

University of L'Aquila

Bachelor's Degree in Computer Science (Erasmus)

Academic Year 2024/2025



Bioinformatics

MiRNASearch

MiRNASearch Web Application

Ata Uzay Kuzey – 303447 – [atauzay.kuzey@student.univaq.it](mailto:atauzay.kuzey@student.univaq.it)

Elif Ece Can – 303487 – [elifece.can@student.univaq.it](mailto:elifece.can@student.univaq.it)

## Contents

1.0 Introduction.....	2
2.0 Materials and Methods.....	2
2.1 Technologies .....	2
2.2 Methods.....	3
3.0 Application Functionality and Usage .....	3
4.0 Conclusion .....	3
Appendix A - Medium Sensitive Prefiltering Parameters for Database Population .....	5
Appendix B - Neo4j Query .....	6
Appendix C - Graphical Interface of the Application.....	7

## 1.0 Introduction

MiRNAs are short RNA molecules that play a crucial role in gene expression by binding to messenger RNAs. However, accurately identifying which target genes and pathways miRNAs are involved in is challenging, due to varied methodologies (e.g., experiment types, scoring metrics) employed by different bioinformatics tools. To address this problem, we designed an application that gathers information from four established prediction databases—PicTar, miRTarBase, TargetScan, and RNA22—to retrieve gene targets and pathways based on user-defined merge strategies. This application aims to increase the efficiency and accuracy of miRNA target and pathway analysis, thus easing the complexities involved in understanding miRNA-gene relationships.

## 2.0 Materials and Methods

In this section technologies and methodologies used in this project will be listed. In this project “Mus Musculus” species is examined, to change the species, specific species tables can be downloaded from the mentioned websites of miRBase, miRTarBase, TargetScan, RNA22, and PicTAR.

### 2.1 Technologies

- **Java:** We implemented the backend APIs using Java — a proven, reliable language.
- **Spring Boot:** We used the Spring Boot framework to streamline application configuration and rapid development of our RESTful services.
- **Neo4j(version 1.6.1):** We use Neo4j as a graph database to model and query relationships among microRNAs, genes, and pathways.
- **miRBase:** This resource is essential for populating our database with comprehensive microRNA data.[miRBase - Downloads](#)
- **miRTarBase:** We use miRTarBase to establish vital connections between microRNAs and their target genes.
- **TargetScan (Release 8.0):** Another key source for inferring microRNA-gene interactions, specifically using data from [TargetScanMouse 8.0](#).
- **RNA22:** This platform provides additional sets of predictions for microRNA-gene connections, accessible via their [Computational Medicine Center at Thomas Jefferson University](#) portal.
- **PicTar:** For further insights into microRNA-gene interactions, we integrate data from [PicTar's tables](#) within the UCSC Genome Browser.
- **KEGG API:** We utilize the [KEGG REST API](#) to establish crucial gene to pathway connections within our data, providing a broader biological context.
- **Vue.js(version 3.4.29):** Used to create the frontend of the application
- **Tailwind.css(version 3.4.9):** Used for frontend styling

## 2.2 Methods

2.2.1: Prefiltering: Before populating the Neo4j database, all four external databases undergo a prefiltering process. The specific parameters for this filtering are detailed in Appendix A. Prefiltering parameters can be specified via command-line before database population.

2.2.2: Web Backend Design: Our backend is implemented in Java Spring Boot. In the backend, a single core method handles all database interactions to retrieve lists of genes, pathways, scores, and query time. That database query itself is designed to incorporate miRNA names, tool types, tool-selection strategies, and miRNA-specific heuristics—thereby minimizing any additional filtering in Java (see Appendix B).

2.2.3: Frontend Design: Our frontend features a single-page UI design, supporting both single and multiple miRNA searches. Results are displayed comprehensively in both table and graph formats, with visual examples provided in Appendix C.

2.2.4: Testing: We wrote and executed unit tests for the core backend methods; these tests are available in the project's GitHub backend repository.

## 3.0 Application Functionality and Usage

The application's primary function is to **search for target genes and pathways** associated with a specific microRNA (miRNA) or a set of miRNAs, based on user-defined parameters. Users can select **specific prediction tools** to employ and determine how to **merge tool results**, choosing options like requiring consensus from at least two tools or using the intersection of all selected tools. When querying multiple miRNAs, users can further refine their results by selecting a **merge strategy based on miRNAs**, such as the intersection of all queried miRNAs or a majority consensus.

Results are presented in both **table and interactive graph formats**. The graph allows users to **zoom in/out, drag nodes, and view tool names** on connections by hovering over them. The results table can be **exported in CSV format**. Within the table, in addition to gene and pathway data, users can view the **tool's score** (if available) or the **experimental origin** of the result if no score is provided. For user convenience, the application also stores **past searches**, enabling users to easily re-query previous parameters. Example visuals are provided in **Appendix C**.

## 4.0 Conclusion

In this project, we have created a user-friendly web application that unifies four leading miRNA–target prediction resources (PicTar, miRTarBase, TargetScan, and RNA22) for *Mus musculus* and leverages the KEGG API to map predicted targets onto biological pathways. Users can specify one or more miRNAs, choose among prediction tools or combine their outputs using customizable selection strategies, and apply heuristic filters when querying multiple miRNAs. Results are presented in both tabular and graphical formats, displaying target genes, associated pathways, confidence scores, supporting experiments, and query

execution time. As a next step, score filtering could be migrated to the client side—allowing users to adjust thresholds dynamically—rather than applying a fixed pre-filter.

## Appendix A - Medium Sensitive Prefiltering Parameters for Database Population

<b>Database</b>	<b>Filtering Criteria</b>
<b>RNA22</b>	Retained only entries with a score less than –18.
<b>TargetScan</b>	A minimum score of 0.00 and an experiment type explicitly identified as “8mer,7mer-m8.” are applied.
<b>PicTar</b>	For medium sensitivity predictions, only those entries with a score greater than 200 were selected.
<b>miRTarBase</b>	Included only entries validated by two or more experimental methods (ex. luciferase reporter assay, qRT-PCR, Western blot, or direct functional studies).

## Appendix B - Neo4j Query

```

MATCH (m:microRNA)
WHERE m.name IN $miRNANames

MATCH (m)-[r]->(t:Target)
WHERE type(r) IN $tools

OPTIONAL MATCH (t)-[:PART_OF_PATHWAY]->(p:Pathway)

WITH
  t,
  t.name AS gene,
  collect(DISTINCT type(r)) AS foundTools,
  collect(DISTINCT p.name) AS pathways,
  collect(
    DISTINCT
    {
      tool: type(r),
      quality: toString(
        CASE
          WHEN r.experiments IS NOT NULL THEN r.experiments
          WHEN r.pct_score IS NOT NULL THEN r.pct_score
          ELSE r.score
        END
      ),
      mirna: m.name
    }
  ) AS connections,
  collect(DISTINCT m.name) AS mirnasWithPrediction

WITH
  gene,
  foundTools,
  [path IN pathways WHERE path IS NOT NULL] AS pathways,
  connections,
  mirnasWithPrediction,
  size(mirnasWithPrediction) AS foundCount,
  CASE
    WHEN toUpper($heuristic) = 'INTERSECTION' THEN size($miRNANames)
    WHEN toUpper($heuristic) = 'MAJORITY' THEN floor(size($miRNANames) / 2.0 + 1)
    ELSE 1
  END AS requiredCount

WHERE
  (
    toUpper($toolSelection) = 'UNION'
    OR
    (toUpper($toolSelection) = 'INTERSECTION' AND size(foundTools) = size($tools))
    OR
    (toUpper($toolSelection) = 'AT_LEAST_TWO' AND size(foundTools) >= 2)
  )
  AND
  foundCount >= requiredCount

RETURN
  gene,
  foundTools AS tools,
  pathways,
  connections
ORDER BY gene

```

## Appendix C - Graphical Interface of the Application

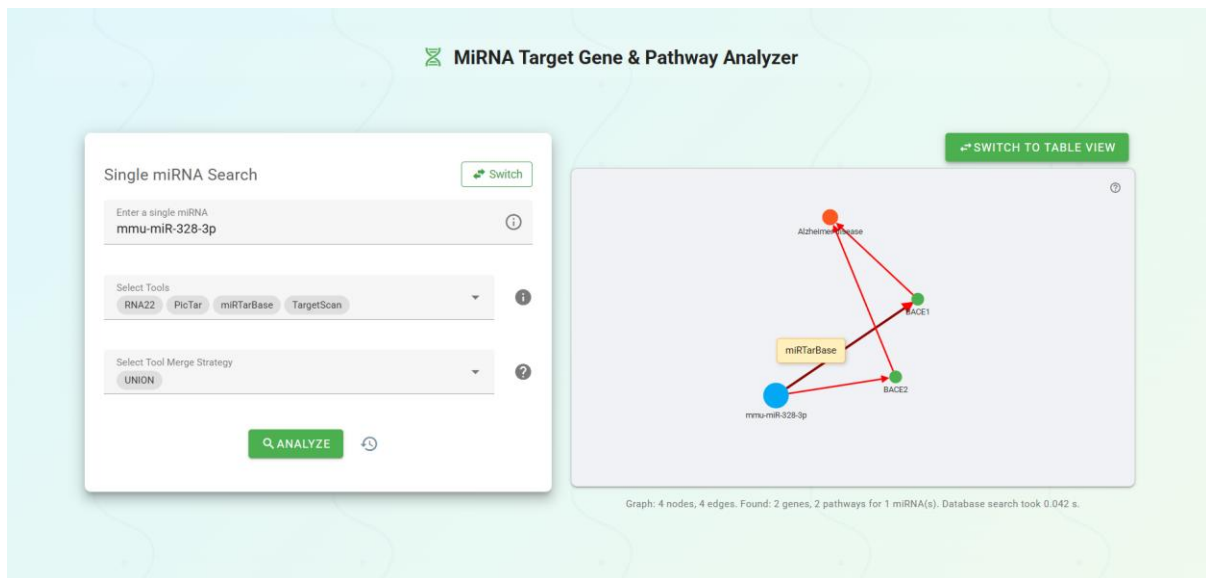


Figure 1: Single-MiRNA Search (mmu-miR-328-3p) Results In Graph

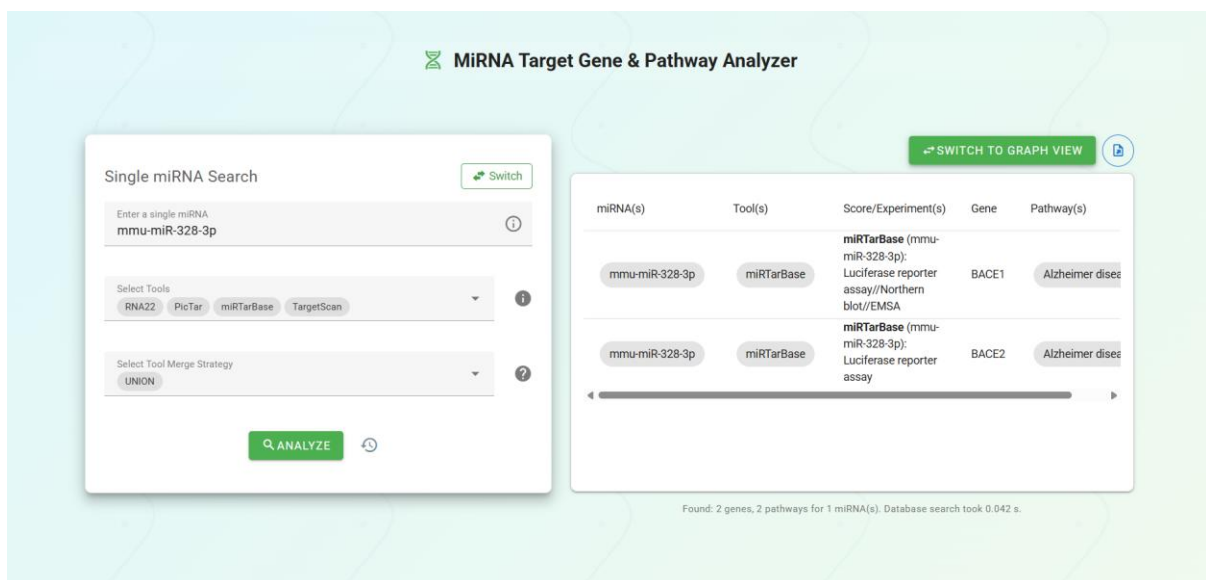


Figure 2: Single-MiRNA Search (mmu-miR-328-3p) Results In Table



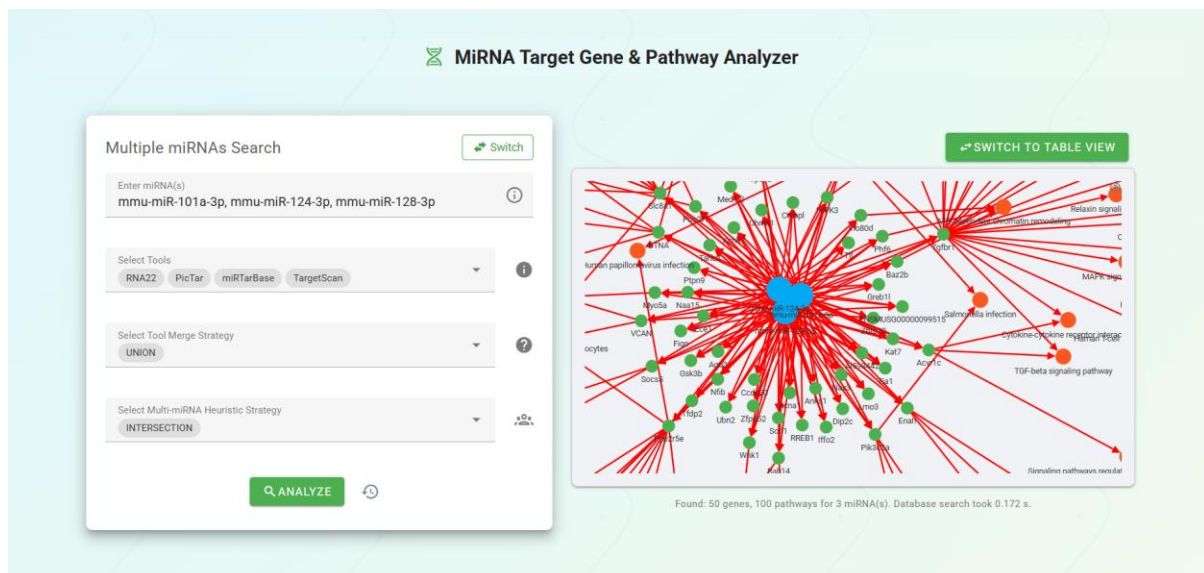


Figure 3: Multiple MiRNA Search (mmu-miR-101a-3p, mmu-miR-124-3p, mmu-miR-128-3p) With Results With Graph

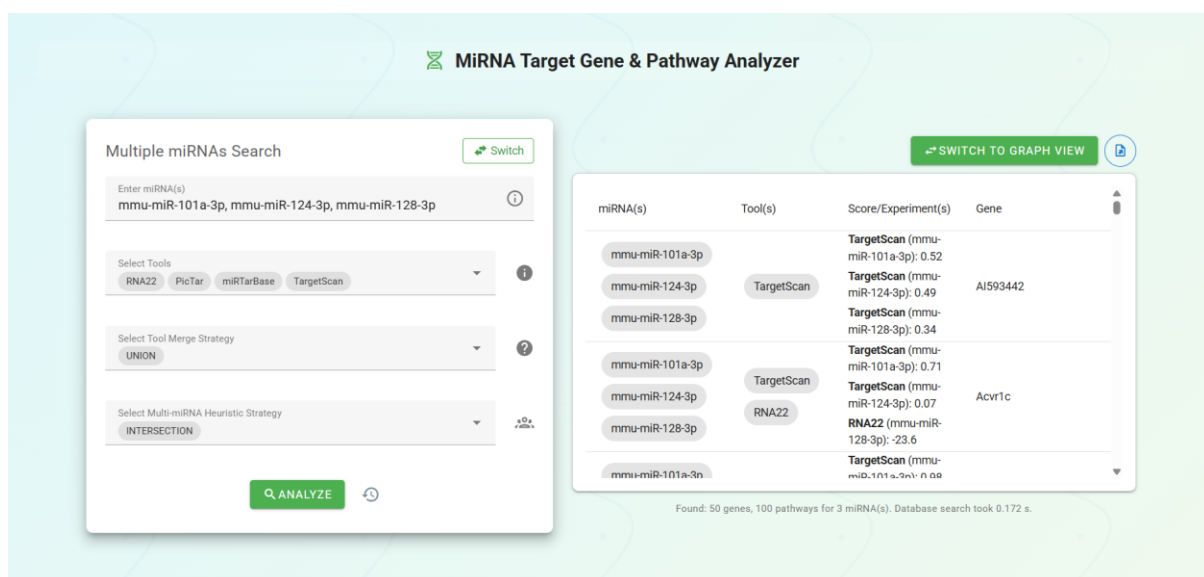


Figure 4: Multiple MiRNA Search (mmu-miR-101a-3p, mmu-miR-124-3p, mmu-miR-128-3p) With Results With Table

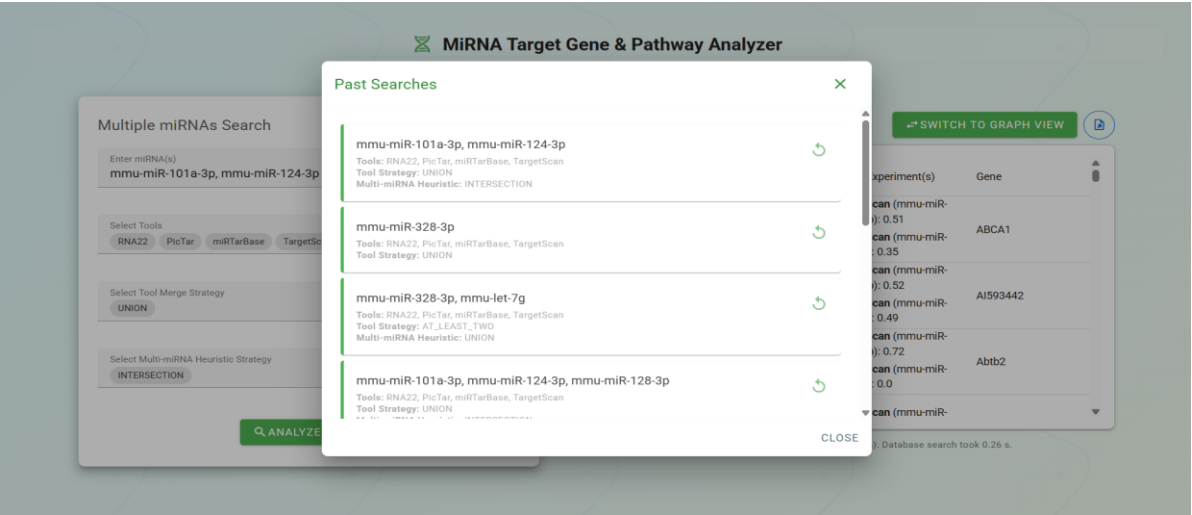


Figure 5: Past Searches of the User