



OMICS-query: A Chatbot Interface for *nf-core/rnaseq* Results

Camilla Callierotti

Milano, xxx

- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**

- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**

Project Motivation

Challenges and Solution

1. COMPLEXITY

RNA-seq analyses generate **multiple data types** for multiple comparisons within multiple sample subsets

2. TIME

Manual result interpretation requires **significant analyst time** and perhaps support

3. EXPERTISE

Researchers may lack the **data skills** required to **access** and **extrapolate** all that the results have to offer

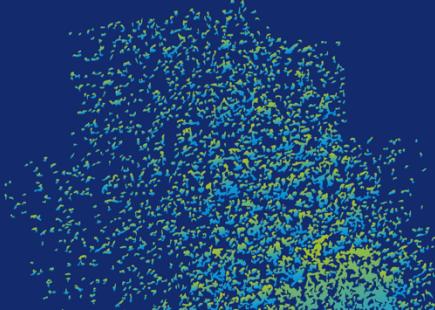
4. SCALABILITY

Growing client base requires more independent result interpretation and less support

OMICS-QUERY...

- ✓ Interprets results through **natural language** queries
- ✓ Generates appropriate **visualizations** automatically
- ✓ Provides scientifically accurate **explanations** of methods
- ✓ Reduces manual **support burden**
- ✓ Improves **client experience**
- ✓ Suggests areas for **improvement**

- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**



Introduction to Large Language Models (LLMs)

Architecture

An LLM is a machine learning model trained to **process, understand, and generate** human language.



After general training, models can be adapted to specific tasks or modes of interaction by using **smaller, labelled** datasets.

- 'instruct': fine-tuned to follow prompted instructions, expect to do something
- 'chat' fine-tuned to work in chatbots, expect to be involved in conversation

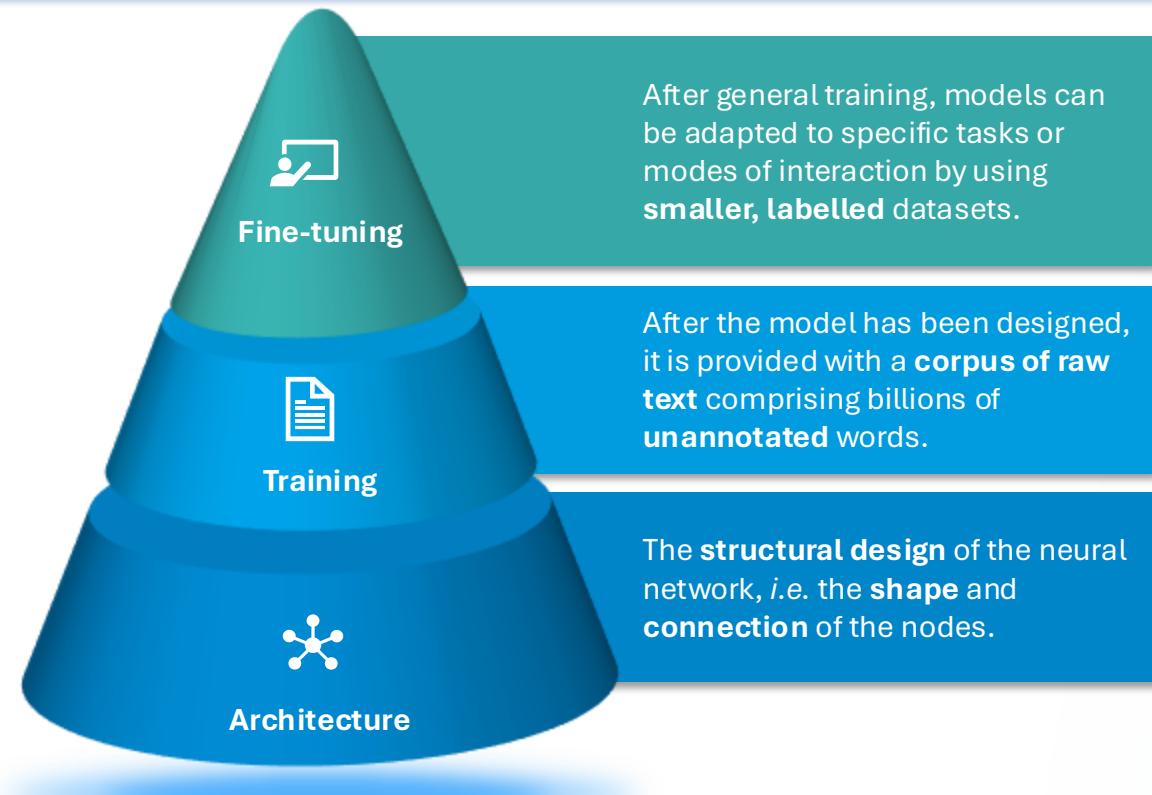
After the model has been designed, it is provided with a **corpus of raw text** comprising billions of **unannotated words** (Wikipedia, PubMed, etc.). The model learns to predict the next word in a sequence, thereby **learning statistical patterns of language, grammar, and semantics** without explicit annotation.

The **structural design** of the neural network, i.e. the **shape and connection** of the nodes, dictating how information flows between them. Different architectures have been developed over time, but the current **standard** architecture for LLMs is **transformer-based**.

Introduction to Large Language Models (LLMs)

Architecture

An LLM is a machine learning model trained to **process, understand, and generate** human language.



The models we work with already incorporate all three layers of specificity provided by their developers (architecture → training → fine-tuning). When we download and use them, they are fully prepared at this foundational level.

Examples:

meta-llama/Llama-4-Maverick-17B-128E

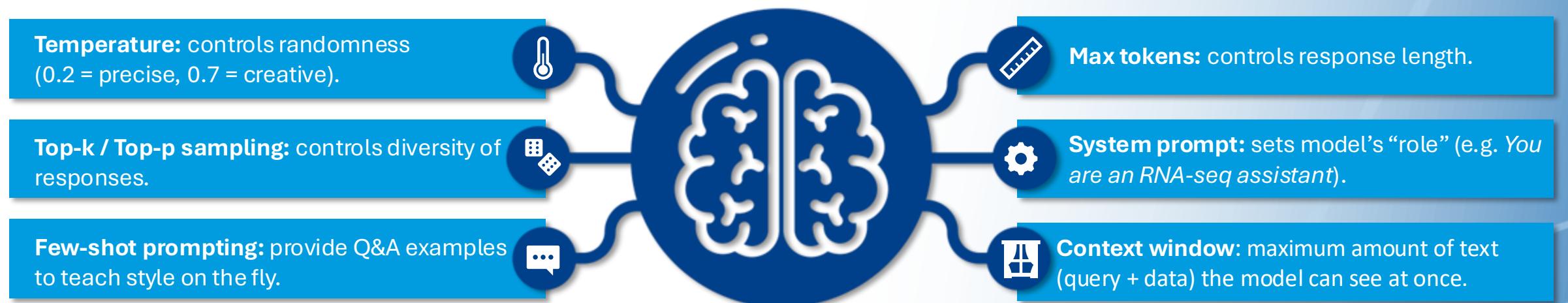
- *Llama architecture*
- *17 billion parameters*
- *Pre-trained and fine-tuned by Meta*
- mistralai/Mixtral-8x7B-Instruct-v0.1**
- *Mixtral architecture*
- *7 billion parameters*
- *Fine-tuned on instruction-following datasets*

Our role begins **after these steps**: we take these ready-made models and apply additional customization to adapt them to our own tasks and data.

Introduction to Large Language Models (LLMs)

Controlling Model Behaviour

*Additional parameters make the **same model** behave **differently**.*

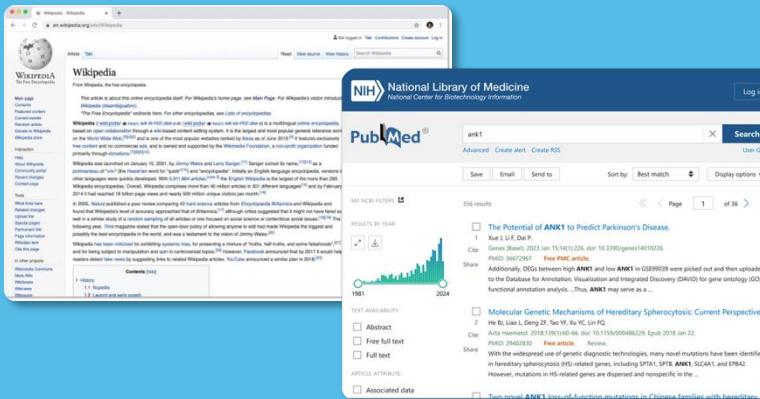


Introduction to Large Language Models (LLMs)

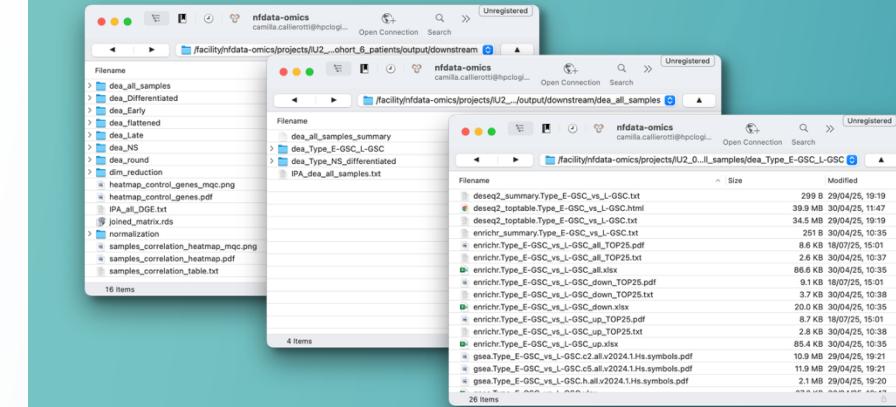
Knowledge

We leverage **pre-trained** and **fine-tuned** LLMs — already structured at the levels of architecture, general training, and instruction/chat specialization — and combine them with our **RNA-seq data** to provide **project-specific insights**, without retraining or modifying the model itself.

LLM Pre-trained Knowledge



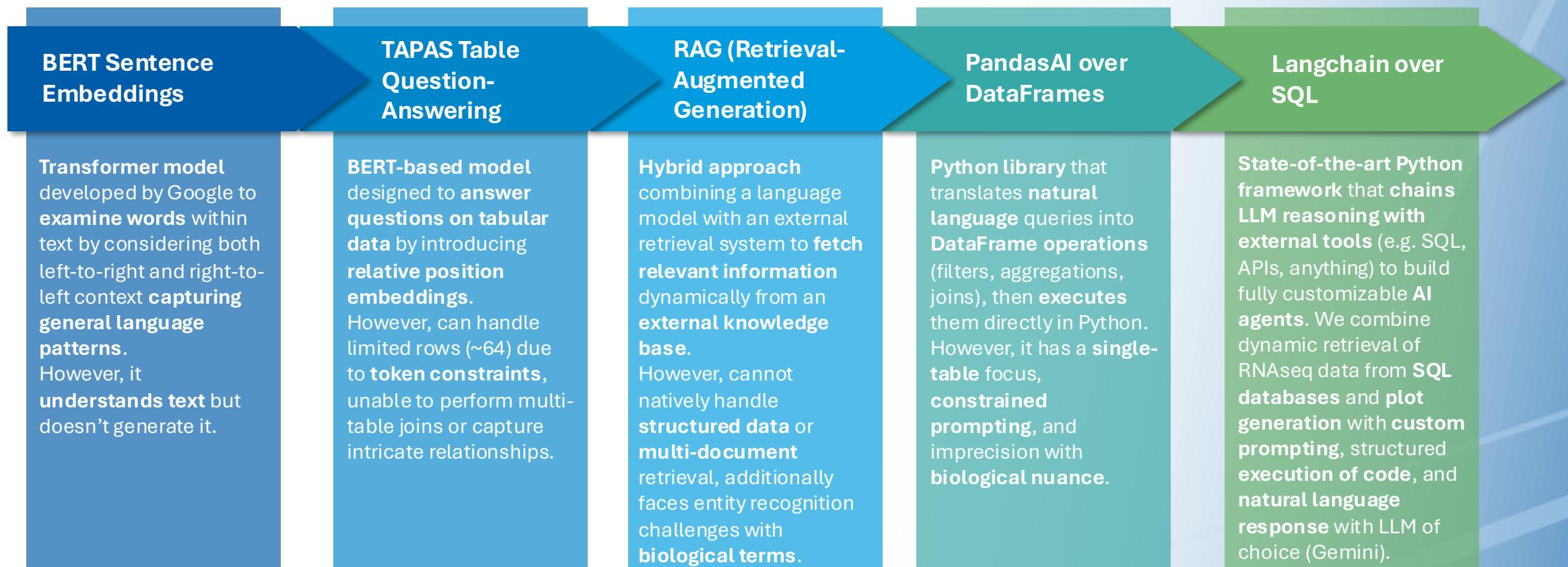
Project-specific RNA-seq Data



- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**

Evolution of Approaches

From Intent Classification to Agentic AI



- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**



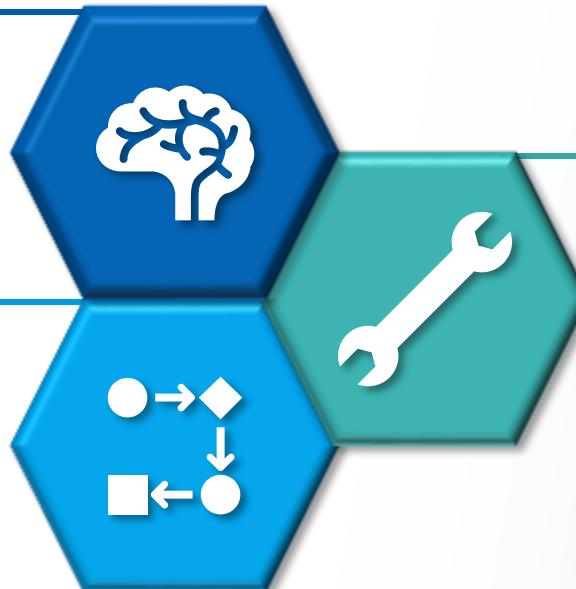
*LangChain provides a framework that connects **LLMs** with **external tools** so they can solve complex problems **autonomously**. Instead of an LLM answering in isolation, LangChain allows it to **reason**, **call tools**, and integrate results back into the **final answer**.*

Agent

Reasoning engine able to decide the appropriate actions to take and the order in which to take them

Chains

Structured sequences of steps that combine model reasoning and tool use into coherent workflows

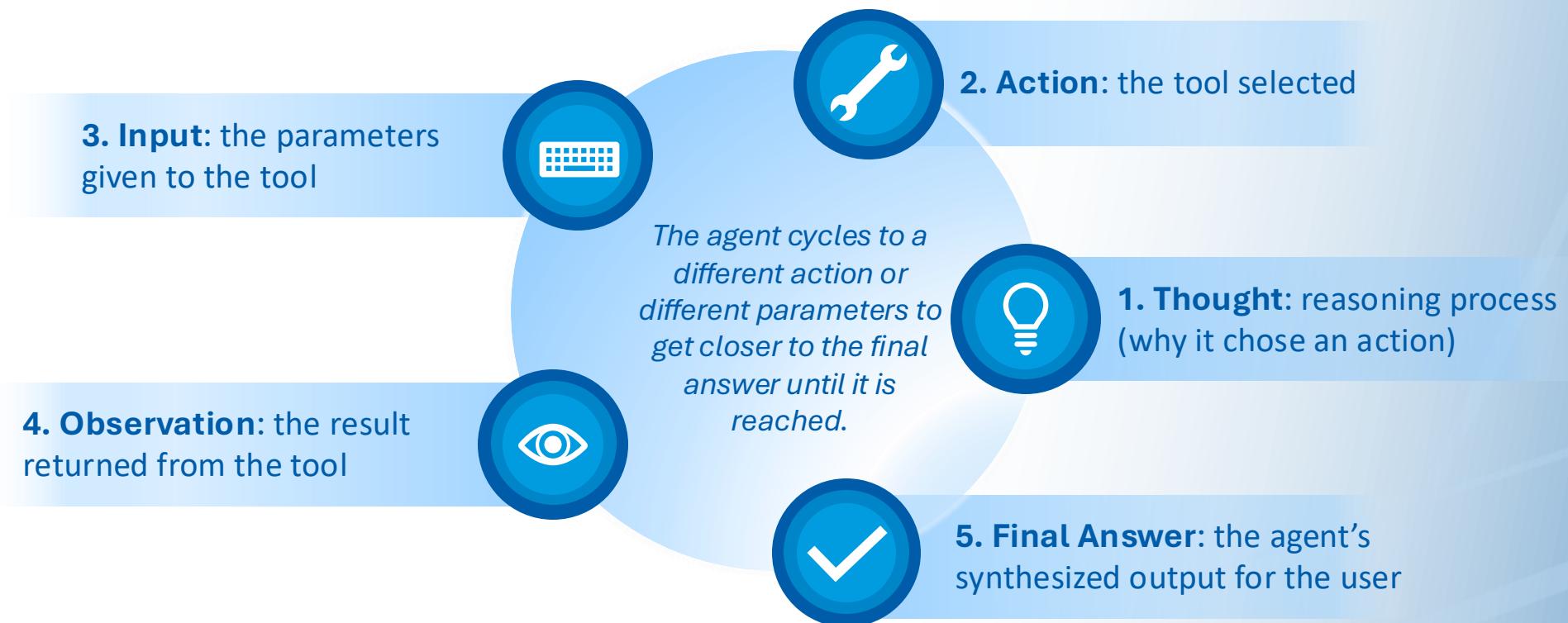


Tools

Interfaces that the agent can make use of (in our case: SQL tool, database schema tool, Plotly tool, and other auxiliary tools)



We make use of a **single agent system** which uses **ReAct logic** (Reasoning and Acting) to **determine the sequence of actions** to take. This prompt engineering technique will work by generating:



LangChain

Agents in Action



LangChain

In practice, this results in a **transparent reasoning loop** visible to developers, showing each **decision, tool call, and correction step until the final response is produced.**

Transparent reasoning improves reliability and debuggability

```
***  
1 > Entering new AgentExecutor chain...  
2  
INFO:httpp:HTTP Request: POST https://api.mistral.ai/v1/chat/completions "HTTP/1.1 200 OK"  
3 < json  
4 {  
5   ...  
6     "action": "Database_Schema",  
7     "action_input": "NS_flattening_yes_vs_no"  
8   }  
9  
INFO:agent:DATABASE_SCHEMA_TOOL called  
10 INFO:agent:Schema result: {'success': True, 'tables': {'dea_NS_Flattening_Yes_No_deseq2':  
11   {'columns': [{ 'name': 'gene_name', 'type': 'TEXT'}, { 'name': 'baseMean', 'type': 'FLOAT'},  
12     ('name': 'log2FoldChange', 'type': 'FLOAT'), { 'name': 'pvalue', 'type': 'FLOAT'}, { 'name':  
13       'padj', 'type': 'FLOAT'}, { 'name': 'Significance', 'type': 'TEXT'}, { 'name': 'GeneID',  
14     'type': 'TEXT'}, { 'name': 'Chr', 'type': 'TEXT'}, { 'name': 'Start', 'type': 'TEXT'},  
15     ('name': 'End', 'type': 'TEXT'), { 'name': 'Strand', 'type': 'TEXT'}, { 'name': 'Length',  
16     'type': 'BIGINT'}, { 'name': 'sample_query', 'type': 'TEXT'}], 'sample_query': 'SELECT * FROM dea_NS_Flattening_Yes_No_deseq2 LIMIT 5'; } }  
17  
Observation: Available tables and their schemas:  
18 Table: dea_NS_Flattening_Yes_No_deseq2  
19 Columns:  
20   - gene_name (TEXT)  
21   - baseMean (FLOAT)  
22   - log2FoldChange (FLOAT)  
23   - pvalue (FLOAT)  
24   - padj (FLOAT)  
25   - Significance (TEXT)  
26   - GeneID (TEXT)  
27   - Chr (TEXT)  
28   - Start (TEXT)  
29   - End (TEXT)  
30   - Strand (TEXT)  
31   - Length (BIGINT)  
32   - Sample query: SELECT * FROM dea_NS_Flattening_Yes_No_deseq2 LIMIT 5;  
33  
34  
35 INFO:httpp:HTTP Request: POST https://api.mistral.ai/v1/chat/completions "HTTP/1.1 200 OK"  
36 json  
37 {  
38   ...  
39     "action": "SQL_Query",  
40     "action_input": "SELECT * FROM dea_NS_Flattening_Yes_No_ora_curated_gene_sets WHERE p.adjust < 0.05 LIMIT 5"  
41   }  
42 INFO:agent:EXECUTING_SQL_QUERY: SELECT * FROM  
43 dea_NS_Flattening_Yes_No_ora_curated_gene_sets WHERE p.adjust < 0.05 LIMIT 5  
44 INFO:agent:QUERY RETURNED 0 rows  
45  
46 Observation: Error: Query execution failed: no such column: p.adjust  
47 INFO:httpp:HTTP Request: POST https://api.mistral.ai/v1/chat/completions "HTTP/1.1 200 OK"  
48 json  
49 {  
50   ...  
51     "action": "SQL_Query",  
52     "action_input": "SELECT * FROM dea_NS_Flattening_Yes_No_ora_curated_gene_sets WHERE  
53       value < 0.05 LIMIT 5"  
54   }  
55 INFO:agent:EXECUTING_SQL_QUERY: SELECT * FROM  
56 dea_NS_Flattening_Yes_No_ora_curated_gene_sets WHERE value < 0.05 LIMIT 5  
57 INFO:agent:QUERY RETURNED 0 rows  
58  
59 Observation: Query returned 0 rows. Here are all the results:
```

```
***  
60  
61 Observation: Query returned 5 rows. Here are all the results:  
62  
63 ID | Description | GeneRatio | BrRatio | RichFactor | FoldEnrichment | zScore | pvalue | padj |  
64 -----  
65 1 | BENPORATH_SUZ12_TARGETS | BENPORATH_SUZ12_TARGETS | 26/122 | 730/14780 | 0.056164385561644  
66 2 | 3.39853967350e-07 | 3.39853967350e-07 | 0.056164385561644  
67 SYT#/PYSL5/GRA2/SLC6A3/ESAN/NTRK3/GRIN1/ERBB4/FRMPO1/NTRK3/PTCH1/ASTN2/HIF1  
68 3 | 2.05e-05  
69  
70 BENPORATH_ES_WITH_H3K27ME3 | BENPORATH_ES_WITH_H3K27ME3 | 26/122 | 798/14780 | 0.052581455620  
71 1.359677604359e-06 | 1.26164866454283e-06 |  
72 SYT#/PYSL5/GRA2/SLC6A3/ESAN/NTRK4/NTRK2/NGRN1/LUNPF3/NP1/ARL9/GRIN1/ERBB4  
73  
74 REACTOME_NEURONAL_SYSTEM | REACTOME_NEURONAL_SYSTEM | 34/122 | 332/14780 | 0.046977963977963  
75 0.000393123509594793 | 0.000393123509594793 | GRIN2/SLC6A3/SLC6A3/LUNPF3/NGRN1/ERBB4/HIF1  
76 MEISSNER_NPC_HCP_WITH_H3K27ME3_AND_H3K27ME3 | MEISSNER_NPC_HCP_WITH_H3K27ME3_AND_H3K27ME3 | 9/12  
77 2.98007114631253 | 1.45534531031059e-07 | 0.000429466244278393 | 0.000429466244278393 | PLXMA  
78 KEGG_NEUROACTIVE_LIGAND_RECECTOR_INTERACTION | KEGG_NEUROACTIVE_LIGAND_RECECTOR_INTERACTION  
79 7.645369084485693 | 1.43065704957967e-06 | 0.0005948672021215226 | 0.000560215181519617 | GRIN1  
80  
81 This is the actual data from the database. Use this to answer the user's question with  
82 specific details.  
83 This data has been stored and is available for plotting if visualization would be  
84 helpful.  
85 INFO:httpp:HTTP Request: POST https://api.mistral.ai/v1/chat/completions "HTTP/1.1 200 OK"  
86 json  
87 {  
88   ...  
89     "action": "Create_Plot",  
90     "action_input": "bar"  
91   }  
92 ***INFO:httpp:HTTP Request: POST https://api.mistral.ai/v1/chat/completions "HTTP/1.1 200 OK"  
93 ======  
94 GENERATED_CODE:  
95 """ python#import plotly.express as px\nfig = px.bar(df, x='ID', y='FoldEnrichment',  
96 title='Fold Enrichment by ID')\nfig.write_html('plot.html')\n======  
97  
98 Observation: SUCCESS: bar plot created successfully  
99 Plot saved to plot.html  
100 INFO:httpp:HTTP Request: POST https://api.mistral.ai/v1/chat/completions "HTTP/1.1 200 OK"  
101 json  
102 {  
103   ...  
104     "action": "Final Answer",  
105     "action_input": "In NS samples, the differentially expressed RNA pathways between  
106 flattening yes and no include several significant pathways. These pathways are:\n107 1. BENPORATH_SUZ12_TARGETS (Fold Enrichment: 4.31, q-value: 3.76e-09)\n108 2. BENPORATH_ES_WITH_H3K27ME3 (Fold Enrichment: 3.95, q-value: 1.24e-06)\n109 3. REACTOME_NEURONAL_SYSTEM (Fold Enrichment: 5.44, q-value: 0.000198)\n110 4. MEISSNER_NPC_HCP_WITH_H3K27ME3_AND_H3K27ME3 (Fold Enrichment: 8.79, q-value: 0.000644)\n111 5. KEGG_NEUROACTIVE_LIGAND_RECECTOR_INTERACTION (Fold Enrichment: 8.26, q-value: 0.000560  
112 0.05.  
113 I have also created a bar plot to visualize the enrichment of these pathways.  
114 The plot shows the fold enrichment values for each pathway, highlighting their  
115 significance in the comparison between Flattening yes and no conditions in NS samples.  
116 ...  
117  
118 109 > Finished chain.  
119  
120
```

Gemini

Gemini as the LLM Backbone



Gemini (Google DeepMind) was selected as the **foundation LLM** for several compelling reasons tailored exactly to our **niche needs** in RNA-seq analysis and biological context:

Multimodal, Large-Context Model

Gemini Pro (especially versions 2.0 and 2.5) supports up to **1 million tokens**, with planned extensions to **2 million**—far beyond the limitations of other LLMs. This allows the model to ingest extensive biological datasets, full gene expression tables, or multi-omics context in one go

Advanced Reasoning Capabilities

The Gemini 2.5 Pro model was designed for deep, **step-by-step reasoning**, which is ideal for **generating** complex biological queries, **interpreting** statistical results, or **formulating** nuanced conclusions

Native Tool Calling

Gemini inherently supports **structured tool use** via function calling with **JSON outputs**, making it ideal to integrate with our SQL, schema, and plotting **tools** in **LangChain**

Deep Biological Knowledge

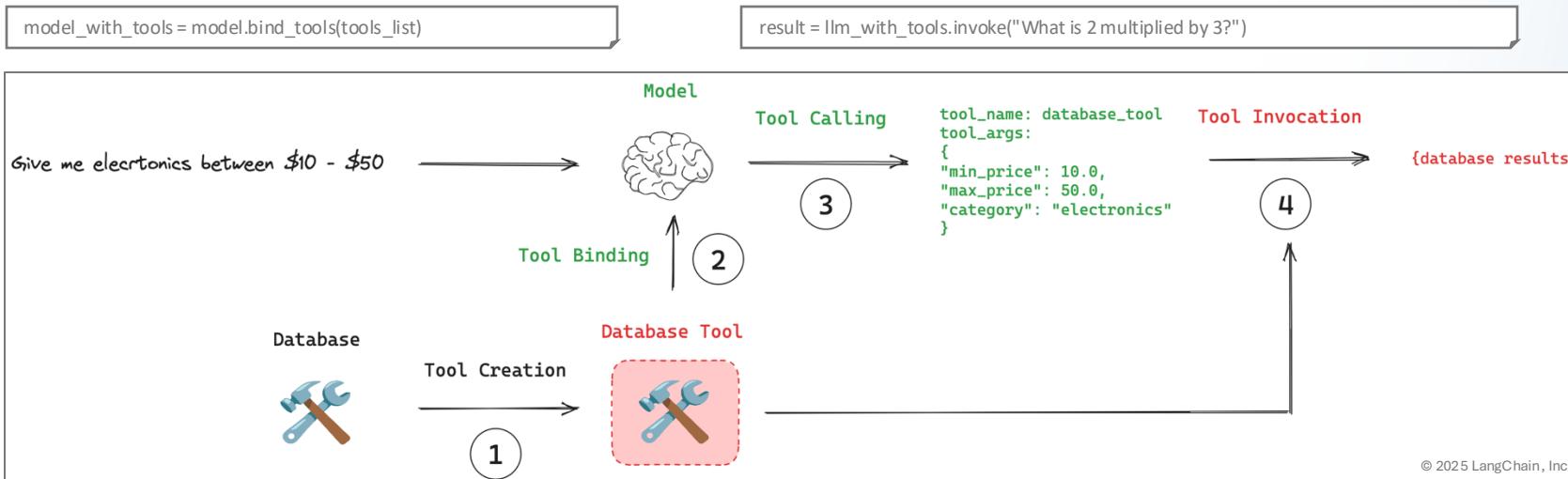
As a Google DeepMind model trained on web documents, scientific literature, and code, Gemini brings a **broad inferential foundation** that includes **genomic** and **bioinformatic contexts**, which supports more accurate interpretation of RNA-seq queries and visualizations—even if it's not explicitly trained in biology

Gemini

Gemini as the LLM Backbone



Gemini (Google DeepMind) was selected as the **foundation LLM** for several compelling reasons tailored exactly to our **niche needs** in RNA-seq analysis and biological context:



```
from langchain_core.tools import tool

@tool
def multiply(a: int, b: int) -> int:
    """Multiply a and b."""
    return a * b
```



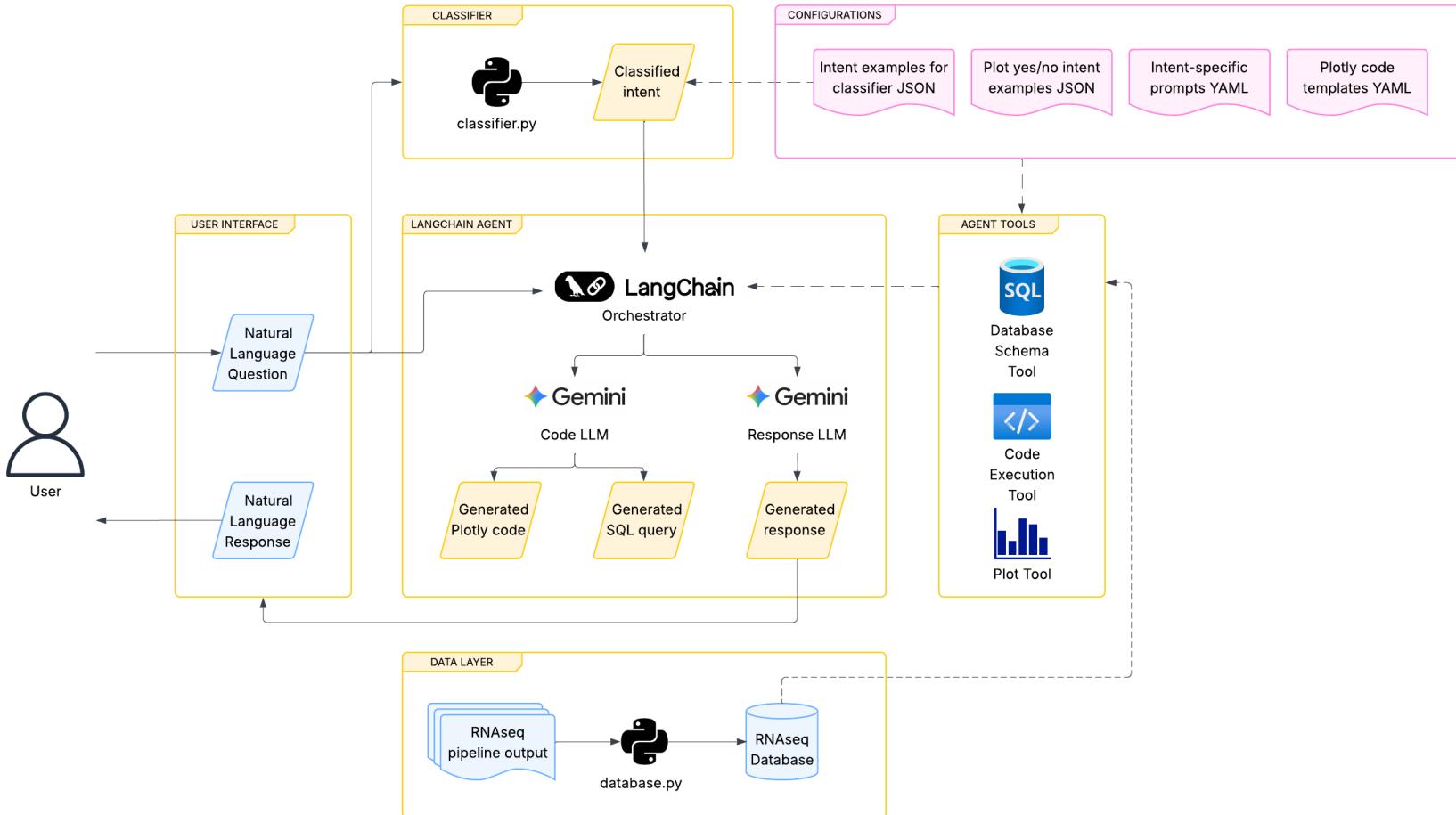
```
# Tool creation
tools = [my_tool]
# Tool binding
model_with_tools = model.bind_tools(tools)
# Tool calling
response = model_with_tools.invoke(user_input)
```

Native Tool Calling

Gemini inherently supports **structured tool use** via function calling with **JSON outputs**, making it ideal to integrate with our SQL, schema, and plotting tools in **LangChain**

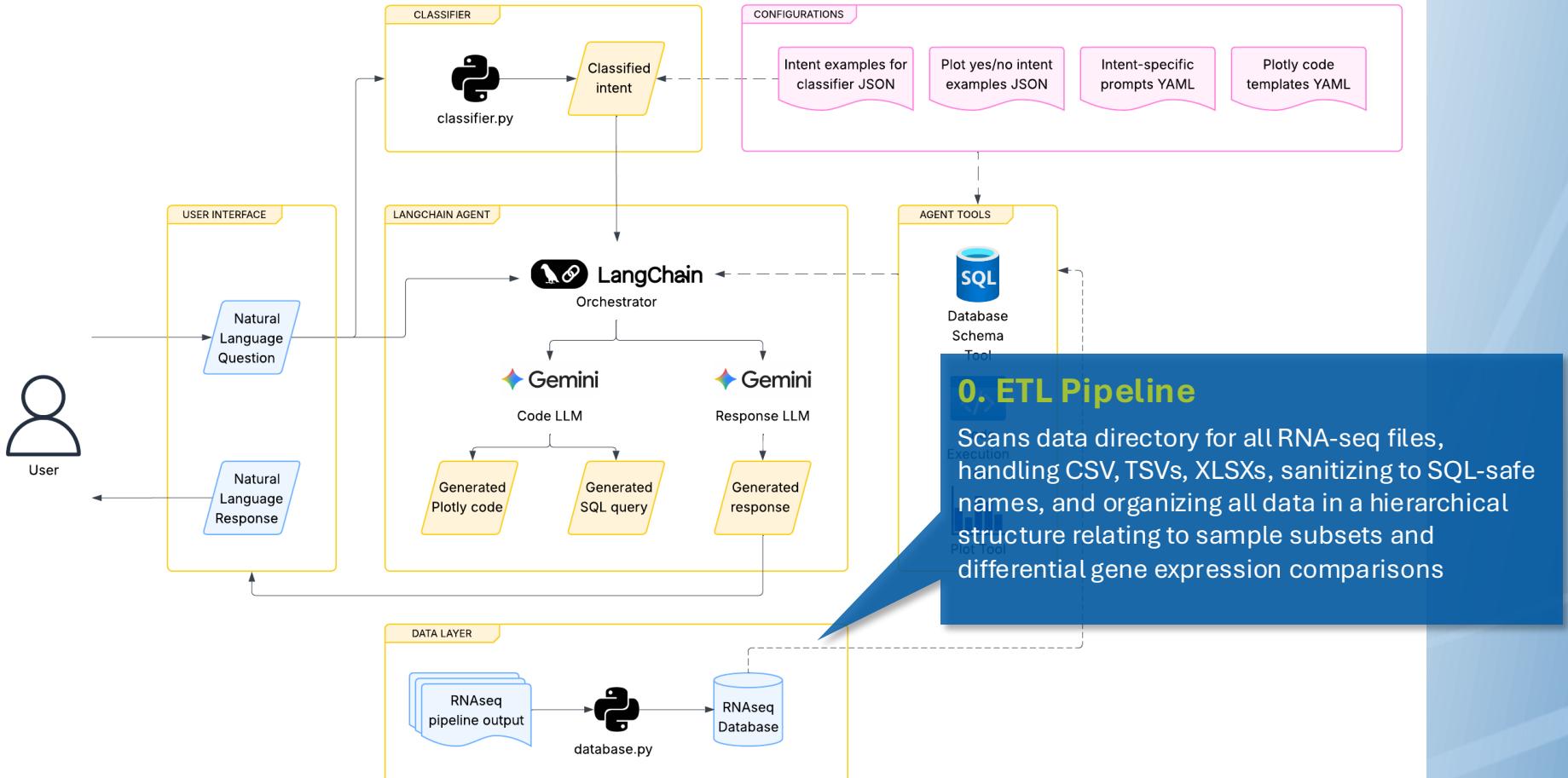
Solution architecture

High Level Architecture



Solution architecture

High Level Architecture



Solution architecture

Aside on SQL Injector

Currently, nf-core pipeline outputs are **file-based** and **scattered** across directories. I developed an **automated ETL pipeline** transforms flat files into **normalized relational database**

```

[nfdatautils]@camilla:~/calligrapher@node02:/factory/nfdata-omics/projects/[I]U_002_GlioSCs_R...
[nfdatautils]@camilla:~/calligrapher@node02:~/nfdata-omics/projects/[I]U_002_GlioSCs_Rhase@...
[nfdatautils]@camilla:~/calligrapher@node02 tree
+- dca_all_samples
|   +- dca_summary
|       +- dca_summary.html
|       +- dca_summary_Type_E-GSC_vs_L-GSC.html
|       +- dca_topable_Type_E-GSC_vs_L-GSC.txt
|       +- dca_topable_Type_E-GSC_vs_L-GSC.html
|       +- enrichr_summary_Type_E-GSC_vs_L-GSC.txt
|       +- enrichr_summary_Type_E-GSC_vs_L-GSC.html
|       +- enrichr_Type_E-GSC_vs_L-GSC_all_top25.txt
|       +- enrichr_Type_E-GSC_vs_L-GSC_all.xlsx
|       +- enrichr_Type_E-GSC_vs_L-GSC_top25.pdf
|       +- enrichr_Type_E-GSC_vs_L-GSC_down_TOP25.txt
|       +- enrichr_Type_E-GSC_vs_L-GSC_top25.pdf
|       +- enrichr_Type_E-GSC_vs_L-GSC_up_top25.txt
|       +- enrichr_Type_E-GSC_vs_L-GSC_up_top25.pdf
|       +- gsea_Type_E-GSC_vs_L-GSC_2.2.all.v2024.1.Hs.symbols.pdf
|       +- gsea_Type_E-GSC_vs_L-GSC_0.5.all.v2024.1.Hs.symbols.pdf
|       +- gsea_Type_E-GSC_vs_L-GSC_0.5.all.v2024.1.Hs.symbols.html
|       +- gsea_Type_E-GSC_vs_L-GSC.xlsx
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_all.xlsx
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.xlsx
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.all.pdf
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.all.xlsx
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.html
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.down.txt
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.down.pdf
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.up.txt
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_top25.xls
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_up.xlsx
|       +- ora_Cp_Type_E-GSC_vs_L-GSC_up.html
|       +- dca2_summary_Type_N5_vs_differentiated.txt
|       +- dca2_topable_Type_N5_vs_differentiated.html
|       +- dca2_topable_Type_N5_vs_differentiated.html
|       +- enrichr_summary_Type_N5_vs_differentiated.txt
|       +- enrichr_summary_Type_N5_vs_differentiated.html
|       +- enrichr_Type_N5_vs_differentiated_all_top25.pdf
|       +- enrichr_Type_N5_vs_differentiated_all.xlsx
|       +- enrichr_Type_N5_vs_differentiated_all.xls
|       +- enrichr_Type_N5_vs_differentiated_down_top25.txt
|       +- enrichr_Type_N5_vs_differentiated_down_top25.xls
|       +- enrichr_Type_N5_vs_differentiated_down_top25.pdf
|       +- enrichr_Type_N5_vs_differentiated_up_top25.txt
|       +- enrichr_Type_N5_vs_differentiated_up.xls
|       +- gsea_Type_N5_vs_differentiated.html.v2024.1.Hs.symbols.pdf
|       +- gsea_Type_N5_vs_differentiated.html.v2024.1.Hs.symbols.html
|       +- gsea_Type_N5_vs_differentiated.html.xls
|       +- gsea_Type_N5_vs_differentiated.html.xlsx
|       +- ora_Cp_Type_N5_vs_differentiated_all.xlsx
|       +- ora_Cp_Type_N5_vs_differentiated_all.xls
|       +- ora_Cp_Type_N5_vs_differentiated_top25.xlsx
|       +- ora_Cp_Type_N5_vs_differentiated_top25.all.pdf
|       +- ora_Cp_Type_N5_vs_differentiated_top25.all.xlsx
|       +- ora_Cp_Type_N5_vs_differentiated_top25.html
|       +- ora_Cp_Type_N5_vs_differentiated_top25.down.txt
|       +- ora_Cp_Type_N5_vs_differentiated_top25.down.pdf
|       +- ora_Cp_Type_N5_vs_differentiated_top25.up.txt
|       +- ora_Cp_Type_N5_vs_differentiated_top25.xls
|       +- ora_Cp_Type_N5_vs_differentiated_up.xlsx
|       +- ora_dca_all_samples.txt
dca_Differential
+- dca_flattening_Yes_vs_No
|   +- dca2_summary_Flattening_Yes_vs_No.html
|   +- dca2_topable_Flattening_Yes_vs_No.html
|   +- dca2_topable_Flattening_Yes_vs_No.txt
|   +- dca2_topable_Flattening_Yes_vs_No.html
|   +- enrichr_Flattening_Yes_vs_No_all_TOP25.pdf
|   +- enrichr_Flattening_Yes_vs_No_all.xlsx
|   +- enrichr_Flattening_Yes_vs_No_all.xls
|   +- enrichr_Flattening_Yes_vs_No_top25.pdf
|   +- enrichr_Flattening_Yes_vs_No_top25.txt
|   +- enrichr_Flattening_Yes_vs_No_down_TOP25.xls
|   +- enrichr_Flattening_Yes_vs_No_down.xls

```



Solution architecture

ETL Pipeline

Currently, nf-core pipeline outputs are **file-based** and **scattered** across directories. I developed an **automated ETL pipeline** transforms flat files into **normalized relational database**

Key Features:

- Dynamic parsing of comparison structures (dea_Type_E-GSC_vs_L-GSC → comparison_variable, comparison1, comparison2)
 - Unified schema for DESeq2 results, enrichment analyses (GSEA/ORA/EnrichR), and metadata
 - Efficient querying with proper indexing and foreign key relationships

Benefits:

- Extremely less error-prone
 - Faster queries
 - Consistent data access patterns
 - Eliminates file path dependencies



Solution architecture

ETL Pipeline

Currently, nf-core pipeline outputs are **file-based** and **scattered** across directories. I developed an **automated ETL pipeline** transforms flat files into **normalized relational database**

Key Features:

- Dynamic parsing of comparison structures (dea_Type_E-GSC_vs_L-GSC → comparison_variable, comparison1, comparison2)
- Unified schema for DESeq2 results, enrichment analyses (GSEA/ORA/EnrichR), and metadata
- Efficient querying with proper indexing and foreign key relationships

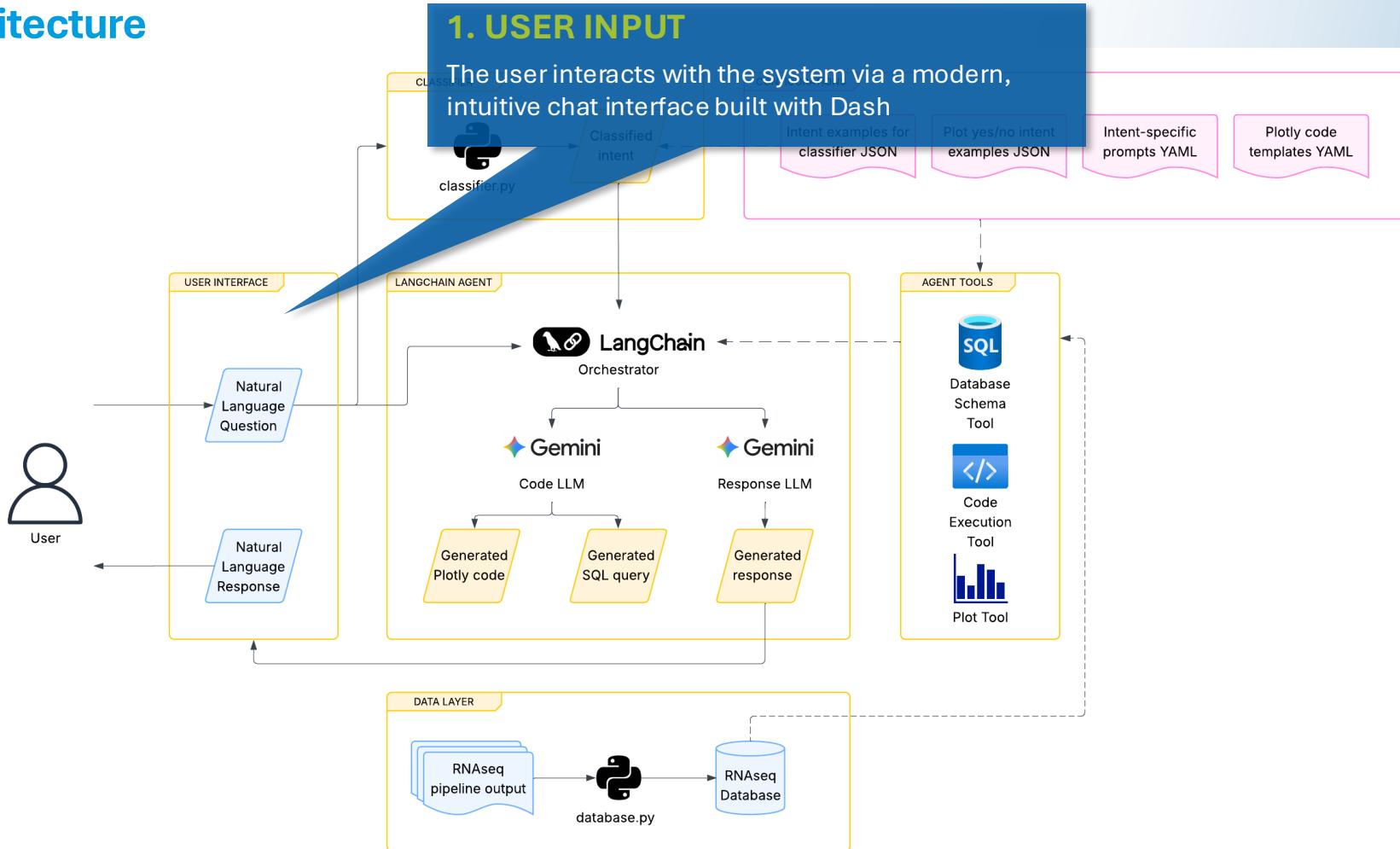
Benefits:

- Extremely less error-prone
- Faster queries
- Consistent data access patterns
- Eliminates file path dependencies



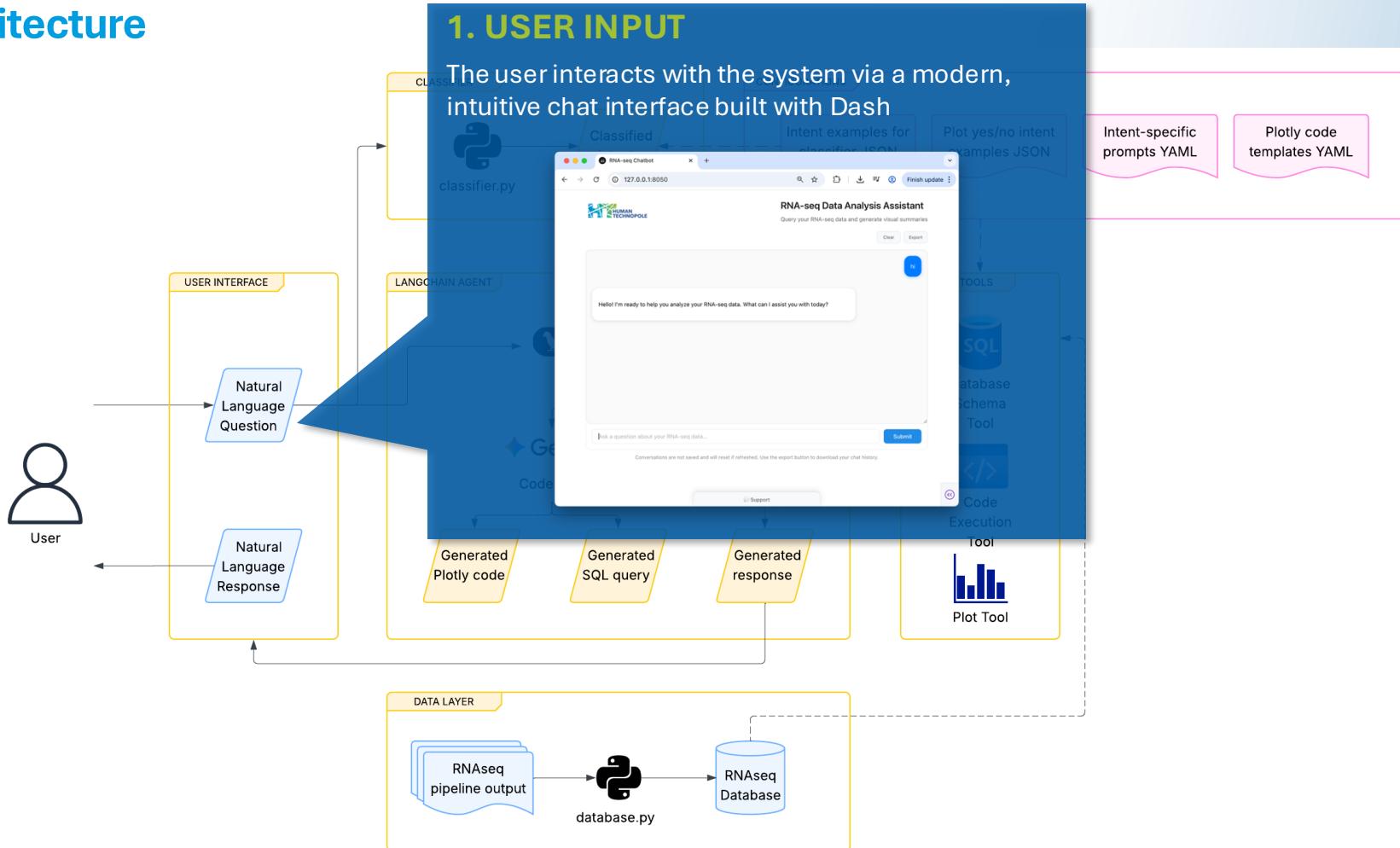
Solution architecture

High Level Architecture



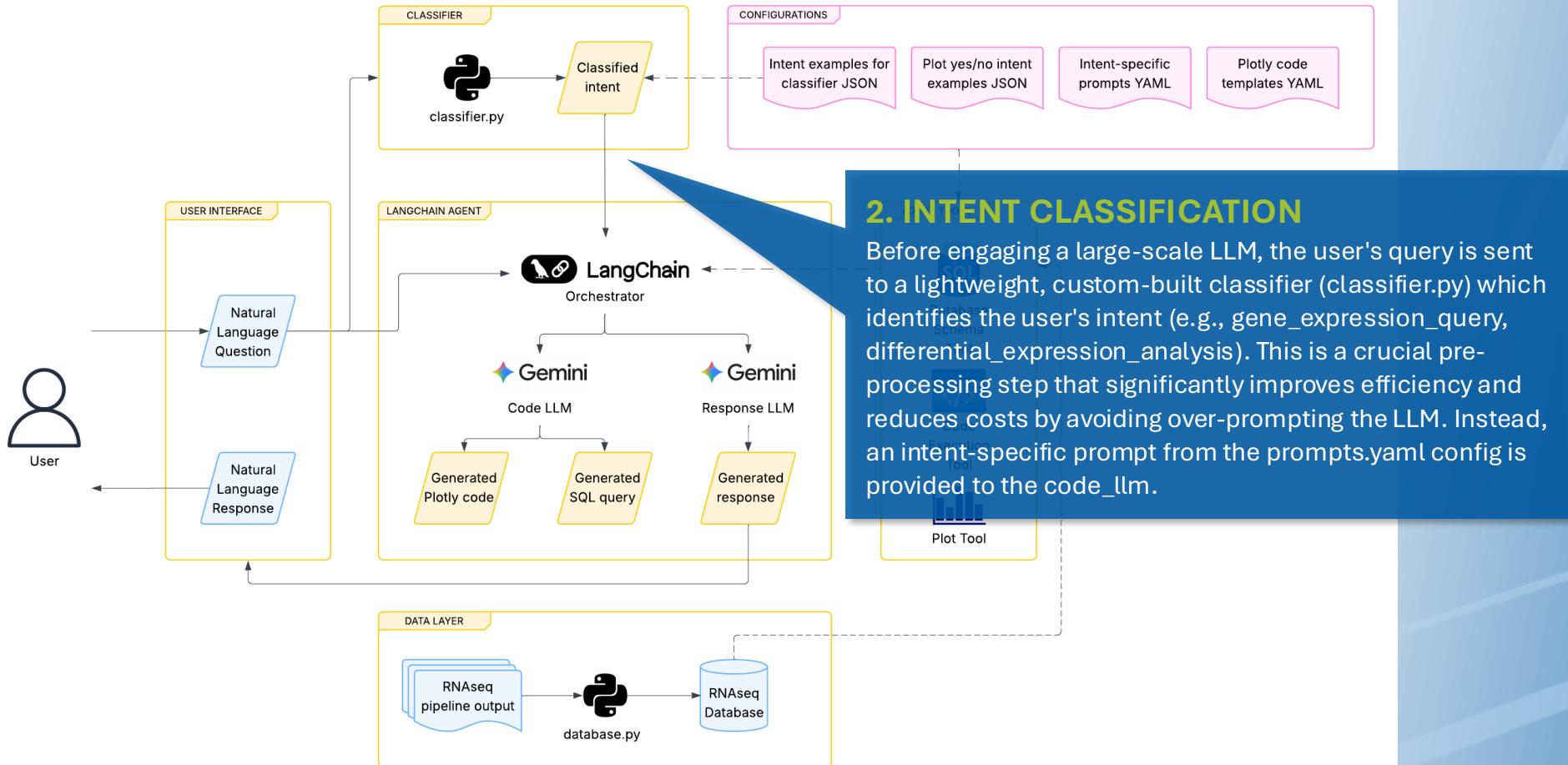
Solution architecture

High Level Architecture



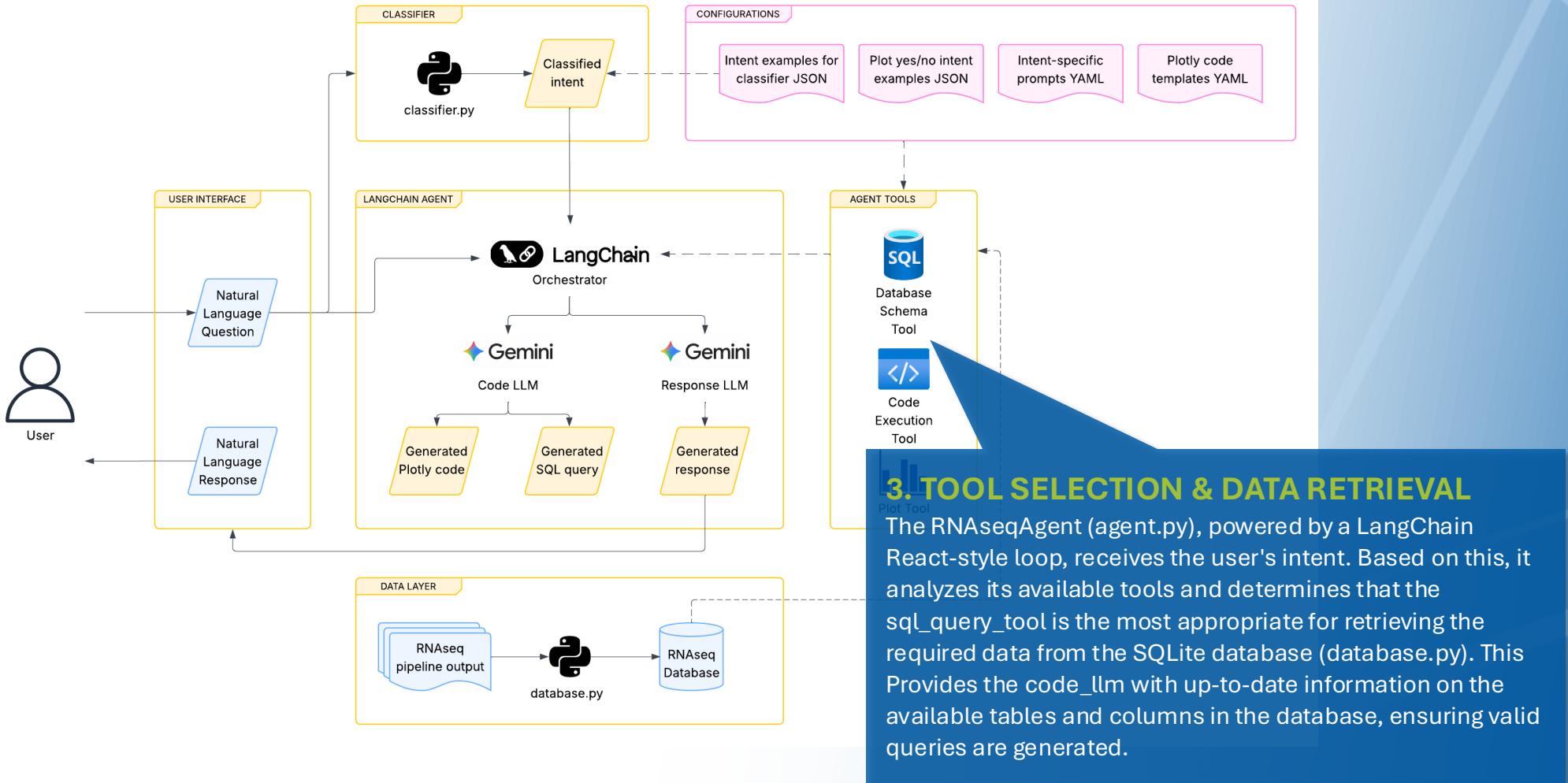
Solution architecture

High Level Architecture



Solution architecture

High Level Architecture

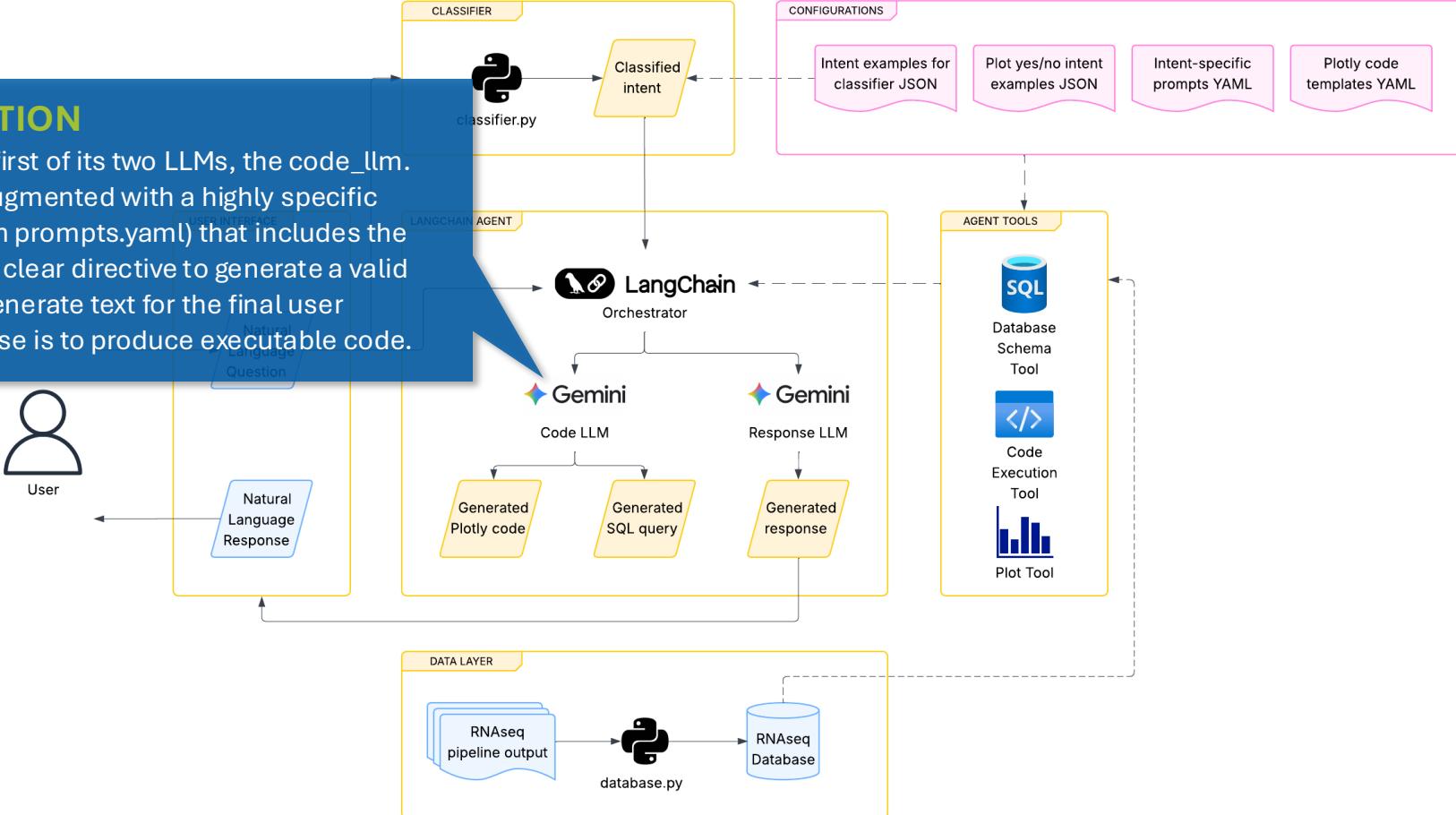


Solution architecture

High Level Architecture

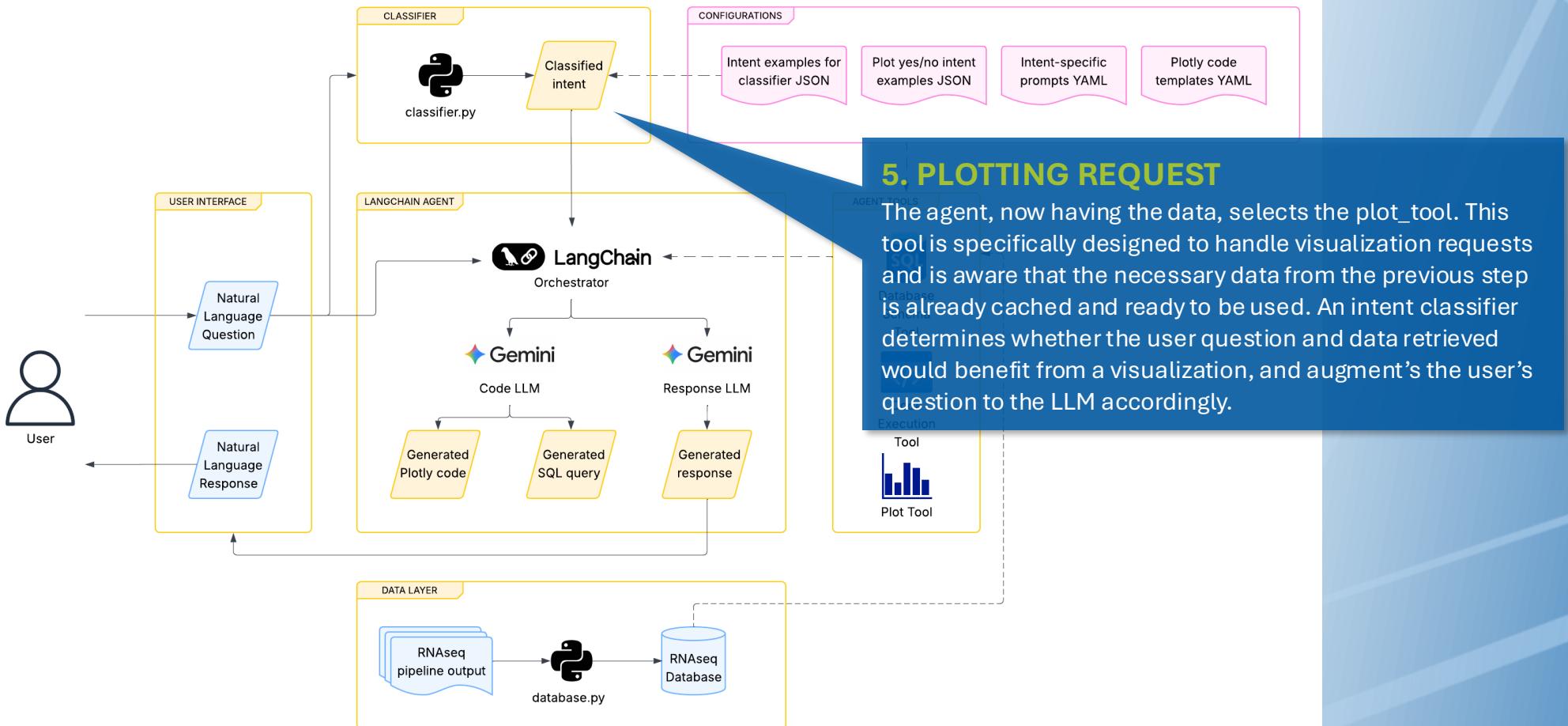
4. CODE GENERATION

The system invokes the first of its two LLMs, the code_llm. The user's question is augmented with a highly specific prompt to its intent (from prompts.yaml) that includes the database schema and a clear directive to generate a valid SQL query. It does not generate text for the final user response; its sole purpose is to produce executable code.



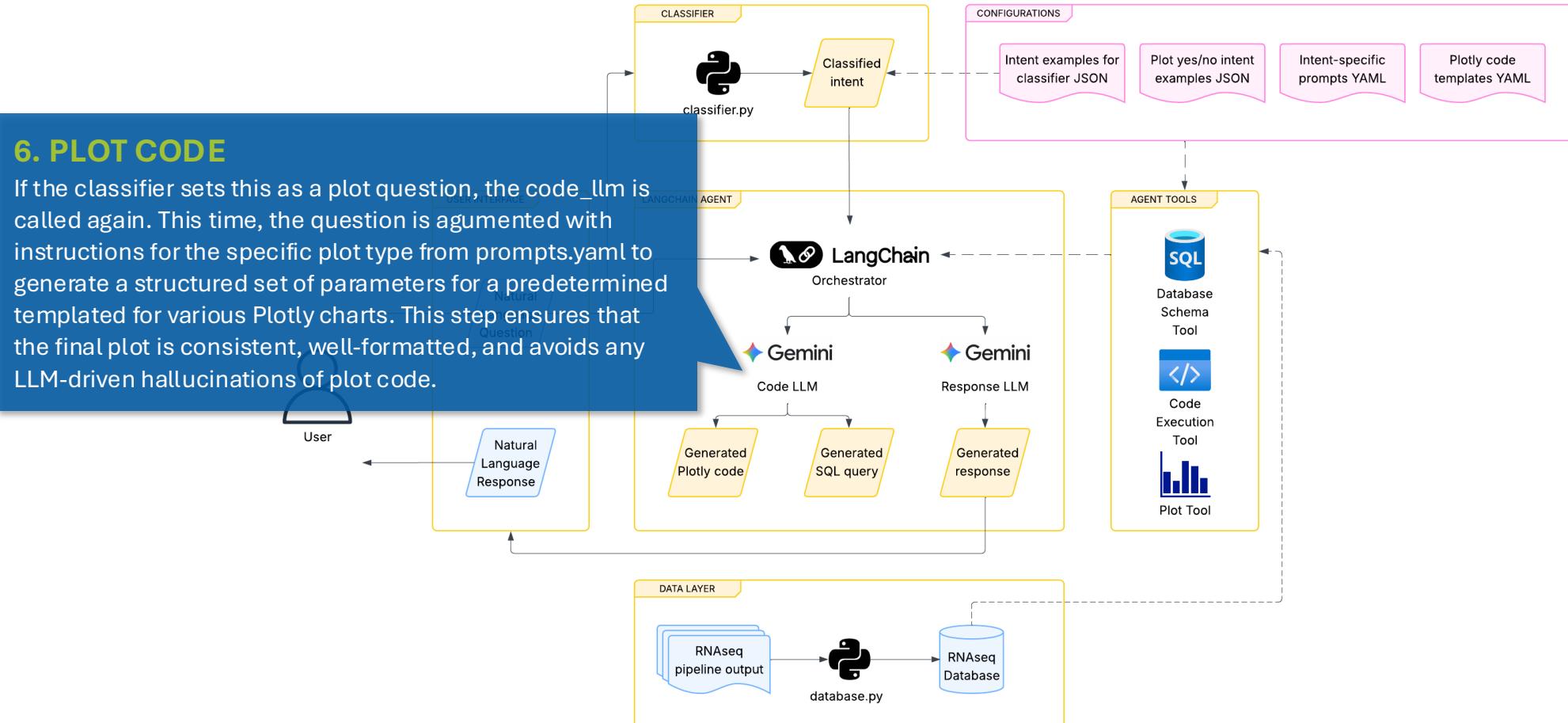
Solution architecture

High Level Architecture



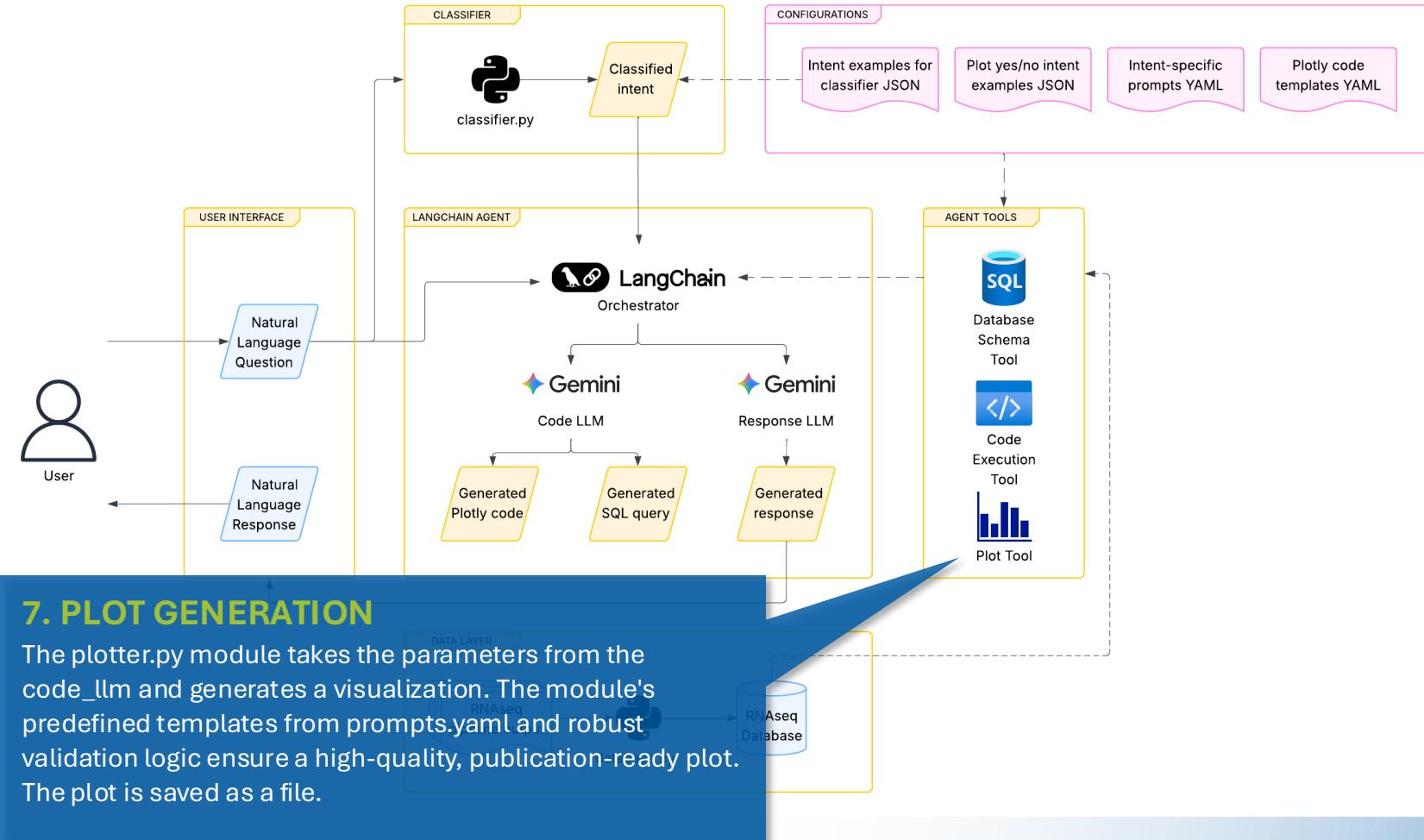
Solution architecture

High Level Architecture



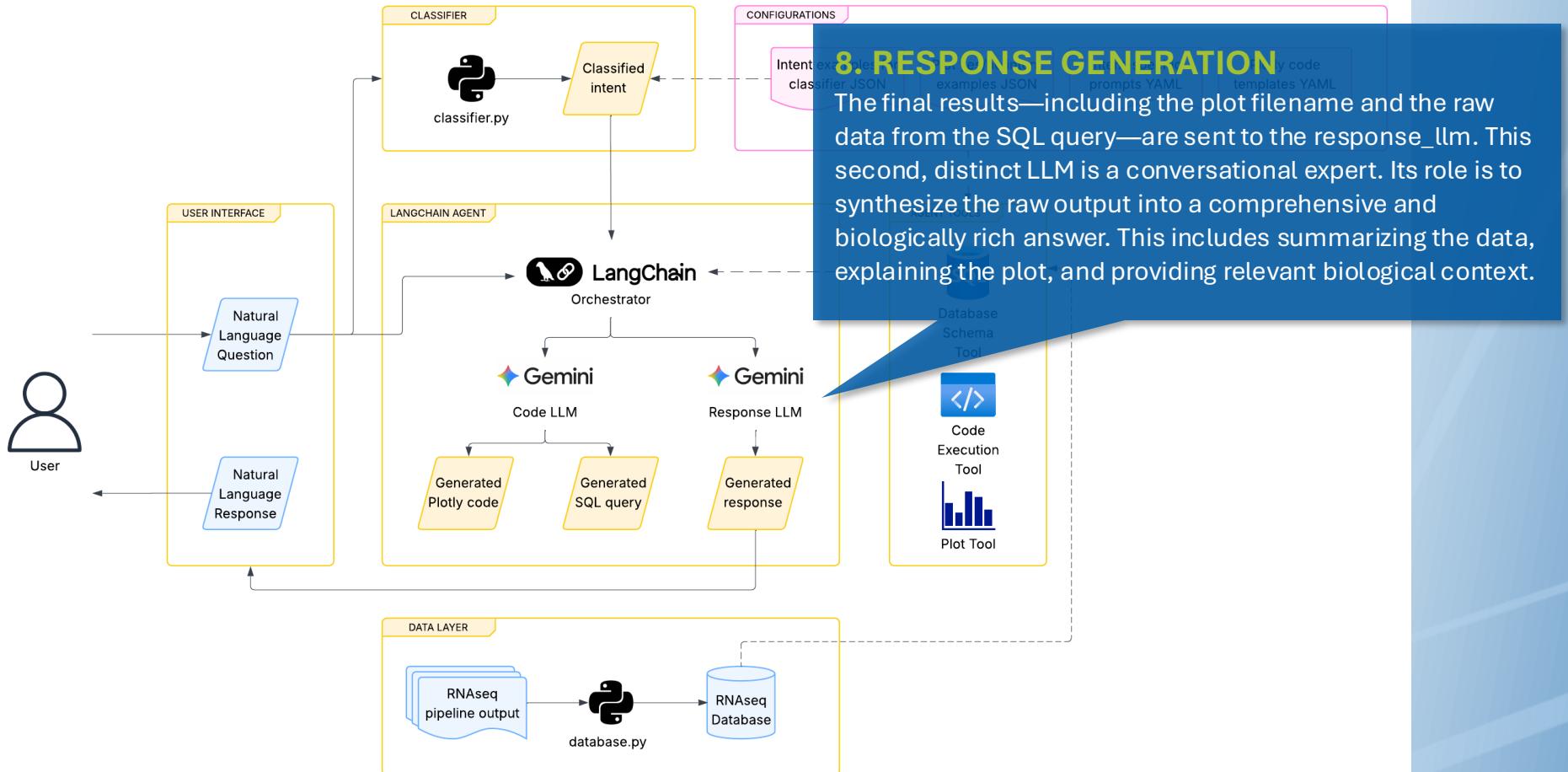
Solution architecture

High Level Architecture



Solution architecture

High Level Architecture

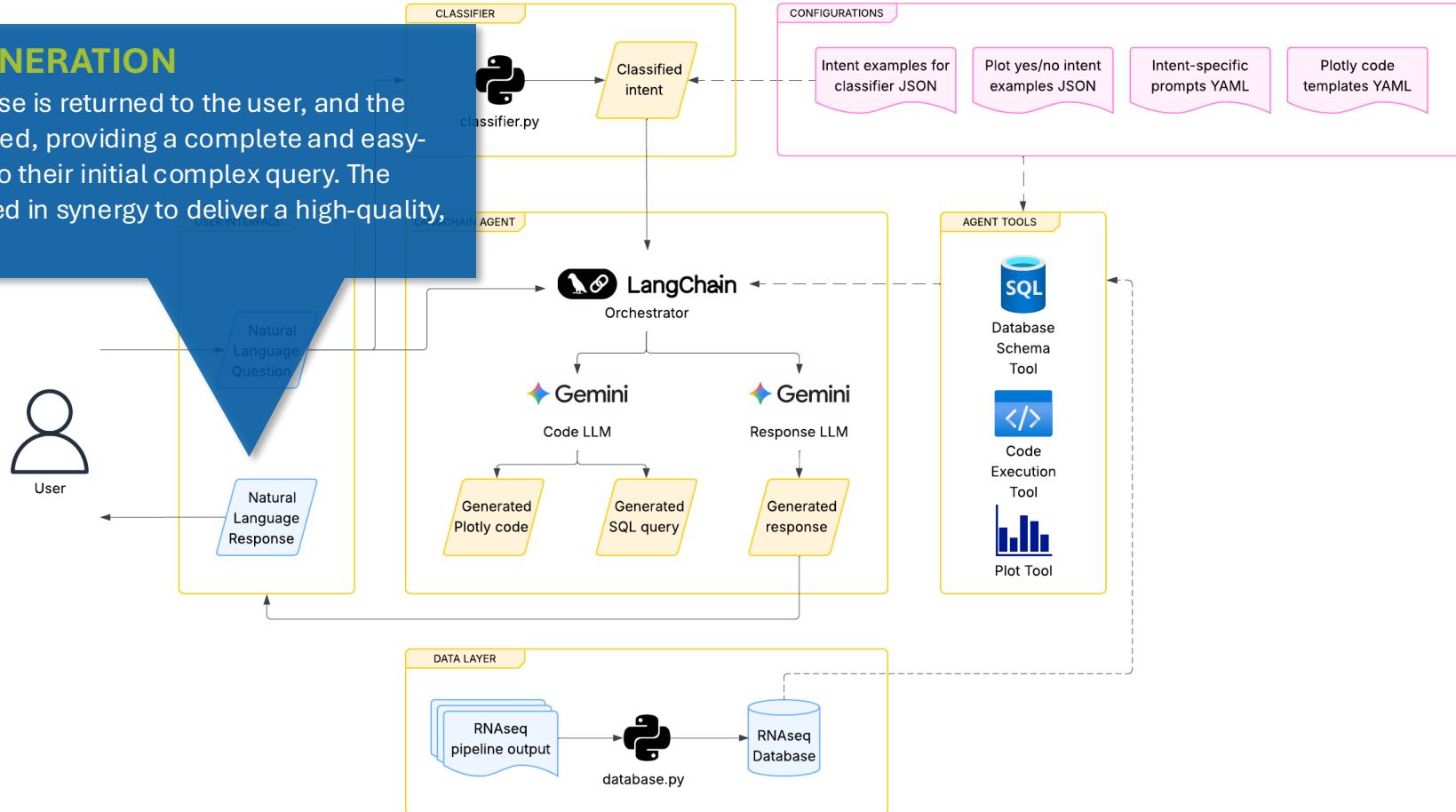


Solution architecture

High Level Architecture

9. RESPONSE GENERATION

The synthesized response is returned to the user, and the generated plot is rendered, providing a complete and easy-to-understand answer to their initial complex query. The entire system has worked in synergy to deliver a high-quality, two-part response



Prompting

Turning Human Intent into Reliable AI Output



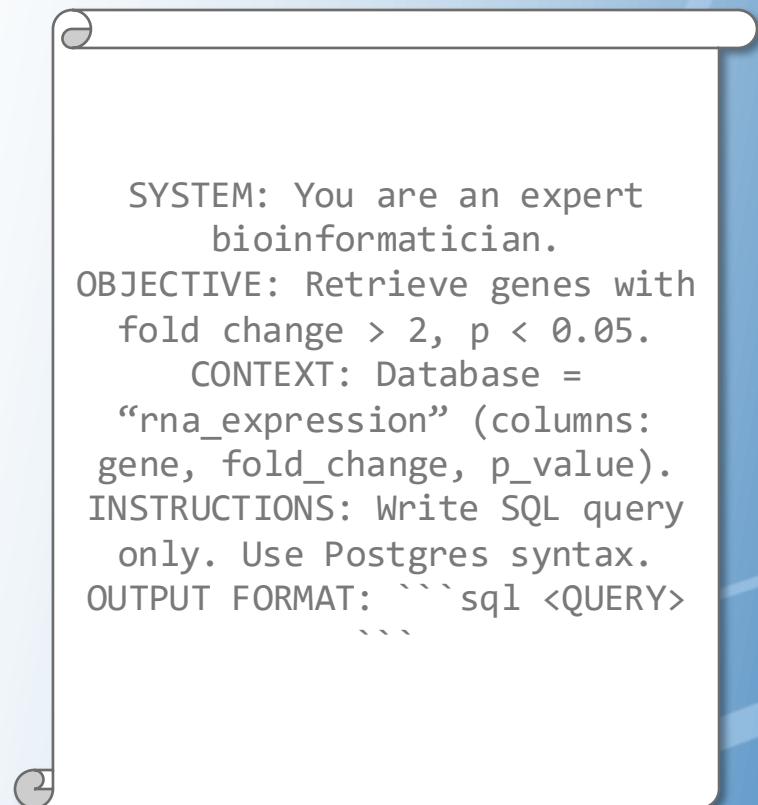
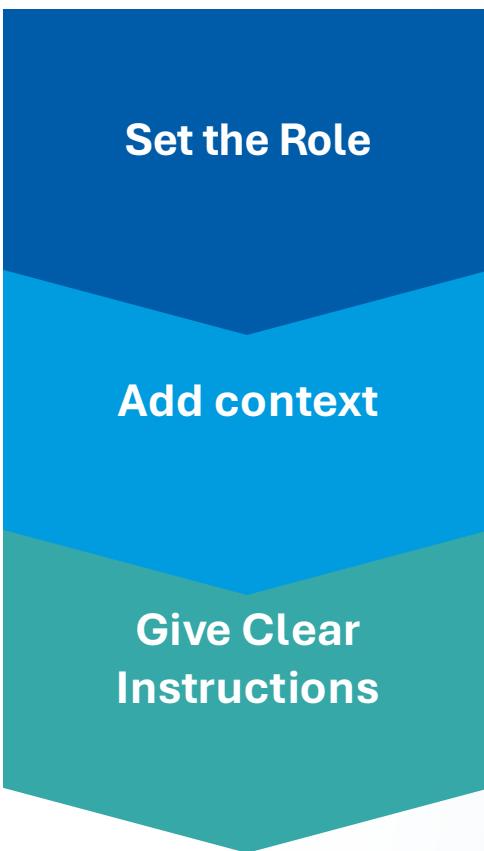
Definition: Prompting is the process of giving structured instructions to LLMs so they produce accurate, useful responses.



Key Role: The prompt is the *interface layer* between human reasoning and machine intelligence.



Impact: Good prompting leads to higher accuracy, lower errors, reproducible outputs.



Data in scope

Data contained is numerous and hierarchical

The chatbot is being designed to **interpret** and **respond** to user queries across **multiple levels** of analysis—from **raw expression patterns** to **advanced functional enrichment**—while maintaining awareness of **sample stratification** (i.e., subsets) and **comparative contexts**.

Specifically, data in scope is:

- Sample metadata
- Normalized counts matrix
- Sample correlation table
- PCA scores
- MDS scores
- Deseq2 results on sample subsets and different comparisons inside each subset
- GSEA results on sample subsets and different comparisons inside each subset, and for different gene lists
- ORA results on sample subsets and different comparisons inside each subset, and for different gene lists

→ *The LLM must understand what data to query based on the naming convention of each SQL table*

Solution Architecture

Error Handling



Code execution errors

This handles **SQL execution failures** due to semantic/logical errors in the SQL query generated by the LLM, and regenerates new, corrected queries, e.g.:

- SQL syntax errors
- Invalid table/column names
- Data type mismatches
- Logic errors in the generated query



Gemini API errors

Handles **LLM call failures** due to infrastructure/service-level problems where the exact same request might succeed if retried, e.g.:

- Rate limiting errors (429 status codes for MistralAI)
- API capacity exceeded errors
- Network connectivity issues
- Temporary service outages

User interface

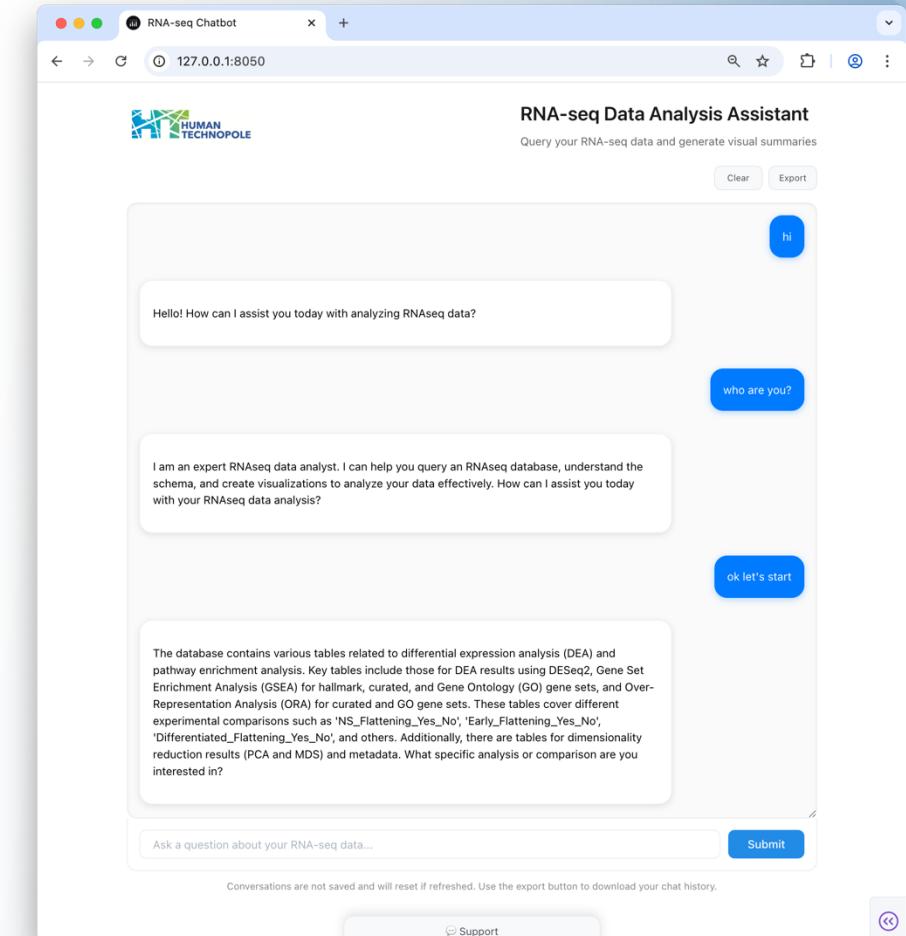
Conversational chat interface

A conversational chat interface was chosen because:

- For an AI chatbot explaining results of a **complex pipeline** like nf-core/rnaseq, **context** and **follow-up** are crucial.
- Researchers will:
 - Often not know exactly **how to phrase** things upfront.
 - Want to **refer back** to previous results.
 - Frequently ask a **series** of related questions (QC → Mapping → Gene Counts → DEGs).

Features:

- Chat bubbles** for input/output.
- Inline plots/images** rendered from handler functions (e.g., PCA, heatmaps).
- Session note** (e.g., “This session is not saved and will reset if refreshed.”).
- Optional: collapsible sections** in output for large results (e.g., DEG tables).
- Scrollable and resizable area** to review chat history and allow clicking to re-execute queries.
- Aptos font** to maintain consistency with Human Technopole tools



- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**

Hello!

Data exploration

The screenshot shows a web browser window titled "RNA-seq Chatbot" at the URL "127.0.0.1:8050". The page is titled "RNA-seq Data Analysis Assistant" and features the Human Technopole logo. A search bar contains the placeholder text "Query your RNA-seq data and generate visual summaries". Below the search bar are "Clear" and "Export" buttons. A blue button labeled "hit tell me about all the data we have" is prominently displayed. The main content area contains a message from the chatbot:

Hi there! I've looked into the data we have available, and it's quite comprehensive for exploring gene expression patterns.

We have a wealth of information from **differential gene expression analyses**, specifically using DESeq2. This means we can identify genes that show significant changes in their expression levels between different experimental conditions or sample groups. The key details we have for these genes include their names, the magnitude and direction of the expression change (`log2FoldChange`), and a measure of statistical significance (adjusted `p-value`). This data is organized by the specific sample groups and the comparisons made between them.

In addition to identifying individual differentially expressed genes, we've also performed **pathway enrichment analyses**. This is a powerful way to understand the biological processes and pathways that are collectively affected by the observed gene expression changes. We have results from different types of pathway analyses, including Gene Ontology, curated gene sets, and hallmark gene sets. For each enriched pathway, we have information on the pathway name, how many genes from our list are involved in that pathway (`Overlap`), and its statistical significance (`P_value`).

To support these analyses, we also have **normalized gene counts**, which represent the expression levels of each gene across all samples after accounting for library size differences. This is crucial for accurate comparisons.

Furthermore, we have **metadata** that describes the samples used in the differential expression analyses, which is essential for understanding the context of the results. We also have **correlation matrices** that show how the expression of genes relates to each other across samples, and **principal component/multidimensional scaling scores** which can help visualize the overall relationships between samples based on their gene expression profiles.

This dataset provides a solid foundation for investigating how gene expression changes in response to your experimental conditions and what biological functions are impacted.

Would you like to dive into any of these specific areas? For example, we could look at the top differentially expressed genes in a particular comparison, or explore which biological pathways are most significantly enriched.

At the bottom of the page, there is a text input field with the placeholder "Ask a question about your RNA-seq data..." and a "Submit" button. A note states "Conversations are not saved and will reset if refreshed. Use the export button to download your chat history." There is also a "Support" link and a small circular icon with a double arrow.

Metadata Questions

Number of replicates

The screenshot shows a web-based RNA-seq Chatbot interface. The title bar reads "RNA-seq Chatbot" and the address bar shows "127.0.0.1:8050". The main content area is titled "RNA-seq Data Analysis Assistant" with the subtitle "Query your RNA-seq data and generate visual summaries". The interface is a conversational chat window.

Initial Question: how many samples do we have

Response: You have a total of 27 unique samples in your dataset. This count was derived from the metadata associated with the multidimensional scaling (MDS) scores. This is a good starting point for our analysis! With 27 samples, we have a solid foundation to explore the differential expression patterns and identify significant biological insights. Would you like to delve into the specific sample groups or comparisons that make up these 27 samples? We can also look at the normalized expression data for these samples to get a sense of the overall gene expression landscape.

Follow-up Question: i want to know what condition these samples are in

Response: It looks like your samples are categorized into several distinct conditions! Based on the data, we can see the following sample types:

- NS: 6 samples
- L-GSC: 6 samples
- L-GSC: 5 samples
- differentiated: 5 samples
- NS-GSC: 3 samples

In addition to these primary classifications, each sample is also linked to a specific Patient ID, a Flattening status (either 'yes' or 'No'), and an experimental Batch (indicated as 'b1' or 'b2'). This information gives us a good overview of the experimental groups and potential sources of variation within your dataset. Would you like to explore how gene expression differs between any of these conditions, or perhaps investigate the impact of flattening or batch effects?

Follow-up Question: how many replicates are there for each condition?

Response: It looks like we have the following number of replicates for each condition:

- L-GSC: 5 replicates
- L-GSC: 6 replicates
- NS: 6 replicates
- NS-GSC: 3 replicates
- differentiated: 6 replicates

This information comes from analyzing the sample types within the dataset. It's great to see a good number of replicates for most conditions, which is crucial for robust statistical analysis. The 'NS-GSC' condition has fewer replicates (3) compared to the others. This difference in sample size might be something to keep in mind when interpreting any differential expression results involving this condition.

Would you like to explore the differential expression results for any of these conditions, or perhaps look at how the samples cluster based on these replicates?

Text Input Field: Ask a question about your RNA-seq data...

Buttons: Support, Submit

Conversations are not saved and will reset if refreshed. Use the export button to download your chat history.

Metadata Questions

Library Size

The screenshot shows a web browser window titled "RNA-seq Chatbot" at the URL "127.0.0.1:8050". The page is titled "RNA-seq Data Analysis Assistant" and features the Human Technopole logo. A blue button labeled "what are the library sizes used" is visible. The main content area displays a list of sample identifiers and their sequencing read counts:

- s17: 72,687,530 reads
- s16: 99,052,770 reads
- s20: 67,195,106 reads
- s23: 63,116,622 reads
- s18: 81,107,366 reads
- s26: 63,057,104 reads
- s19: 82,035,846 reads
- s27: 63,143,726 reads
- s21: 83,422,047 reads
- s22: 75,054,364 reads
- s24: 74,458,823 reads
- s11: 92,922,684 reads
- s2: 100,140,644 reads
- s25: 75,204,220 reads
- s3: 110,250,669 reads

A note explains that these numbers represent the total count of sequencing reads generated for each individual sample. It also states that larger library size generally means more sequencing data, which can increase confidence in detecting even lowly expressed genes and improve accuracy of gene expression quantification.

You'll notice some variation in these library sizes across your samples. This is quite common and can be influenced by factors such as the amount of RNA used for library preparation or the efficiency of the library construction process itself. It's good practice to account for these differences during downstream analysis, often through normalization, to ensure that comparisons between samples are fair and robust.

Would you like to explore how these library sizes might relate to any observed gene expression patterns, or perhaps visualize this distribution?

At the bottom, there is a text input field "Ask a question about your RNA-seq data..." and a "Submit" button. A small note says "Conversations are not saved and will reset if refreshed. Use the export button to download your chat history." There is also a "Support" button and a "Logout" icon.

Metadata Questions

Gene expression distribution

The screenshot shows a web-based RNA-seq Chatbot interface. At the top, it says "RNA-seq Chatbot" and "127.0.0.1:8050". Below that is the "RNA-seq Data Analysis Assistant" logo with the tagline "Query your RNA-seq data and generate visual summaries". A search bar and a "Clear" button are also present.

In the main area, a blue button contains the question: "what is the expression of gapdh in all samples?". The chatbot's response follows:

Certainly! I've examined the expression levels of GAPDH across all your samples.
The normalized expression values for GAPDH in each sample are as follows:

- s17: 2210.43
- s16: 2097.69
- s20: 1448.63
- s23: 1610.53
- s18: 2861.36
- s26: 2871.71
- s19: 2096.03
- s27: 1979.15
- s21: 1480.09
- s22: 1557.34
- s24: 2723.68
- s11: 1917.29
- s2: 1110.37
- s25: 2238.19
- s3: 3428.51

As you can see, s3 exhibits the highest expression of GAPDH with a value of 3428.51, while s2 shows the lowest expression at 1110.37.

GAPDH (Glyceraldehyde-3-phosphate dehydrogenase) is a well-known housekeeping gene, essential for glycolysis. It's often used as a reference for normalizing gene expression data because its levels are generally expected to be stable. While the variations observed here might not be as dramatic as those in genes undergoing differential expression, they can still be biologically meaningful. For instance, differences in cellular metabolic activity or stress responses could influence GAPDH levels. It's always good practice to be aware of these variations, especially when using GAPDH as a normalization control.

Would you like to explore the expression of other housekeeping genes or perhaps visualize these GAPDH expression levels to better understand the distribution across your samples?

At the bottom, there is a text input field "Ask a question about your RNA-seq data...", a "Submit" button, and links for "Conversations are not saved.", "Support", and "Download your chat history."

Metadata Questions

Metadata per condition

The screenshot shows a web-based RNA-seq Chatbot interface. At the top, it says "RNA-seq Chatbot" and "127.0.0.1:8050". Below that is the "RNA-seq Data Analysis Assistant" logo and the text "Query your RNA-seq data and generate visual summaries". There are "Clear" and "Export" buttons.

A message box contains the following text:

Certainly! I've retrieved the metadata associated with different cell types from your dataset.
The data indicates the following samples are associated with each cell type:

- NS (Normal Sample):** This category includes samples s17 (from Patient R006, Batch b2), s21 (from Patient GBM03C, Batch b2), s2 (from Patient R008, Batch b1), and s25 (from Patient GBM05C, Batch b2).
- E-GSC (Embryonic Germ Stem Cell):** Samples in this group are s16 (from Patient R006, Batch b2), s20 (from Patient GBM03C, Batch b2), s24 (from Patient GBM05C, Batch b2), and s11 (from Patient LE13, Batch b1).
- L-GSC (Leukemic Germ Stem Cell):** This group consists of samples s23 (from Patient GBM03C, Batch b2), s19 (from Patient R006, Batch b2), and s27 (from Patient GBM05C, Batch b2).
- differentiated:** Samples categorized as differentiated are s18 (from Patient R006, Batch b2), s26 (from Patient GBM05C, Batch b2), s22 (from Patient GBM03C, Batch b2), and s3 (from Patient R008, Batch b1).

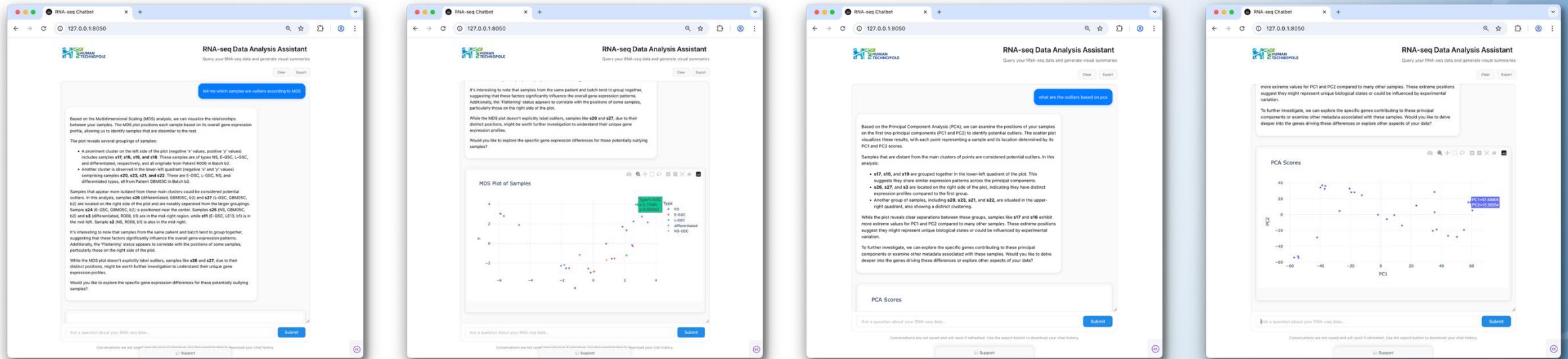
This breakdown provides a clear overview of which samples belong to each cell type, along with their associated patient and batch information. This is crucial for understanding the experimental design and for interpreting any downstream analyses.

Would you like to explore the metadata for a specific cell type in more detail, or perhaps look at other metadata fields?

At the bottom, there's a text input field "Ask a question about your RNA-seq data..." and a "Submit" button. A note says "Conversations are not saved and will reset if refreshed. Use the export button to download your chat history." There's also a "Support" button and a double-left arrow icon.

Correlation Table Questions

Outlier detection



Count Matrix Questions

Most expressed genes

The screenshot shows a web-based RNA-seq Chatbot interface. At the top, it says "RNA-seq Chatbot" and "127.0.0.1:8050". Below that is the "RNA-seq Data Analysis Assistant" logo and the tagline "Query your RNA-seq data and generate visual summaries". There are "Clear" and "Export" buttons.

In the main area, a blue button asks "what are the most expressed genes across all samples?". A response follows:

Certainly! I've analyzed your RNA-seq data to identify the most highly expressed genes across all your samples.

The top ten most expressed genes are:

- ACTB (Beta-actin)
- SPARC (Secreted Protein, Acidic and Rich in Cysteine)
- FTL (Ferritin Light Chain)
- PKM (Pyruvate Kinase, Muscle)
- TMSB4X (Thymosin Beta-4 X-linked)
- MT-RNR2 (Mitochondrial Ribosomal RNA 2)
- SPP1 (Secreted Phosphoprotein 1, also known as osteopontin)
- YWHAZ (Tyrosine 3-monooxygenase/tryptophan 5-monooxygenase activation protein zeta)
- TUBA1A (Tubulin Alpha 1A)
- EIF4G2 (Eukaryotic Translation Initiation Factor 4 Gamma 2)

This list represents genes with the highest overall expression levels when summing their normalized counts across all 27 samples. You'll notice a mix of genes here. For instance, **ACTB** and **TUBA1A** are well-known housekeeping genes involved in maintaining basic cellular structure and function, so their high expression is expected in most cell types.

Other genes like **SPARC** are involved in extracellular matrix remodeling and cell adhesion, **FTL** in iron storage, **PKM** in energy metabolism (glycolysis), and **TMSB4X** in actin dynamics. **MT-RNR2** is a component of the mitochondrial ribosome, essential for protein synthesis within mitochondria. **SPP1** plays roles in cell adhesion and immune responses, while **EIF4G2** is crucial for protein synthesis initiation.

The presence of both fundamental housekeeping genes and genes involved in more specific cellular processes suggests a broad representation of cellular activities within your samples.

Would you like to explore the expression patterns of any of these genes in more detail, perhaps by looking at their individual sample profiles or comparing their expression between specific groups of samples?

At the bottom, there's a text input field "Ask a question about your RNA-seq data..." and a "Submit" button. A note says "Conversations are not saved and will reset if refreshed. Use the export button to download your chat history."

Dimensionality reduction questions

Number of PCs to retain

The screenshot shows a web browser window titled "RNA-seq Chatbot" at the URL "127.0.0.1:8050". The page is titled "RNA-seq Data Analysis Assistant" and features the Human Technopole logo. A blue callout box contains the question: "based on the data, how many principal components would you retain?". Below this, a text box provides an explanation: "To determine how many principal components to retain, we can use the explained variance ratio. The explained variance ratio tells us how much variance is captured by each principal component. Typically, we want to retain enough principal components to explain a significant portion of the variance in the data, such as 80% or 90%." It then lists the explained variance ratios for 10 principal components: PC1: 57.19%, PC2: 54.14%, PC3: 1.51%, PC4: 14.70%, PC5: 2.76%, PC6: 7.00%, PC7: 5.06%, PC8: 14.07%, PC9: 0.08%, and PC10: 0.18%. It notes that the cumulative explained variance for the first two components is approximately 111.33%, which is more than enough to capture the majority of the variance. The text concludes by stating that retaining the first two components means they capture the most important patterns of gene expression variation in the dataset, which could represent major biological processes or experimental conditions.

Dimensionality reduction questions

Plot PCA by Condition and Interpret

The image displays two side-by-side screenshots of a web-based RNA-seq Data Analysis Assistant chatbot. Both screenshots are from the same session, showing a PCA plot and its interpretation.

Left Screenshot:

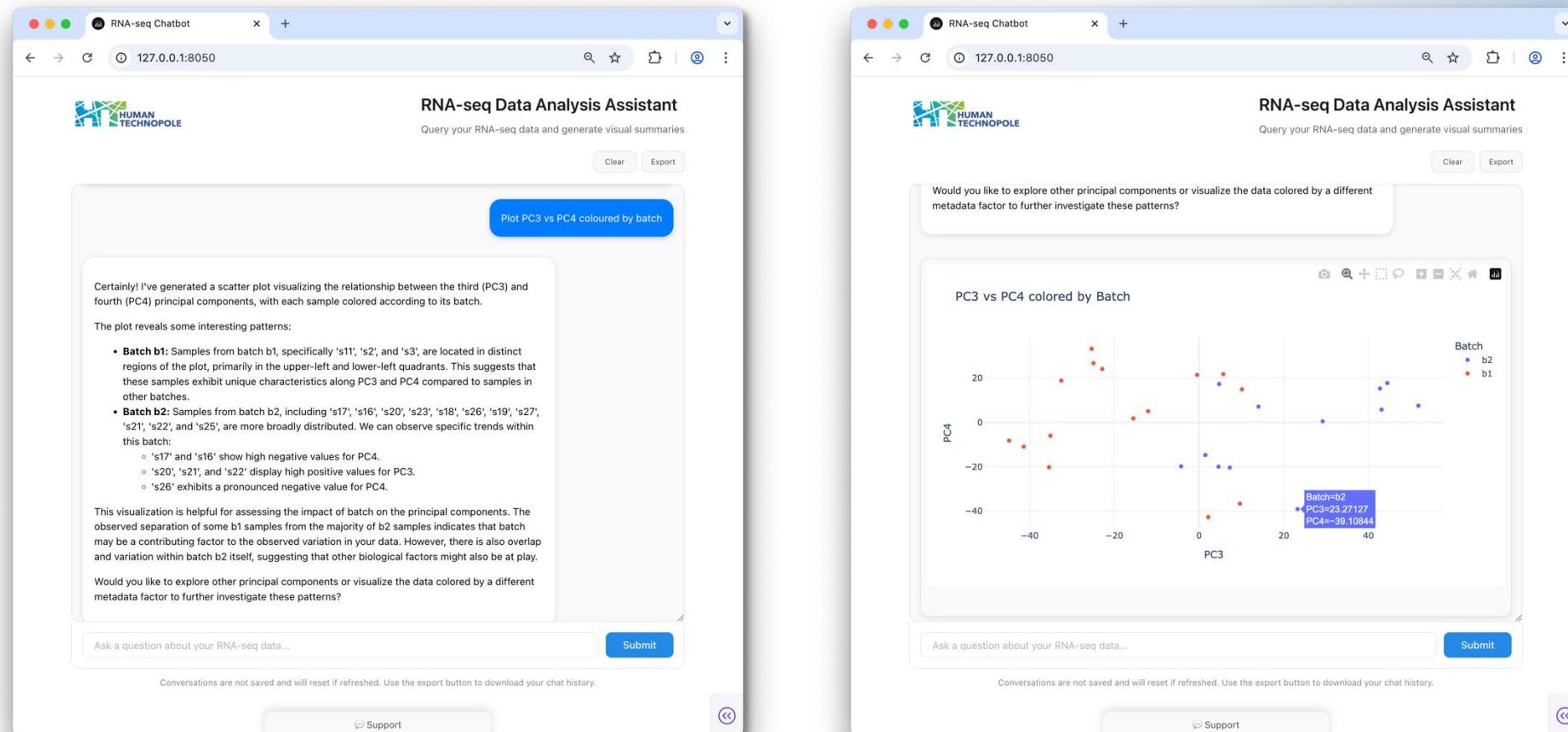
- Header:** RNA-seq Chatbot, 127.0.0.1:8050
- Title:** RNA-seq Data Analysis Assistant
- Text:** "Hello! I've generated a Principal Component Analysis (PCA) plot for your RNA-seq data, colored by patient. This visualization helps us understand the major sources of variation in your samples."
The PCA plot shows how your samples are distributed in a reduced dimensional space, with the first two principal components (PC1 and PC2) explaining the largest proportions of the variance in the data. By coloring the points according to the 'Patient' metadata, we can observe how samples from the same patient group together.
Interpretation of the PCA Plot:
Looking at the plot, we can see that samples from the same patient generally cluster together. For instance, samples 's17', 's16', and 's19' all originate from patient 'R006' and are located in close proximity on the plot. Similarly, samples 's20', 's23', and 's21' from patient 'GBM03C' show a tendency to group together. This clustering by patient indicates that the individual patient is a significant factor influencing the gene expression profiles observed in your dataset. The biological or technical characteristics unique to each patient are driving a substantial portion of the observed variability.
While patient of origin appears to be a strong driver of variation, it's also worth noting the distribution of samples within each patient group. For example, within patient 'GBM03C', samples 's20', 's23', and 's21' are relatively close, but there's still some spread. This suggests that other factors might also be contributing to the differences in gene expression.
To further explore these patterns, we can examine other principal components or investigate how other metadata variables, such as 'Type', 'Flattening', or 'Batch', might influence sample clustering. This could reveal additional biological insights or potential technical confounders.
The PCA plot is available for your review at: [plots/scatter_09_10_00_08_36.html](#). Please let me know if you'd like to explore any specific aspects of this plot or investigate other dimensions of variation in your data!"
- Buttons:** Clear, Export, Submit, Support

Right Screenshot:

- Header:** RNA-seq Chatbot, 127.0.0.1:8050
- Title:** RNA-seq Data Analysis Assistant
- Text:** "other metadata variables, such as 'Type', 'Flattening', or 'Batch', might influence sample clustering. This could reveal additional biological insights or potential technical confounders.
The PCA plot is available for your review at: [plots/scatter_09_10_00_08_36.html](#). Please let me know if you'd like to explore any specific aspects of this plot or investigate other dimensions of variation in your data!"
- Figure:** PCA Plot Colored by Patient
- Legend:** Patient
R006 (blue dot)
GBM03C (red dot)
GBM05C (green dot)
LE13 (purple dot)
R008 (orange dot)
R004 (cyan dot)
- Buttons:** Ask a question about your RNA-seq data..., Submit, Support

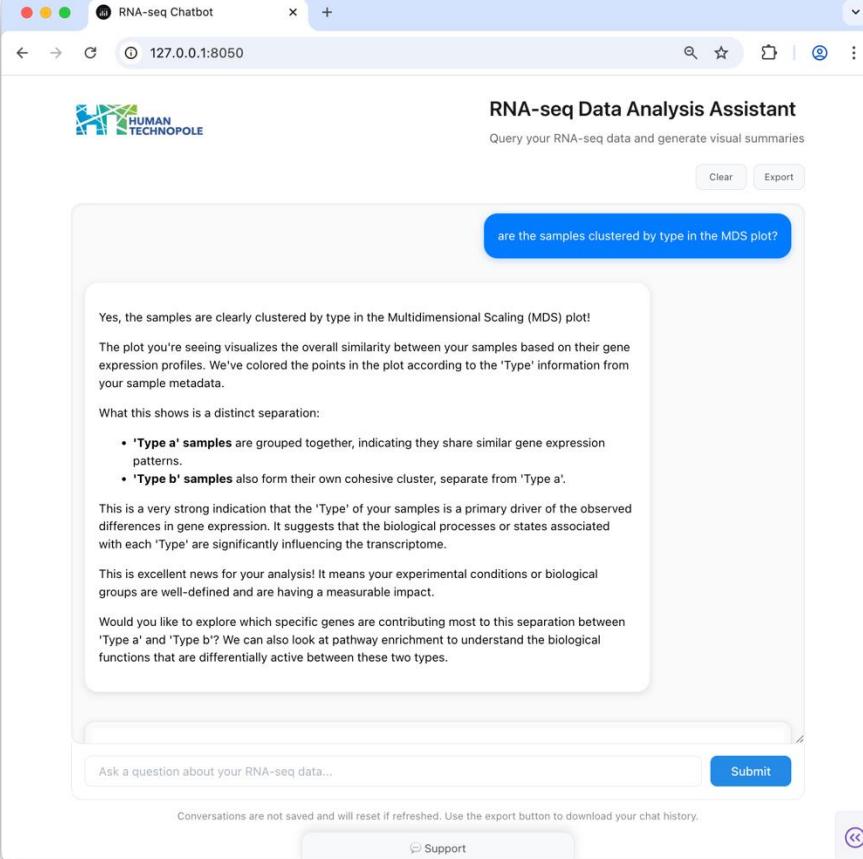
Dimensionality reduction questions

Plot PC3 vs PC4 coloured by batch

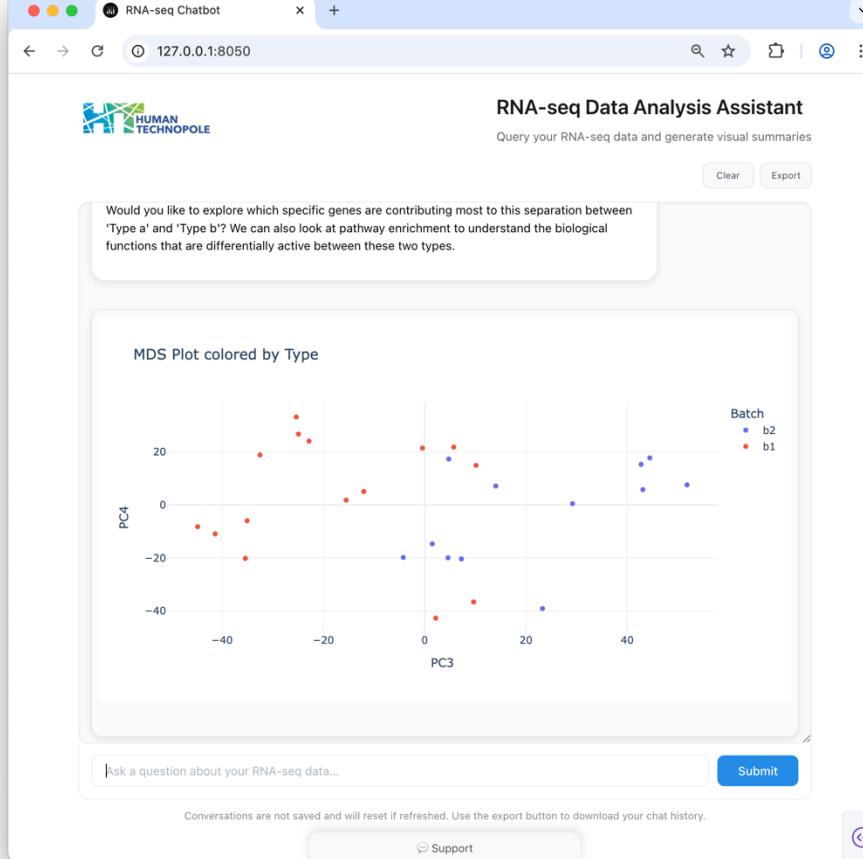


Count Matrix Questions

MDS Clustering by Condition



The screenshot shows the RNA-seq Data Analysis Assistant interface. A blue button at the top asks, "are the samples clustered by type in the MDS plot?". Below it, a message box states: "Yes, the samples are clearly clustered by type in the Multidimensional Scaling (MDS) plot! The plot you're seeing visualizes the overall similarity between your samples based on their gene expression profiles. We've colored the points in the plot according to the 'Type' information from your sample metadata. What this shows is a distinct separation: • 'Type a' samples are grouped together, indicating they share similar gene expression patterns. • 'Type b' samples also form their own cohesive cluster, separate from 'Type a'. This is a very strong indication that the 'Type' of your samples is a primary driver of the observed differences in gene expression. It suggests that the biological processes or states associated with each 'Type' are significantly influencing the transcriptome. This is excellent news for your analysis! It means your experimental conditions or biological groups are well-defined and are having a measurable impact. Would you like to explore which specific genes are contributing most to this separation between 'Type a' and 'Type b'? We can also look at pathway enrichment to understand the biological functions that are differentially active between these two types." At the bottom, there's a text input field "Ask a question about your RNA-seq data..." and a "Submit" button.



The screenshot shows the same RNA-seq Data Analysis Assistant interface. A message box asks, "Would you like to explore which specific genes are contributing most to this separation between 'Type a' and 'Type b'? We can also look at pathway enrichment to understand the biological functions that are differentially active between these two types." Below it is a scatter plot titled "MDS Plot colored by Type". The plot shows points representing samples, color-coded by batch (blue for b2, red for b1). The x-axis is labeled "PC3" and ranges from -40 to 40. The y-axis is labeled "PC4" and ranges from -40 to 20. The plot clearly separates two clusters of points corresponding to the two types mentioned in the message box above.

Differential Expression Analysis

Differentially Expressed Genes in Given Experiment

The image displays two side-by-side screenshots of the "RNA-seq Chatbot" application, version 127.0.0.1:8050. Both screenshots feature the "HUMAN TECHNOPOLE" logo at the top left.

Screenshot 1 (Left): This screenshot shows the initial response to a query about differentially expressed genes. The message box contains:

what genes are differentially expressed in the Type_NS_differentiated comparison and can you plot them?

Certainly! I've analyzed the differential gene expression data for the 'Type_NS_differentiated' comparison.

We identified **10,170 genes** that are significantly differentially expressed. These genes meet the criteria of having an adjusted p-value (*padj*) less than 0.05 and an absolute log₂ fold change greater than 1, indicating a robust change in expression.

Here are some of the top differentially expressed genes:

- TPM1 shows a significant downregulation with a log₂ fold change of -3.74.
- TALON1 is also strongly downregulated, with a log₂ fold change of -5.36.
- EDN1 exhibits a very pronounced downregulation, with a log₂ fold change of -10.56.
- On the upregulation side, HMGCS1 shows an increase in expression with a log₂ fold change of 2.25.
- SREBF2 is also upregulated, with a log₂ fold change of 1.19.

To help visualize these findings, I've generated a volcano plot. This plot displays the magnitude of the expression change (log₂ fold change) against the statistical significance (adjusted p-value) for each gene. Genes that are significantly upregulated are typically shown in red, downregulated genes in blue, and genes with non-significant changes in gray. You can find the plot here: [plots/volcano_09_10_00_03_54.html](#).

Would you like to explore the functions or pathways associated with these differentially expressed genes, or perhaps focus on a specific subset of these genes?

Screenshot 2 (Right): This screenshot shows the response after the user has viewed the initial results. The message box contains:

change of 2.25.
• SREBF2 is also upregulated, with a log₂ fold change of 1.19.

To help visualize these findings, I've generated a volcano plot. This plot displays the magnitude of the expression change (log₂ fold change) against the statistical significance (adjusted p-value) for each gene. Genes that are significantly upregulated are typically shown in red, downregulated genes in blue, and genes with non-significant changes in gray. You can find the plot here: [plots/volcano_09_10_00_03_54.html](#).

Would you like to explore the functions or pathways associated with these differentially expressed genes, or perhaps focus on a specific subset of these genes?

Volcano Plot of Type_NS_differentiated Comparison: A scatter plot showing Log2 Fold Change on the x-axis (ranging from -15 to 10) and -Log10 P-value on the y-axis (ranging from 0 to 40). The plot is divided into four quadrants by dashed lines at Log2 Fold Change = 0 and -Log10 P-value = 0.5. Blue dots represent downregulated genes, and red dots represent upregulated genes. A legend indicates:

- Significance
- Downregulated (blue)
- Upregulated (red)

A callout box highlights a red dot at approximately (Log2 Fold Change: 5.5, -Log10 P-value: 35), labeled:

Significance: Upregulated
Log2 Fold Change: 5.5
Upregulated
gene_name: COL2A1
p=0.05

At the bottom of both screenshots, there is a text input field with placeholder text "Ask a question about your RNA-seq data..." and a "Submit" button. Below the input field, a note says "Conversations are not saved and will reset if refreshed. Use the export button to download your chat history." There is also a "Support" link at the bottom.

Differential Expression Analysis

Comparison with most differentially expressed genes

The image displays two side-by-side screenshots of the "RNA-seq Chatbot" application, version 127.0.0.1:8050. Both screenshots feature a "HUMAN TECHNOPOLE" logo at the top left.

Screenshot 1 (Left): RNA-seq Data Analysis Assistant

Text in the main area:

what genes are differentially expressed in the Type_NS_differentiated comparison and can you plot them?

Response text:

Certainly! I've analyzed the differential gene expression data for the 'Type_NS_differentiated' comparison.
We identified **10,170 genes** that are significantly differentially expressed. These genes meet the criteria of having an adjusted p-value (padj) less than 0.05 and an absolute log2 fold change greater than 1, indicating a robust change in expression.

Top differentially expressed genes:

- TPM1 shows a significant downregulation with a log2 fold change of -3.74.
- TALON1 is also strongly downregulated, with a log2 fold change of -5.36.
- EDN1 exhibits a very pronounced downregulation, with a log2 fold change of -10.56.
- On the upregulation side, HMGCS1 shows an increase in expression with a log2 fold change of 2.25.
- SREBF2 is also upregulated, with a log2 fold change of 1.19.

To help visualize these findings, I've generated a volcano plot. This plot displays the magnitude of the expression change (log2 fold change) against the statistical significance (adjusted p-value) for each gene. Genes that are significantly upregulated are typically shown in red, downregulated genes in blue, and genes with non-significant changes in gray. You can find the plot here: [plots/volcano_09_10_00_03_54.html](#).

Would you like to explore the functions or pathways associated with these differentially expressed genes, or perhaps focus on a specific subset of these genes?

Volcano Plot of Type_NS_differentiated Comparison

Ask a question about your RNA-seq data... Submit

Conversations are not saved and will reset if refreshed. Use the export button to download your chat history.

Screenshot 2 (Right): RNA-seq Data Analysis Assistant

Text in the main area:

change of 2.25.
• SREBF2 is also upregulated, with a log2 fold change of 1.19.

Response text:

To help visualize these findings, I've generated a volcano plot. This plot displays the magnitude of the expression change (log2 fold change) against the statistical significance (adjusted p-value) for each gene. Genes that are significantly upregulated are typically shown in red, downregulated genes in blue, and genes with non-significant changes in gray. You can find the plot here: [plots/volcano_09_10_00_03_54.html](#).

Would you like to explore the functions or pathways associated with these differentially expressed genes, or perhaps focus on a specific subset of these genes?

Volcano Plot of Type_NS_differentiated Comparison

-Log10 P-value

Significance

- Downregulated (blue)
- Upregulated (red)

Log2 Fold Change

Significance: Upregulated
Log2 Fold Change: 5.59092
Upregulated gene name: COL2A1
p=0.05

Ask a question about your RNA-seq data... Submit

Conversations are not saved and will reset if refreshed. Use the export button to download your chat history.

Pathway Analysis Questions

Gene Set Enrichment Analysis

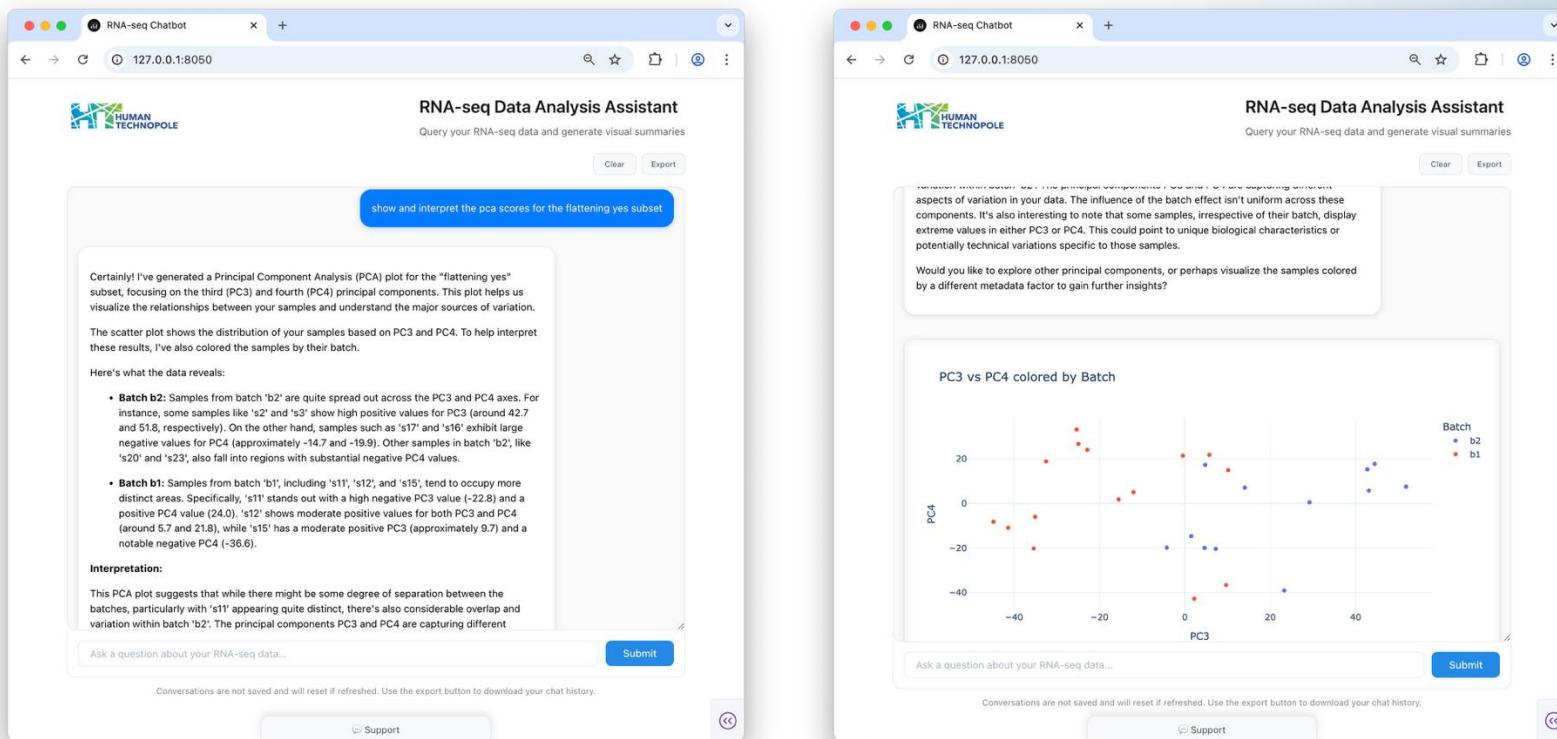
The screenshot shows a web browser window titled "RNA-seq Chatbot" at the URL "127.0.0.1:8050". The page is titled "RNA-seq Data Analysis Assistant" and features the Human Technopole logo. A search bar contains the placeholder text "Query your RNA-seq data and generate visual summaries". Below the search bar are "Clear" and "Export" buttons. A blue button contains the question: "what are the top gsea pathways enriched in the Type_NS_differentiated comparison?". The main content area displays a response: "Certainly! Based on the analysis of the 'Type_NS_differentiated' comparison, here are the top Gene Set Enrichment Analysis (GSEA) pathways that show significant enrichment: The most highly enriched pathway is **Epithelial Mesenchymal Transition**, with a very strong statistical significance (adjusted p-value of 3.43e-20) and a substantial overlap of 65 genes out of 200 in the gene set. This suggests a major biological shift towards a more migratory and invasive cell state. Following closely, we see **Estrogen Response Early** (adjusted p-value 3.82e-09, 47/200 genes) and **UV Response Dn** (adjusted p-value 4.15e-09, 38/144 genes). The enrichment of estrogen response pathways could indicate a sensitivity to or activation of estrogen signaling, while the "UV Response Dn" might suggest a decreased cellular response to UV damage. Additionally, the **Collagen-Containing Extracellular Matrix (GO:0062023)** pathway is significantly enriched (adjusted p-value 6.02e-09, 73/373 genes), pointing to substantial changes in the extracellular environment and its components. Finally, **Hypoxia** is also among the top enriched pathways (adjusted p-value 6.52e-08, 44/200 genes), indicating that cells in this comparison may be experiencing or responding to low oxygen conditions. These findings highlight key biological processes that are significantly altered in the 'Type_NS_differentiated' comparison. The enrichment of pathways like Epithelial Mesenchymal Transition and Hypoxia can have profound implications for cell behavior, such as motility, invasiveness, and adaptation to stressful environments. Would you like to explore the specific genes involved in any of these top pathways, or perhaps investigate other comparisons?" At the bottom, there is a text input field "Ask a question about your RNA-seq data..." and a "Submit" button. A note at the bottom states: "Conversations are not saved and will reset if refreshed. Use the export button to download your chat history." A "Support" button is also present.

- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**

Corner Cases

Vague Questions

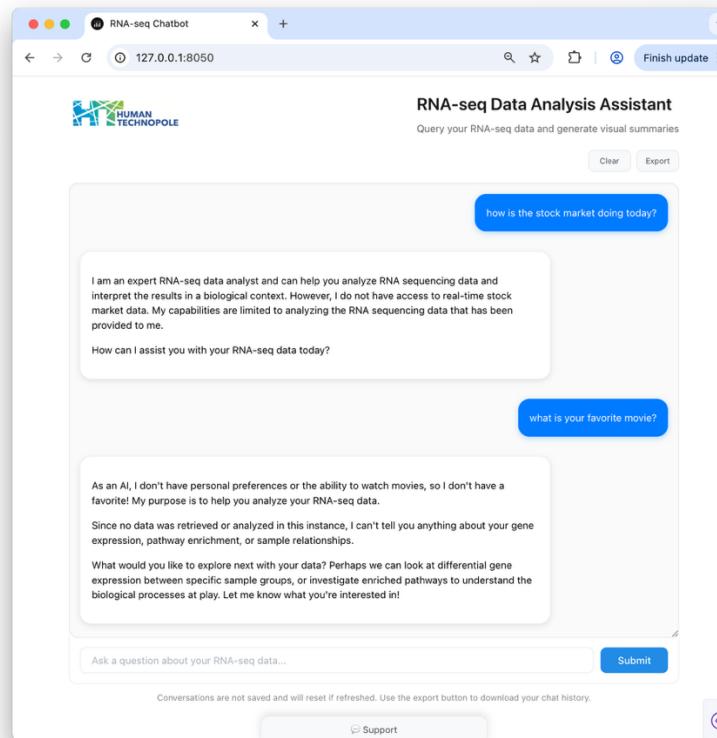
The agent tries to extrapolate meaningful insights.



Corner Cases

Off-Topic Questions

The agent does not answer and politely restates its role.



- 1. Project Motivation**
- 2. Introduction to Large Language Models (LLMs)**
- 3. Explored Approaches & Their Limitations**
- 4. Solution Architecture**
- 5. Use Cases**
- 6. Corner Cases**
- 7. Next Steps**

Next Steps

Road to Deployment and Client Integration



Implementing a Relational Data Architecture for nf-core outputs:

- We should establish a **consistent nomenclature** and a **predictable directory hierarchy** for automated discovery
- Given that an ETL pipeline is now in place, should **standardize nf-core pipeline outputs** for reliable ETL



Collaboration with IU1 for seamless report integration:

- **Containerization** (Docker/Kubernetes) for scalable deployment
- **API endpoint** integration for real-time data access
- **Automated report generation** pipeline integration



Multi-Omics Technology Scaling, next: WGS/WES, by adapting:

- **Intent Classifier**: Expand training data with variant calling, CNV, and mutation queries
- **Prompt Engineering**: Add WES/WGS-specific analysis patterns and genomics terminology
- **Visualization Prompting**: Implement Manhattan plots, circos plots, variant density plots, and genome browsers

Sources & Further Reading

- <https://community.aws/content/2ZVa61RxToXUFzcuY8Hbut6L150/what-is-an-instruct-model?lang=en>
- <https://mistral.ai/news/announcing-mistral-7b>
- https://towardsdatascience.com/integrating-llm-agents-with-langchain-into-vica-d18a5c8583c6/?utm_source=chatgpt.com
- <https://python.langchain.com/docs/concepts/>
- <https://www.ibm.com/think/topics/react-agent>