# 1. Introduction

To gain molecular insight into cellular function, researchers analyze various biological layers, including genomic DNA, chromatin structure, messenger RNA (mRNA), non-coding RNAs, proteins, post-translational modifications, and metabolites. Since the discovery of RNA’s role as the intermediary between the genome and the proteome, RNA sequencing (RNA-seq) has become a cornerstone of transcriptomic research. By quantifying mRNA levels across the genome, RNA-seq enables comprehensive profiling of gene activity, offering critical insights into cellular responses, disease mechanisms, and therapeutic effects (Byron et al., 2016).

## Research Context: Radiation-Induced Salivary Gland Injury

This paper centers its analytical walkthrough on a publicly available multi-omics dataset (GSE155902), which profiles radiation-induced dysfunction in mouse salivary glands. Ionizing radiation, commonly used in head and neck cancer therapy, often causes irreversible salivary gland damage, leading to xerostomia (dry mouth) and impaired quality of life. The dataset includes RNA-seq data from control and irradiated mouse parotid glands, offering a biologically meaningful context in which to explore differential gene expression, pathway perturbations, and cross-platform implementation.

By situating the workflow within this disease-relevant model, the paper not only demonstrates computational techniques but also highlights how transcriptomic analyses can uncover molecular mechanisms underlying tissue injury and repair.

## Scope of the Paper

This white paper presents a structured and reproducible RNA-seq differential expression pipeline in both R and Python. The focus is on gene expression changes between control and irradiated salivary gland tissues, with the goal of:

* Identifying differentially expressed genes (DEGs)
* Performing KEGG pathway enrichment
* Comparing analysis workflows across R and Python environments
* Evaluating the consistency and differences in DEG detection and enrichment outputs between platforms

A strong emphasis is placed on good practices for data quality control, transformation, and interpretation of results through visualizations. Although R and Python offer different tooling ecosystems, this guide demonstrates their complementary capabilities for reproducible transcriptomic analysis.

Note: While tools for batch effect correction (e.g., sva, ruv, ComBat) were considered, no batch correction was performed as the dataset included no batch metadata and exploratory plots showed no confounding structure.

# 2. Data Types & Study Design

The study design consists of six mouse samples:

* **Control group**: 3 unirradiated samples (UT1, UT2, UT3)
* **Irradiated group**: 3 samples exposed to radiation (IR1, IR2, IR3)

Input data:

* RNA-seq raw gene count files (Ensembl IDs)

Each RNA-seq file was processed and combined into a single count matrix with genes as rows and samples as columns.

# 3. Analysis Pipelines

|  |  |  |
| --- | --- | --- |
| **Feature** | **R Pipeline** | **Python Pipeline** |
| Normalization & DE | DESeq2 | pyDESeq2 (native Python reimplementation) |
| Enrichment Tool | clusterProfiler (KEGG via Entrez) | gprofiler-official (KEGG via Ensembl) |
| Visualizations | EnhancedVolcano, pheatmap | matplotlib, seaborn |
| DEG Output File | deseq2\_results\_R.csv | deseq2\_results\_python.csv |
| Filtered DEGs | padj < 0.05 and absolute log2FoldChange > 1 | padj < 0.05 and absolute log2FoldChange > 1 |
| Enrichment Output | kegg\_enrichment\_R.csv | gprofiler\_enrichment\_results\_python.csv |
| Figures | Volcano, PCA, sample distance, KEGG barplot, heatmap | Volcano, PCA, sample distance, KEGG barplot, heatmap |

# 4. Quality Control & Variance Stabilization

Both pipelines applied variance-stabilizing transformation (VST) for quality control:

* PCA plots showed clear separation between control and irradiated samples
* Sample-to-sample distance heatmaps demonstrated consistent clustering by condition

No batch correction was applied, as PCA showed no batch-like artifacts and no batch metadata was available.

# 5. Differential Gene Expression

R and Python both:

* Filtered out low-count genes
* Modeled gene expression with respect to condition
* Exported ranked DEG tables (log2 fold change, p-value, adjusted p-value)

**Comparison of DEG Results:**

* Number of DEGs (padj < 0.05 and |log2FC| > 1):
  + R: 1,050 genes (from deseq2\_results\_R.csv)
  + Python: 1,082 genes (from deseq2\_significant\_genes\_python.csv)
* Overlap between R and Python significant gene lists: 998 shared genes (95% agreement)
* Top genes (ranked by adjusted p-value) were consistent across pipelines, including **Cdkn1a**, **Gadd45a**, and **Ddit3**.

Minor discrepancies in DEG lists may arise from implementation differences in filtering or ID mapping.

# 6. Functional Enrichment Analysis

Both workflows performed KEGG pathway enrichment:

* R: clusterProfiler::enrichKEGG() using Entrez IDs
* Python: gprofiler-official via GProfiler using Ensembl IDs

**Comparison of Enrichment Results:**

* Top shared pathways: Hematopoietic cell lineage, Cytokine-cytokine receptor interaction, Cell adhesion molecules, Chemokine signaling pathway, Human T-cell leukemia virus 1 infection, Leukocyte transendothelial migration, and T cell receptor signaling pathway.
* Additional pathways (R only): Pancreatic cancer, Circadian rhythm, Primary immunodeficiency

These results show strong biological agreement with slight divergence due to ID mapping strategies and statistical cutoff implementation.

# 7. Visual Summaries and Interpretation

**PCA and Heatmap Comparison**

|  |  |
| --- | --- |
| Python | R |
|  |  |
|  | A screen shot of a chart  AI-generated content may be incorrect. |

The PCA plots from both R and Python pipelines show clear separation between control and irradiated samples. Notably, the R-based PCA explains a larger proportion of variance along PC1 (47%) compared to the Python version (26.2%). Despite this, both visualizations support strong condition-based clustering. Sample-to-sample distance heatmaps generated in R further confirm this structure, with irradiated and control samples forming distinct clusters. The top 30 variable gene heatmaps show overlapping gene sets and similar expression dynamics, though differences in color palette and dendrogram layout arise from default settings in seaborn and pheatmap.

Both heatmaps reveal consistent clustering of irradiated versus control samples, and while color palettes and dendrogram structures differ slightly due to software defaults, the underlying gene expression patterns and separation of sample groups remain aligned between platforms.

**Volcano Plot Comparison**

|  |  |
| --- | --- |
| Python | R |
|  |  |

Both volcano plots highlight highly significant genes such as Cdkn1a, Gdf15, and Eda2r, with a similar distribution centered around log2 fold change = 0. In both cases, the y-axis represents −log10 of the adjusted p-value (padj), ensuring consistency in significance scaling across platforms. However, there are minor discrepancies in classification, for instance, while Cxcl13 appears in both plots, it is labeled as significant for both adjusted p-value and fold change in the Python version (red), whereas the R version classifies it as significant only by adjusted p-value (blue). This divergence may result from slight differences in log2FoldChange thresholds or rounding behavior across implementations.

**KEGG Pathway Barplots**

|  |  |
| --- | --- |
| Python | R |
|  | A graph of different colored bars  AI-generated content may be incorrect. |

KEGG pathway barplots from both R and Python pipelines demonstrate strong biological consistency, with the top seven enriched pathways overlapping between platforms. While both plots highlight immune-related pathways such as hematopoietic cell lineage, cytokine-cytokine receptor interaction, and chemokine signaling, the R-based output includes three additional pathways. However, Pancreatic cancer, Circadian rhythm, and Primary immunodeficiency are not observed in the Python output. These differences may arise from varying statistical cutoffs or ID mapping strategies.

The visual styles also differ: R visualizes adjusted p-values (p.adjust) using a red-to-blue color scale and bar lengths based on gene counts, whereas Python displays raw p-values transformed via -log10(p) with a diverging color map. Despite these technical distinctions, the biological message remains consistent across platforms, supporting the robustness of the enrichment results.

# 8. Reflections on Tooling in R vs Python

The comparative workflow highlights key differences and strengths in the R and Python ecosystems for transcriptomic analysis:

* **RNA-seq Analysis**: R remains the gold standard, particularly with packages like DESeq2 and clusterProfiler offering mature, well-documented, and biologically validated pipelines. However, pyDESeq2 provides a native Python alternative that enables differential expression analysis without relying on R integration.
* **Visualization**: Python provides more flexible and customizable plotting via matplotlib and seaborn, though R’s domain-specific tools like EnhancedVolcano and pheatmap offer faster default outputs.
* **Identifier Mapping**: The use of Entrez (R) vs Ensembl (Python) IDs introduces subtle differences in gene set recognition during enrichment, highlighting the importance of ID consistency.
* **Reproducibility**: Both languages support reproducible workflows via RMarkdown or Jupyter notebooks. R has stronger community standards in bioinformatics, but Python allows for better integration with downstream machine learning or multi-omics platforms.
* **Limitations**: While pyDESeq2 is a promising tool, it may not yet match the full feature set and stability of R’s DESeq2. Similarly, enrichment tools in Python often rely on external APIs, which can affect reproducibility.

Ultimately, choice of platform should reflect the user’s broader goals, tool familiarity, and the intended integration with downstream analytics.

# 9. Conclusion

This white paper presented a reproducible RNA-seq differential expression pipeline in both R and Python using mouse salivary gland data after radiation exposure. Despite differences in implementation and toolsets, both pipelines produced consistent biological signals with high agreement on DEGs and pathway enrichment.

By comparing these parallel analyses, we demonstrate that both R and Python offer reliable approaches to transcriptomic analysis. R’s ecosystem remains the standard for bioinformatics, but Python’s growing libraries allow seamless integration into machine learning workflows.

Ultimately, language choice should reflect team expertise, existing infrastructure, and analytical goals. When validated appropriately, either approach can uncover meaningful biological insight.

Future work may explore expanding this dual-pipeline approach to multi-omics integration, enabling broader comparisons across data types and toolchains.

# 

# 10. References

Byron, S. A., Van Keuren-Jensen, K. R., Engelthaler, D. M., Carpten, J. D., & Craig, D. W. (2016). Translating RNA sequencing into clinical diagnostics: Opportunities and challenges. *Nature Reviews Genetics, 17*(5), 257–271. <https://doi.org/10.1038/nrg.2016.10>

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., ... & Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology, 17*, 13. <https://doi.org/10.1186/s13059-016-0881-8>

Meeks, L. M., de Oliveira Pessoa, D., Martinez, J. A., Limesand, K. H., & Padi, M. (2021). *Integrating metabolomics and transcriptomics reveals convergent pathways driving radiation-induced salivary gland dysfunction* [Data set]. NCBI Gene Expression Omnibus.<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE155902>​[NCBI+1PMC+1](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE155902&utm_source=chatgpt.com)

Meeks, L. M., de Oliveira Pessoa, D., Martinez, J. A., Limesand, K. H., & Padi, M. (2020). *Supplementary data for integrating metabolomics and transcriptomics reveals convergent pathways driving radiation-induced salivary gland dysfunction* [Data set]. Zenodo.<https://doi.org/10.5281/zenodo.4391402>