A Biomedical Benchmark and Modular RAG Framework for Text2Cypher

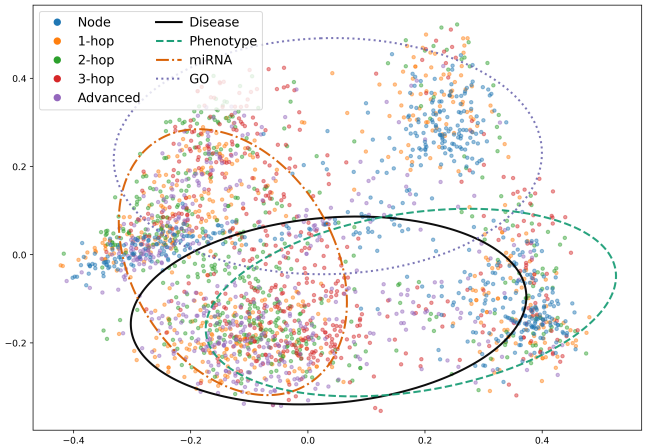
Supplementary Material

Emanuele Cavalleri, Nada Bou Kanaan, Marco Mesiti

This Supplementary Material is organized as follows. Section S1 provides additional details of *bio2C*, focusing on the semantic distribution of its questions in the embedding space. Section S2 presents representative biomedical queries from each dataset category and an example of the prompt generated by the `S+RAG+O` pipeline. Section S3 reports detailed results across the five *bio2C* query categories and compares the behavior of GPT- and LLaMA-based models. Finally, Section S4 presents the complete results obtained on the manually curated MedT2C subset.

S1 *bio2C* Analysis

The five query categories were designed to progressively expose models to increasingly expressive fragments of the Cypher language. Besides graph traversals of increasing depth (single-node retrieval, one-hop, two-hop, and three-hop navigations), the benchmark covers a broad range of filtering conditions, logical operators, aggregation functions, and collection-processing constructs. Filtering predicates include comparisons over node and relationship properties (e.g., `n.p op val`, with $op \in \{=, <, >, \neq\}$), relationship and node type predicates (e.g., `TYPE(r)` and `LABELS(n)`), string matching operators (e.g., `STARTS WITH`, `ENDS WITH`, `CONTAINS`, and regular-expression matching), and boolean combinations through `AND`, `OR`, and `NOT`. Queries also exploit collection-oriented operators, including `COUNT`, `SIZE`, `ANY`, and `ALL`, to express constraints over sets of values. The most advanced category further introduces analytical and multi-stage query constructs through the `WITH` clause, enabling aggregation functions such as `COUNT`, `MAX`, `MIN`, `AVG`, and top-k ranking. In addition, the benchmark includes Cypher operators for collection manipulation and conditional expressions, including `COLLECT`, `REDUCE`, and `CASE`. Together, these constructs cover the vast majority of Cypher features required to formulate realistic analytical queries over bioKGs. To facilitate manual verification, every query was designed to return fewer than 30 answers. All Cypher queries were executed against the RNA-KG view to verify both syntactic validity and semantic correctness before inclusion in the benchmark.



Supp. Fig. S1. PCA projection of the dataset embeddings.

Figure S2 compares the distribution of node labels and relationship types in the RNA-KG view with their occurrence in *bio2C* questions. For brevity, only the ten types most frequently represented in *bio2C* are shown. Since the graph and the benchmark differ by several orders of magnitude, KG frequencies are displayed on a logarithmic scale.

As shown in Figure S2a, *bio2C* covers the main entity types represented in the RNA-KG view, while intentionally increasing the relative presence of RNA-centric entities. This enrichment reflects their central role in the selected view and ensures that a substantial fraction of the questions requires reasoning over RNA-specific entities. Other relevant biomedical classes, including diseases, genes, phenotypes, cancers, and GO concepts, remain broadly represented.

A similar trend emerges for relationship types (Figure S2b). Relations directly involving RNAs and diseases like `causes_or_contributes_to_condition`, `regulates_activity_of`, `expresses`, `under_expressed_in`, and `over_expressed_in`, are prominently represented in *bio2C*. Conversely, `has_phenotype`, although being the most frequent relationship in the RNA-KG view, is less prevalent in the benchmark. This relation primarily encodes standard and well-established biomedical knowledge, which is generally less informative to investigate through NL questions.

To further evaluate *bio2C* semantic coverage, a Principal Component Analysis (PCA) was developed. Embeddings were obtained using the OpenAI's *text-embedding-3-large* model on the set of NL questions. The same embedding model is used to construct the vector database adopted by the RAG pipelines, i.e., the PCA provides a two-dimensional projection of the database latent space. Figure S1 highlights the dataset categories in the PCA that provide a well-balanced coverage of the latent space, indicating that each category meaningfully represent the semantic variability of the underlying query formulations.

The latent space is uniformly populated, with no dominant regions, supporting the ability of the dataset to capture diverse semantic patterns. Covariance ellipses highlight correlations among the main node types as they appear within queries. As expected, disease and phenotype nodes exhibit strong overlap, reflecting both their semantic proximity and their frequent co-occurrence in Cypher queries. miRNAs form a dense cluster located near the center of the latent space, indicating its central role in the database semantics. This cluster shows significant overlap with disease, phenotype, and GO annotations, suggesting that miRNAs act as a cross-cutting semantic component across multiple query contexts.

S2 Representative Queries and Prompt Example

Table S1 reports one representative query for each *bio2C* category, together with its biomedical motivation, NL formulation, Cypher translation, and a sample execution output. Figure S3 shows the prompt generated by the `S+RAG+O` pipeline for the Advanced query. Boxes distinguish the enhanced schema summary, the retrieved NL–Cypher example, and its execution output. For brevity, ellipses indicate omitted schema elements, and only the TOP_1 most similar example is shown.

S3 Detailed Results with *bio2C* Categories

Table S2 reports the performance of $\texttt{GPT4o}^{\texttt{FT}}$ with the `RAG` pipeline across the query categories. For each metric, a trend is observed: after an initial performance drop for *1-hop* queries wrt. *Node* queries, performance partially recovers for *2-hop* queries and further improves for *3-hop* queries, before decreasing again for the *Advanced* category.

Category	JW	Jac	Cov	Pass
Node	95.6	87.4	94.3	92.0
1-hop	94.2	80.1	84.5	78.0
2-hop	95.6	89.2	91.4	90.0
3-hop	96.9	91.1	92.0	92.0
Advanced	91.3	73.6	84.4	78.0

Supp. Table S2. Performance of $\texttt{GPT4o}^{\texttt{FT}}$ with the `RAG` pipeline across query categories.

Figure S4 reports the average Pass computed for all models across query categories through the `RAG` pipeline. Across all categories, GPT-based models consistently outperform LLaMA-based ones (a significant gap can be observed in the figure between the two categories of LLMs). With the exception of `L8B`, all models exhibit the same trend discussed above, where *1-hop* queries are more challenging than *2-hop* and *3-hop* queries.

S4 Detailed Results on MedT2C

Table S3 reports the complete performance of the RAG-oriented pipelines on the manually curated MedT2C subset. Overall, configurations incorporating retrieved examples achieve the strongest semantic performance, with `S+RAG` obtaining the best results for $\texttt{GPT4o}^{\texttt{FT}}$.

Category	Biomedical motivation	NL question	Cypher query	Output (sample)

Node	Identifying carcinoma-related diseases associated with metastatic progression support oncological analyses and disease stratification.	Return diseases whose label contains “carcinoma” and whose description mentions “metastasis”.	`MATCH (d:Disease)`
			`WHERE d.Label CONTAINS 'carcinoma'`
			`AND d.Description CONTAINS 'metastasis'`
			`RETURN d.Label AS Disease,`
			`d.Description AS Description`
				{‘Disease’: ‘primary non-gestational choriocarcinoma of ovary’, ‘Description’: ‘primary non-gestational choriocarcinoma of ovary is a rare ovarian germ cell malignant tumor …’}, {‘Disease’: ‘salivary gland acinic cell carcinoma’, ‘Description’: ‘a carcinoma of the salivary gland characterized by serous acinar cell differentiation …’}, …

1-hop	Retrieving experimental evidence supporting miRNA dysregulation help prioritize candidate biomarkers for cardiovascular diseases such as heart failure.	Which experimental methods report miRNAs with a sequence length of 22 nucleotides that are under-expressed in cardiovascular diseases whose descriptions start with “failure of the heart”?	`MATCH (m:miRNA)-[r:'under expressed in']`
			`->(c:'Cardiovascular disease')`
			`WHERE SIZE(m.Sequence) = 22`
			`AND c.Description STARTS WITH`
			`'failure of the heart'`
			`RETURN r.Method AS Method,`
			`m.Label AS miRNA,`
			`c.Label AS Disease,`
			`c.Description AS Description`
				{‘Method’: [‘qrt-pcr etc’, ‘northern blotting’], ‘miRNA’: ‘hsa-miR-1-5p’, ‘Disease’: ‘congestive heart failure’, ‘Description’: ‘failure of the heart to pump a sufficient amount of blood to meet the needs of the body tissues …’}, {‘Method’: [‘qrt-pcr etc’, ‘northern blotting’, ‘quantitative reverse transcriptase pcr’], ‘miRNA’: ‘hsa-miR-1-3p’, ‘Disease’: ‘congestive heart failure’, …}, …

2-hop	Linking specific miRNAs to biological processes and their molecular functions support the functional characterization of miRNAs involved in RNA- and DNA-related regulatory mechanisms.	Identify molecular functions whose labels contain “RNA” or “DNA” and that are part of biological processes in which miRNAs whose labels end with a number participate. Return the miRNA, biological process, and molecular function labels.	`MATCH (m:miRNA)-[:participates_in]`
			`->(bp:'Biological process')`
			`-[:has_part]`
			`->(mf:'Molecular function')`
			`WHERE (toLower(mf.Label) CONTAINS 'rna'`
			`OR toLower(mf.Label) CONTAINS 'dna')`
			`AND m.Label = '.*[0-9]$'`
			`RETURN m.Label AS miRNA,`
			`bp.Label AS BiologicalProcess,`
			`mf.Label AS MolecularFunction`
				{‘miRNA’: ‘hsa-miR-543’, ‘BiologicalProcess’: ‘DNA metabolic process’, ‘MolecularFunction’: ‘DNA nuclease activity’}, {‘miRNA’: ‘hsa-miR-107’, ‘BiologicalProcess’: ‘transcription by RNA polymerase II’, ‘MolecularFunction’: ‘RNA polymerase II activity’}, {‘miRNA’: ‘hsa-miR-206’, ‘BiologicalProcess’: ‘RNA biosynthetic process’, ‘MolecularFunction’: ‘DNA-directed 5’-3’ RNA polymerase activity’}, …

3-hop	Tracing miRNA-regulated genes through genetic interactions to disease-associated genes support the identification of indirect molecular mechanisms and disease pathways influenced by *hsa-miR-29a-3p*.	Identify the diseases caused or contributed to by genes that genetically interact with genes regulated by the miRNA “hsa-miR-29a-3p”. Return the labels of the miRNA, the regulated gene, the interacting gene, and the disease.	`MATCH (m:miRNA {Label: 'hsa-miR-29a-3p'})`
			`-[:regulates_activity_of]->(g1:Gene)`
			`-[:genetically_interacts_with]->(g2:Gene)`
			`-[:causes_or_contributes_to_condition]`
			`->(d:Disease)`
			`RETURN m.Label AS miRNA,`
			`g1.Label AS RegulatedGene,`
			`g2.Label AS InteractingGene,`
			`d.Label AS Disease`
				{‘miRNA’: ‘hsa-miR-29a-3p’, ‘RegulatedGene’: ‘ITIH5’, ‘InteractingGene’: ‘CGB5’, ‘Disease’: ‘polycystic ovary syndrome’}, {‘miRNA’: ‘hsa-miR-29a-3p’, ‘RegulatedGene’: ‘ITIH5’, ‘InteractingGene’: ‘CDY1’, ‘Disease’: ‘spermatogenic failure, Y-linked, 2’}, …

Advanced	Aggregating TANRIC scores across over-expressed pre-miRNAs support the prioritization of cancer types characterized by stronger pre-miRNA dysregulation signals.	Which cancers have the highest average TANRIC score across pre-miRNAs that are over-expressed in them?	`MATCH (p:pre_miRNA)-[r:over_expressed_in]`
			`->(c:Cancer)`
			`WHERE r.TANRIC_score IS NOT NULL`
			`RETURN c.Label AS Cancer,`
			`AVG(r.TANRIC_score) AS avg_TANRIC_score`
			`ORDER BY avg_TANRIC_score DESC`
				{‘Cancer’: ‘glioblastoma’, ‘avg_TANRIC_score’: 0.4320150465339608}, {‘Cancer’: ‘lung adenocarcinoma’, ‘avg_TANRIC_score’: 0.3956548875707243}

Supp. Table S1. Representative biomedical queries across the five *bio2C* complexity levels.

	L8B	L8B^{FT}	GPT4om	GPT4om^{FT}	GPT4o	GPT4o^{FT}
Vanilla	69.9	86.9	85.3	**96.2**	85.2	95.6
Schema	73.9	77.6	89.5	95.4	90.3	95.0
RAG	73.9	77.6	93.5	95.2	94.2	95.9
RAG+O	73.9	77.6	93.2	95.6	94.4	95.9
S+RAG	**89.9**	**87.0**	**94.8**	96.0	**95.1**	**96.4**
S+RAG+O	76.1	75.3	94.6	96.1	94.8	95.8

(a) Average `JW` scores.

	L8B	L8B^{FT}	GPT4om	GPT4om^{FT}	GPT4o	GPT4o^{FT}
Vanilla	2.0	33.0	2.0	74.5	2.0	70.2
Schema	6.0	24.3	25.3	79.4	50.5	78.5
RAG	6.0	24.3	45.7	67.0	58.2	79.9
RAG+O	6.0	24.3	49.0	72.4	58.3	76.5
S+RAG	**37.3**	**41.8**	**54.8**	78.7	67.0	**87.2**
S+RAG+O	0.0	0.0	54.2	**81.3**	**69.8**	82.3

(b) Average `Jac` scores.

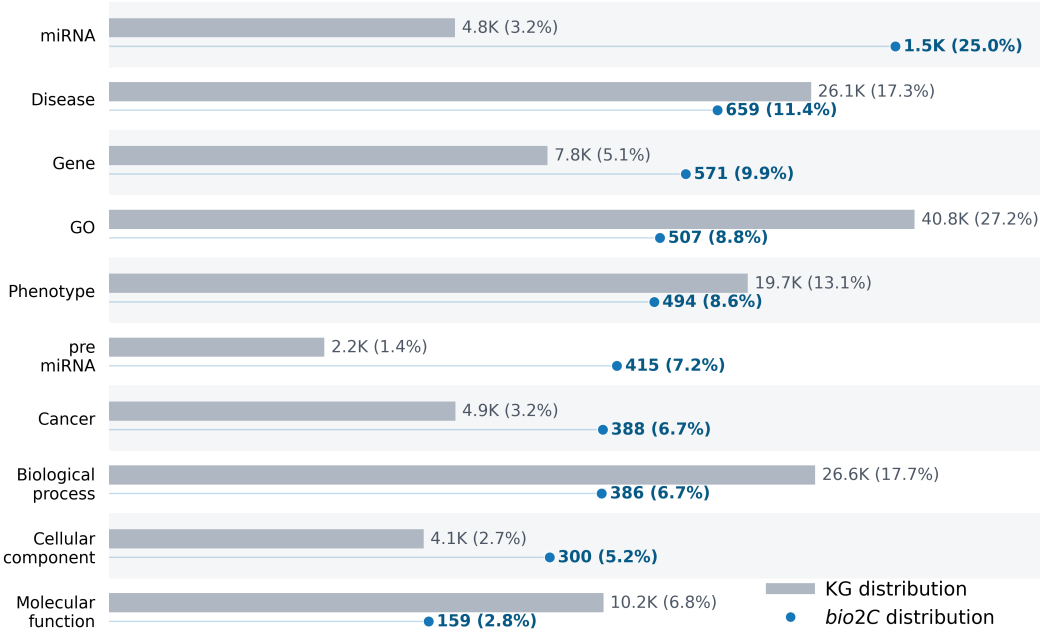
	L8B	L8B^{FT}	GPT4om	GPT4om^{FT}	GPT4o	GPT4o^{FT}
Vanilla	2.0	45.9	2.0	78.0	2.0	75.0
Schema	9.5	26.3	56.1	**86.0**	66.3	80.3
RAG	9.5	26.3	53.6	72.0	62.0	85.3
RAG+O	9.5	26.3	58.4	78.0	64.0	80.0
S+RAG	**40.7**	**46.0**	**58.5**	83.0	71.0	**88.8**
S+RAG+O	0.0	0.0	57.9	85.0	**75.0**	84.0

(c) Average `Cov` scores.

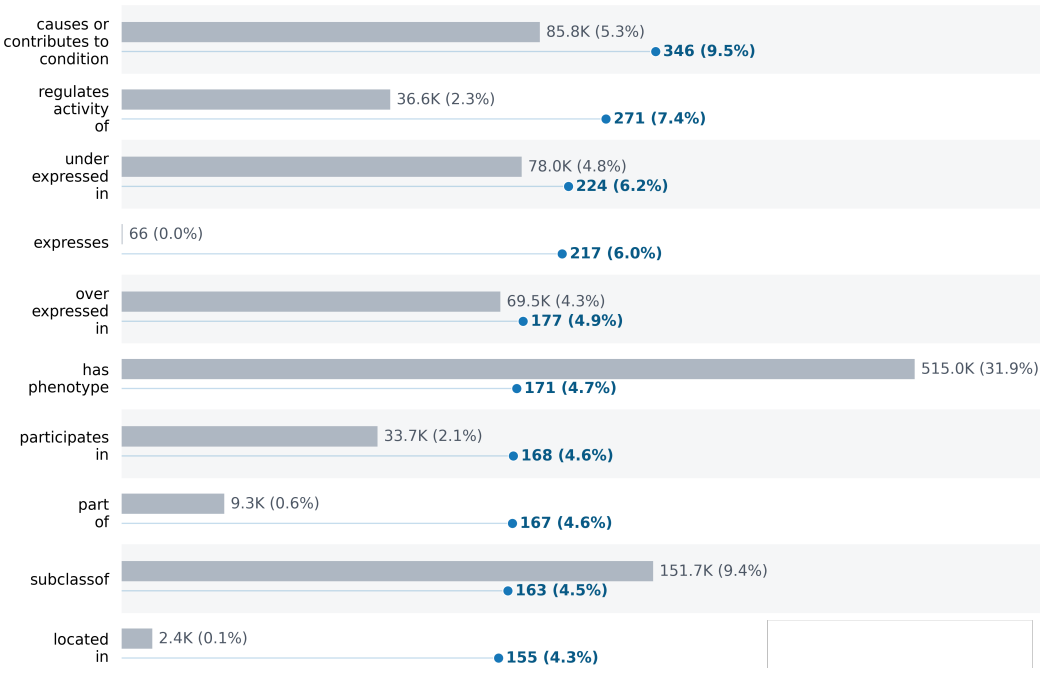
	L8B	L8B^{FT}	GPT4om	GPT4om^{FT}	GPT4o	GPT4o^{FT}
Vanilla	2.0	44.0	2.0	74.0	2.0	70.0
Schema	6.0	26.0	46.0	**82.0**	60.0	74.0
RAG	6.0	26.0	46.0	68.0	54.0	80.0
RAG+O	6.0	26.0	**52.0**	74.0	56.0	76.0
S+RAG	**34.0**	**46.0**	**52.0**	78.0	64.0	**84.0**
S+RAG+O	0.0	0.0	**52.0**	**82.0**	**70.0**	80.0

(d) Average `Pass` scores.

Supp. Table S3. Performance for MedT2C.



(a) Distribution of node types in the RNA-KG view and *bio2C*.



(b) Distribution of relationship types in the RNA-KG view and *bio2C*.

Supp. Fig. S2. Comparison between the distribution of the RNA-KG view and *bio2C*.

Instructions

```
Given an input question, convert it to a Cypher query.  No pre-amble.
Based on the Neo4j graph schema below, write a Cypher query that would answer the user's question.
Return only Cypher statement, no backticks, nothing else.
```

Schema

Node properties:

```
pre_miRNA
  - Description:  STRING Example:  "homo sapiens (human) microrna hsa-mir-625 precursor"
  - Label:  STRING Example:  "microRNA hsa-mir-625 precursor"
  - Sequence:  STRING Example:  "AGGGUAGAGGGAUGAGGGGGAAAGUUCUAUAGUCCUGUAAUUAGAUCUCA"
  ...

Cancer
  - Label:  STRING Example:  "carcinoid syndrome"
  - Description:  STRING Example:  "a rare tumor characterized by malignant ..."
  ...

...
```

Relationships:

```
(:pre_miRNA)-[:over_expressed_in]->(:Cancer)
(:pre_miRNA)-[:under_expressed_in]->(:Cancer)
(:pre_miRNA)-[:causes_or_contributes_to_condition]->(:Cancer)
...
```

Relationship properties:

```
over_expressed_in
  - FDR: FLOAT Min:  0.0, Max:  0.009989965
  - Source:  LIST Item Example:  "dbDEMC"
  - PubMedID: LIST Item Example:  "23188185"
  - TANRIC_score:  FLOAT Min:  0.3956548875707243, Max:  0.4320150465339608
  ...

under_expressed_in
  - FDR: FLOAT Min:  8.25E-145, Max:  0.009999129
  - Source:  LIST Item Example:  "dbDEMC"
  - PubMedID: LIST Item Example:  "25712342"
  ...

causes_or_contributes_to_condition
  - PubMedID: LIST Item Example:  "26752646"
  - RNAsister_score:  FLOAT Min:  0.968971, Max:  1.0
  - Source:  LIST Item Example:  "DisGeNET"
  ...
...
```

Retrieved example

```
The following example provides useful patterns for querying the graph database (Neo4j outputs are also reported).

[Query natural language]
Which cancers have the highest number of pre-miRNAs that are over-expressed with a TANRIC score greater than 0.2?
Return each cancer and the number of associated over-expressed pre-miRNAs, ranked by this number.

[Query Cypher]
MATCH (p:pre_miRNA)-[r:over_expressed_in]->(c:Cancer)
WHERE r.TANRIC_score > 0.2
RETURN c.Label AS Cancer, COUNT(r) AS nPreMiRNAs
ORDER BY nPreMiRNAs DESC
```

Execution output

[Output Neo4j]

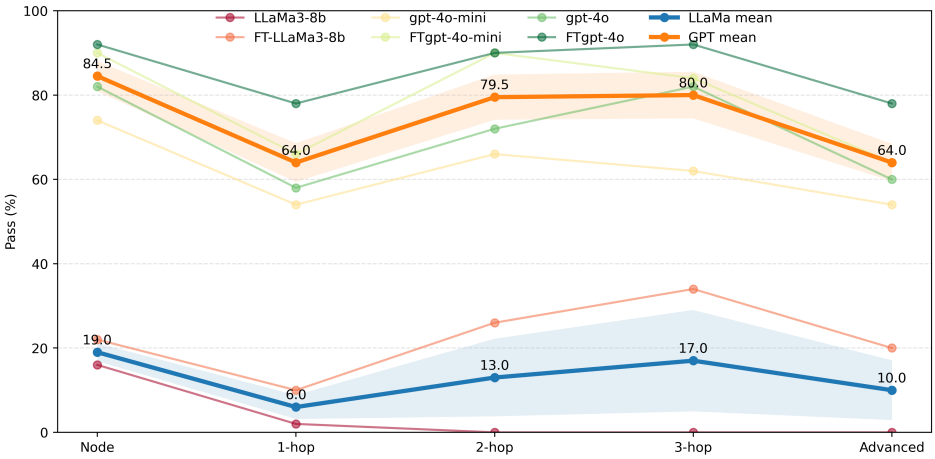
```
{
  "Cancer":  "glioblastoma",
  "nPreMiRNAs":  1
}

{
  "Cancer":  "lung adenocarcinoma",
  "nPreMiRNAs":  1
}
```

NL question

Question: Which cancers have the highest average TANRIC score across pre-miRNAs that are over-expressed in them?
Cypher query:

Supp. Fig. S3. Prompt for the S+RAG+O pipeline.



Supp. Fig. S4. Average Pass performance across query categories and models for the RAG pipeline. Bold lines represent the mean performance across GPT and LLaMA models. GPT models outperform LLaMA ones; all models follow a similar trend, with *1-hop* queries being more challenging than *2-hops* and *3-hops* queries.