[3] samtools: Twelve years of SAMtools and BCFtools, Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, Heng Li (GigaScience, Volume 10, Issue 2, February 2021, giab008) https://doi.org/10.1093/gigascience/giab008

[4] hisat2: https://daehwankimlab.github.io/hisat2/
[5] subread: Liao, Y., Smyth, G. K., & Shi, W. (2013). The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. Nucleic acids research, 41(10), e108. https://doi.org/10.1093/nar/gkt214

[6] pydeseq2: @article{muzellec2022pydeseq2, title={PyDESeq2: a python package for bulk RNA-seq differential expression analysis}, author={Muzellec, Boris and Telenczuk, Maria and Cabeli, Vincent and Andreux, Mathieu}, year={2022}, doi = {10.1101/2022.12.14.520412}, journal={bioRxiv}, }

[7] sanbomics (gene mapping) + pydeseq2 tutorial: https://github.com/mousepixels/sanbomics_scripts/blob/main/PyDeseq2_DE_tutorial.ipynb

[8] tf keras auto-encoder: Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org

[9] Code troubleshooting: StackOverflow & OpenAI. (2023). ChatGPT 3.5 [Large language model]. https://chat.openai.com