Introduction

The study used for this Python data analysis project is: *Fascin actin-bundling protein 1 regulates non-small cell lung cancer progression by influencing the transcription and splicing of tumorigenesis-related genes*[1]. In the past decade, lung cancer remains the leading cause of cancer-related deaths worldwide, with a continually increasing incidence. The largest subtype of lung cancer, non-small cell lung cancer (NSCLC), though approached with continuous advancements in therapy, remains a challenge for patients to recover fully. As such, there is an urgent need for advancements in early detection diagnosis and treatment strategies. Ribosome-binding proteins (RBPs) are protein-coding genes known to regulate post-transcriptional gene expression[2]. Recently, multitudes of studies have demonstrated that RBPs can modulate different types of tumour progression by mediating the expression and function of oncogenes and tumour suppressor genes through RNA modifications[3]. Specifically, the overexpression of fascin-actin binding protein 1 (FSCN1) is associated with tumour malignant phenotypes, and in a recent esophageal cancer study, FSCN1 has shown function characteristics to RBPs[4]. However, the molecular mechanisms that FSCN1 plays as an RBP in NSCLC remain unexplored. By performing bulk RNA-seq on a human NSCLC cell line, they aimed to identify differentially expressed genes and alternatively spliced genes following FSCN1 knockdown. There are many different types of data used in this study. However, the core of the study was the RNA-seq analysis that provided a genome-wide view of gene expression following FSCN1 knockdown. Total RNA was extracted from A549 cells transfected with siFSCN1 (small interfering RNA to knock down FSCN1 expression), and Illumina NovaSeq was used to generate the reads for bioinformatics analysis. After filtering the reads, they then used it to count gene reads and fragments per kb of transcript per million fragments mapped (FPKM) values and further performed downstream differential expression analysis.

References

1. Sun Q, Liu R, Zhang H, Zong L, Jing X, Ma L, Li J, Zhang L. 2023. Fascin actin-bundling protein 1 regulates non-small cell lung cancer progression by influencing the transcription and splicing of tumorigenesis-related genes. PeerJ. 11:e16526. doi:https://doi.org/10.7717/peerj.16526. [accessed 2024 Jan 4]. https://pubmed.ncbi.nlm.nih.gov/38077434/.
2. Neelamraju Y, Hashemikhabir S, Janga SC. 2015. The human RBPome: From genes and proteins to human disease. Journal of Proteomics. 127:61–70. doi:https://doi.org/10.1016/j.jprot.2015.04.031.
3. Li Z, Shi J, Zhang N, Zheng X, Jin Y, Wen S, Hu W, Wu Y, Gao W. 2022. FSCN1 acts as a promising therapeutic target in the blockade of tumor cell motility: a review of its function, mechanism, and clinical significance. Journal of Cancer. 13(8):2528–2539. doi:https://doi.org/10.7150/jca.67977. https://www.jcancer.org/v13p2528.htm.
4. Cai H, Wang R, Tang Z, Lu T, Cui Y. 2022. FSCN1 Promotes Esophageal Carcinoma Progression Through Downregulating PTK6 via its RNA-Binding Protein Effect. Frontiers in Pharmacology. 13. doi:https://doi.org/10.3389/fphar.2022.868296.

Methods

The data analysis for this project focused on creating a Principal Component Analysis (PCA) plot and a volcano plot, corresponding to Figures 3B and 3C in the paper. The analysis utilized RNA-seq data from the GEO database (accession number GSE234859), which included pre-processed FPKM values and gene read counts.

PCA Plot

In the PCA analysis, the process began with data import and preprocessing, where the FPKM matrix was read from a text file using pandas. Next, a metadata dataframe was created to store sample information, crucial for grouping and colouring in the subsequent visualization (also for running differential expression analysis). The core of the PCA computation utilized sklearn's PCA function. In addition, the data was standardized using StandardScaler to ensure that all features contributed equally to the analysis before computing PCA. After computing PCA, the results were transformed into a pandas dataFrame for easier manipulation. The visualization of PC1 versus PC2 consisted of three components: a scatter plot for individual samples, group means, and ellipses representing group distributions. The groups were distinguished by condition, as referenced in the metadata. Constructing the ellipses involved several computational steps, including calculating the covariance matrix, eigenvalues, and eigenvectors. Lastly, the angle, width, and height of the ellipses were derived from these computations and used to generate the visualization in Matplotlib.

Volcano Plot/Differential Expression Analysis

Differential expression analysis was performed using DESeq2 and the results were used to generate the volcano plot. While DESeq2 is an R package, the analysis was conducted in Python by leveraging the rpy2 module, enabling interoperability between Python and R. First, the gene count data was read into Python as a pandas dataframe and converted to an R dataframe using the pandas2ri module (as well as metadata). A DESeq2 object was then created using the count data and a metadata dataframe previously created. The differential expression analysis was executed using DESeq2 in the R environment. The results were then converted back to a pandas dataframe, retaining only the columns for adjusted p-values (padj) and log2 fold change (log2FoldChange). To identify significantly upregulated and downregulated genes, an additional column was created based on the significance thresholds defined in the reference study (|log2FoldChange| > 2 and padj < 0.05). Finally, the volcano plot was generated by plotting log2FoldChange against the negative log of padj. To replicate the original figure, the volcano plot also highlighted the number of upregulated, downregulated, and other genes.

For the complete analysis and plotting code, please refer to this repository:
https://github.com/Nathan2400912/Lung_Cancer_RNASeq_Analysis.git

Results

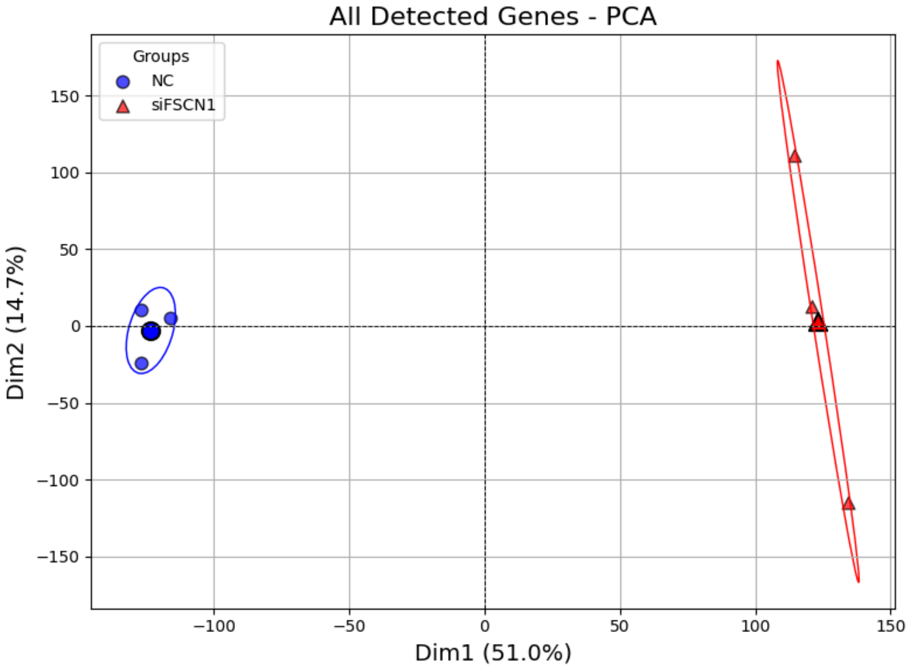

Figure 1. PCA based on FPKM values of FSCN1 gene knockdowns (siFSCN1) and control (NC) A549 cell samples.

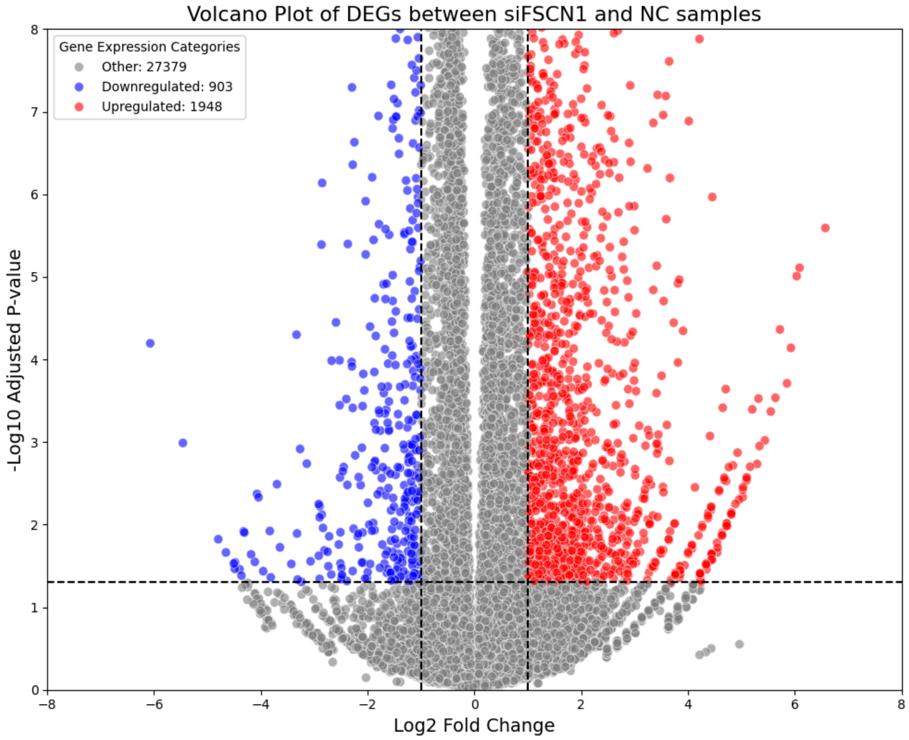

Figure 2. Volcano Plot of Differentially Expressed Genes Between siFSCN1 and NC Samples.

Both figures look extremely similar to the ones published in the study. Figure 1 displays the separation of samples based on gene expression profiles in two principal components (Dim1 and Dim2). The siFSCN1 samples (red triangles) and NC samples (blue circles) form distinct clusters, with Dim1 (accounting for 51.0% of the variance) capturing most of the separation between the groups. This clear separation indicates that the knockdown of FSCN1 (siFSCN1) significantly impacts gene expression profiles, differentiating them from the control (NC). The tight clustering within each group also suggests reproducibility of the samples, as the replicates show minimal variance within their respective groups. Figure 2 reveals the differential expression of genes between siFSCN1 and NC samples. A total of 903 genes are significantly downregulated (log2FoldChange < -1 and padj < 0.05), represented by blue points, while 1,948 genes are significantly upregulated (log2FoldChange > 1 and padj < 0.05), represented by red points. The majority of genes (27,379) fall into the "Other" category, as they do not meet the threshold for significant differential expression. The plot confirms that FSCN1 knockdown induces widespread changes in gene expression, with many genes showing significant alterations, further emphasizing the impact of FSCN1 in NSCLC.

Discussion

In reflecting on this project, I reached several conclusions regarding the reproducibility and transparency of scientific studies, particularly concerning data analysis and visualization. First of all, a significant challenge arose from the disparities in data conversion methods between Python and R. Many studies seem to rely on R for data analysis, however, the same type of analysis performed in R may not produce the same results using Python modules. For example, I had to use the R interface through rpy2 to run DESeq2 in Python to produce the same results. What is more, the methods and parameters for analysis as well as software versions are rarely well documented. As such, this ambiguity poses substantial obstacles to replicating results across different platforms and leaves room for misinterpretation. This lack of clarity also extends to the processing of raw data, with many studies failing to provide essential counts data in GEO, further hindering reproducibility efforts. Other than data issues, working on this project also revealed concerns in data visualization in many studies, with numerous figures lacking clarity or suffering from incomplete presentation. For example, the y-axis of the volcano plot published in this study only goes up to 8, however, the whole plot extends to y=300 (full plot in GitHub). These observations underscore the pressing need for more rigorous standards in data handling, analysis, and presentation to ensure the reliability and replicability of scientific research. Nevertheless, there's a growing awareness of the importance of good data management practices, with more recent publications showing improved data handling and availability. For instance, the study I used did a much better job compared to others by clearly laying out their methods and including all their data for others to replicate their analysis. In the future, I hope to see significant improvements in the clarity, reproducibility, and overall quality of scientific studies, ultimately enhancing the reliability and impact of research across various fields.