M-Cypher: A GQL Framework Supporting Motifs

Demonstrated by Covid-19 Knowledge Graph Analysis

Xiaodong Li[1], Reynold Cheng[1], Matin Najafi[1], Kevin Chang[2], Xiaolin Han[1], Hongtai Cao[2]
{xdli,ckcheng,mnajafi,xlhan}@cs.hku.hk,{kcchang,hongtai2}@illinois.edu
[1]The University of Hong Kong, [2]University of Illinois at Urbana-Champaign

ABSTRACT

Graph databases witness the rise of Graph Query Language (GQL) in recent years, which enables non-programmers to express a graph query. However, the current solution does not support motif-related queries on knowledge graphs, which are proven important in many real-world scenarios. In this paper, we propose a GQL framework for mining knowledge graphs, named M-Cypher. It supports motif-related graph queries in an effective, efficient and user-friendly manner. We demonstrate the usage of the system by the emerging Covid-19 knowledge graph analytic tasks.

CCS CONCEPTS

• **Theory of computation** → *Pattern matching*; • **Information systems** → *Network data models*; *Query languages for non-relational engines*; • **Computing methodologies** → *Motif discovery*.

KEYWORDS

motif, GQL, Covid-19 knowledge graph

ACM Reference Format:
Xiaodong Li[1], Reynold Cheng[1], Matin Najafi[1], Kevin Chang[2], Xiaolin Han[1], Hongtai Cao[2]. 2020. M-Cypher: A GQL Framework Supporting Motifs: Demonstrated by Covid-19 Knowledge Graph Analysis. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3417440

1 INTRODUCTION

Recently there is an increasing interest in graph data, representing each data entity as a node (a.k.a. a vertex) and each relation between two entities as an edge. The need for efficient and effective storing and querying of such data has led the development of graph databases. The graph data are usually with hierarchical, complex, or even arbitrary structures in order to maintain data from different domains, e.g., the knowledge graphs.

Knowledge Graph. As shown in Figure 1, the knowledge graph is usually presented as a directed graph with attributes (data values and labels) on nodes and edges, denoting the domain knowledge from different areas. Knowledge graphs are at the core of many tools, e.g., voice assistants and recommendation systems [6, 14]. In this paper, we generate the Covid-19 knowledge graph by integrating data from OpenKG, HPO, NCBI and DrugBank, with 48K nodes and 815K edges, following the schema in Figure 1 [1].

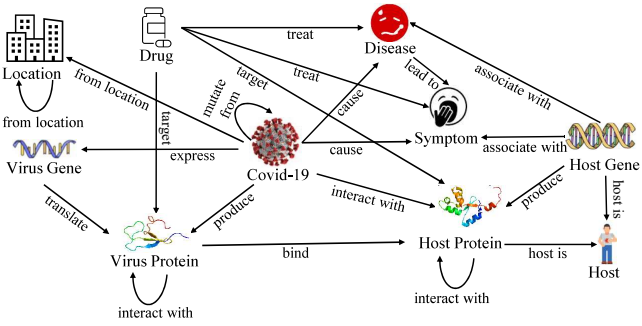


Figure 1: Schema of the Covid-19 knowledge graph, modeling 9 node types and 10 edge types.

GQL. However, the query for knowledge graph is difficult to be expressed because of the complex structure and attributes, especially for the non-programmers. To answer it, standardized Graph Query Languages (GQL) are designed in order to support flexible querying in graph databases, just like SQL for relational database systems. For example, Cypher is one of the most popular GQL currently in both research and industrial communities [6]. However, Cypher and all other GQLs lack support of motifs.

Motifs. Recently, motif-based graph analysis becomes an important tool for discovering insight from knowledge graphs [10, 14]. A motif, which is also known as higher-order structure or graphlet, is a small subgraph pattern [15]. As pointed out by [3], a motif is a fundamental building block of large and complex networks, and it enables "high-order semantics" analysis, which has been shown to be more effective than traditional "graph-edge-based" solutions in a range of problems, such as link prediction [2], graph clustering [3] and node ranking [18]. Currently, there are few graph databases supporting motif-related queries on knowledge graphs.

M-Cypher. Since existing graph databases cannot enable efficient expression or execution of motif-related queries on knowledge graphs, we build a query processing framework based on Cypher to support motif-related queries, named M-Cypher. M-Cypher provides access to motif-related functionalities that is not available in Cypher in an efficient and user-friendly manner. By designing a powerful parser (i.e., Figure 1), M-Cypher integrates the features from both motifs (e.g., higher-order semantics for better effectiveness) and standard Cypher syntax (e.g., user-friendly expression in a normative natural-language specification). Also, we design an efficient subgraph matching engine to speed up the motif-related

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3417440

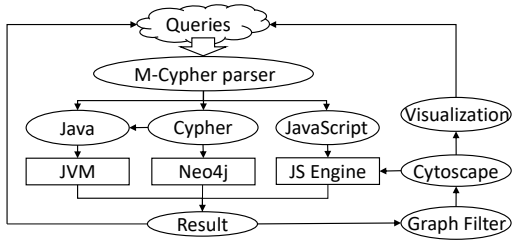
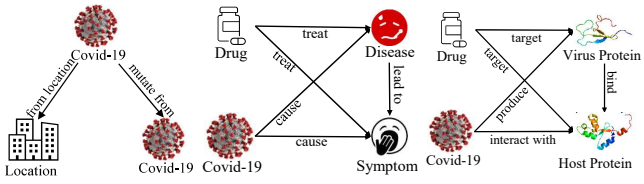


Figure 2: Flowchart of the M-Cypher framework.



(a) M_1 **(b)** M_2 **(c)** M_3

Figure 3: Illustration of three motifs from Figure 1, where M_1 is for Covid-19 spreading analysis, and M_2 and M_3 are for drug re-purposing.

queries by avoiding over-counting and reusing the common substructures. We also develop a network visualization module to make it easier for interaction.

2 CURRENT AND FUTURE GQL

Cypher, one of the most successful GQLs, allows creating, reading, updating and traversing (CRUT) elements of knowledge graphs. The queries can be expressed by compact fixed or variable length patterns which combine visual representations of node and edge topologies, with label existence and property value predicates [6]. By matching the queried pattern against graph data elements, a query can extract references to nodes, edges and paths of interest [7]. However, currently Cypher does not support subgraph isomorphism queries, which is necessary in the motif-related tasks.

On the other hand, recently many graph databases start supporting motifs, i.e., Table 1. However, these graph databases basically have no GQL features, and cannot deal with knowledge graphs. In the table, basic operations include *a. CRUT operations*, *b. GQL feature*, *c. motif counting* [15] *d. motif discovery* [10] and *e. visualization*; advanced operations include *f. motif connectivity* [12] *g. motif clique* [10] *h. motif-based graph clustering* [3] *i. motif-based node ranking* [18] and *j. motif-based link prediction* [2]. From the table, most graph databases only work for basic operations. Thus we develop M-Cypher, which supports all operations listed in the table, as well as other emerging functions that are related to motifs.

3 M-CYPHER FRAMEWORK

In this section, we will introduce the syntax and semantics of M-Cypher queries and the advanced algorithms supported by M-Cypher. The project is open-sourced in [1].

Syntax and Semantics. The knowledge graph is a multi-relational graph composed of entities (nodes) and relations (edges), which

Table 1: Current graph databases supporting motifs.

Graph DBs	Basic Operations					Advanced Operations				
	a.	*b.*	*c.*	*d.*	*e.*	*f.*	*g.*	*h.*	*i.*	*j.*
SNAP [9]			✓					✓		
Bio4j [16]	✓	✓	✓							
ISMAGS [17]	✓		✓	✓	✓					
G.Crunch [8]	✓		✓	✓	✓					
MC-Explorer [10]			✓	✓	✓		✓			
M-Cypher	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

are triplets of facts <h, r, t> (head entity h, relation r, tail entity t). For example, Cypher will express the query Q_1="*find the human proteins which are targeted by Covid-19*" as

```
MATCH (h1:Virus)-[r1:interact_with]->(t1:HostProtein)
-[r2:host_is]->(t2:Host)
WHERE h1.name = "SARS-CoV-2" AND t2.name = "Human"
RETURN t1.name
```

Note that in Q_1, the tail entity t_1 of the first relation r_1 is also the head entity of the second relation r_2. Each node is surrounded by parentheses in the format of `(variable:label)`, e.g., `(h1:Virus)`. Similarly, each relation is surrounded by square brackets in the format of `[variable:label]`, e.g., `[r1:interact_with]`. The arrow indicates the direction of the edge. The clause composed by entities and relations after `MATCH` is the pattern to be matched, and the clause led by `WHERE` is the constrain. Finally, the user can specify what to be returned in the clause of `RETURN`.

However, subgraph isomorphism queries are not supported in Cypher, making it difficult to embed motif features into the query pattern. Though `MATCH` clause can support some small motifs (e.g., paths), it cannot describe bigger motifs with complex structures. Also, Cypher executes the matching query by a sequence of join operations, leading to low efficiency and over-counting [6], which severely affects the mining effectiveness. Thus we propose a new scheme to express and execute the motif-related queries.

In the query syntax level, we introduce a fixed primitive M, denoting the motif pattern to be matched and it can be specified by the user in a graphical user interface (e.g., Figure 4a). Then M can be used to describe the motif-related queries. For example, the motif-counting query can be issued by `MATCH (m:M) RETURN COUNT(m)`, or `RETURN MCOUNT(M)`. For a motif-instance enumeration query, the users can directly call `MATCH (m:M) RETURN m`, which will return a list of motif-instances. The parser of M-Cypher running in JavaScript Engine will recognize the motif-related primitives, and forward it into Neo4j first for processing. Next, the inputs of the motif-related functions are collected and passed to the algorithms run in JVM via Neo4j Java Driver (e.g., Figure 2). Finally, the query results are passed to the visualization module based on Cytoscape. Note that the motif input module in Figure 4a allows submitting multiple

[1]The motif clique demonstrates the two host proteins "NR3C1" and "POU1F1" which are both targeted by the Covid-19 virus, and share a set of symptoms (denoted by the symptom ID from the HPO group https://hpo.jax.org). Note that the host genes are hidden for clarity since host protein and host gene are of one-to-one mappings.

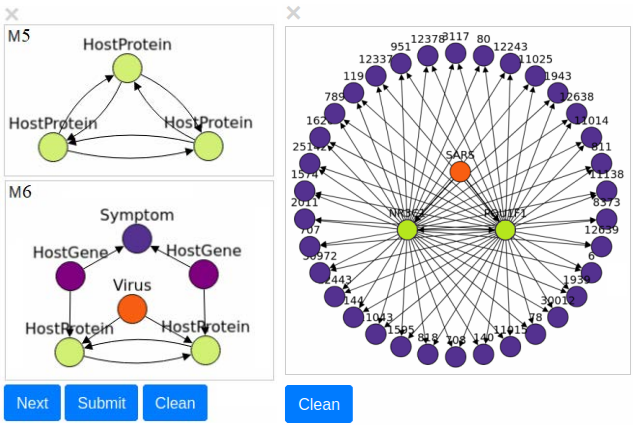


Figure 4: M-Cypher user interface for (a) motif input and (b) visualization of the motif clique[1] composed by Motif M_6.

motifs by clicking "Next", denoted as $\mathcal{M} = \{M_i | i = 1, 2, ..., t\}$[2]. As we will show, it is used for the functions which require multiple motifs. Also, for users who have no idea about which motif to use, there is a function developed in M-Cypher semantics called MDIS. To find the significant motifs in the knowledge graph, a user can specify (a) the maximum size of the motif as k and (b) the set of labels of interest as L. The node of interest s is an optional choice, where motif-instances containing s will be counted when this parameter is issued. Then MDIS will return the counting number of motif-instances $f(M_i)$ for each possible motif M_i of labels from L, and the user can pick the motifs of high frequencies.

```
MATCH (s:Virus) WHERE s.name="SARS-CoV-2"
CALL MDIS(s,4,["Drug","Virus","Disease","Symptom"])
YIELD permutation AS motif, frequency
RETURN motif, frequency ORDER BY frequency DESC
```

Also, we can use the connectivity of the motif-instances to extract information. For example, for the task Q_2="*find the cities where similar Covid-19 viruses are detected to those in Hong Kong*", the corresponding query can be issued in the following manner, with M from Figure 3a. Note that "$*$" means that the pattern is of variable length, e.g., there will be a sequence of motif-instances to connect the cities s and t (e.g., Figure 5) [12].

```
MATCH (s:Location)-[m:M*]->(t:Location)
WHERE s.name = "Hong Kong" RETURN t.name
```

In the execution level, we design an efficient algorithm which can avoid over-counting. In this paper, we focus on the node-induced subgraph, which is a popular setup in motif area [12]. Given a set of nodes V_m, the node-induced subgraph (V_m, E_m) is the subgraph of the original graph (V, E) such that $(u, v) \in E_m \iff (u, v) \in E$ where $u, v \in V_m$. For each label of the nodes, we build an index

[2]The default motif primitive M is M_1 if the user does not specify which motifs to use. Also, we skip the edge labels in the UIs for clarity, which are unique when the end nodes are fixed in the current Covid-19 knowledge graph.

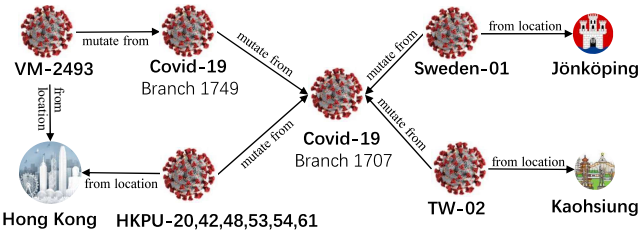


Figure 5: Motif connectivity demonstration to find the cities connected with Hong Kong among Covid-19 spread.

Table 2: Motif-related queries supported by M-Cypher.

Function	Input	Output	Function	Input	Output
MDIS	$[s], k, L$	$(M_i, f(M_i))$	MAM	$\mathcal{M}$	A
MPPR	$s, \mathcal{M}, b, \delta$	(s_i, p_i)	MCLQ	s, M	Q
MCON	S, M	$\varphi(S)$	MCC	s, M	Q
MFV	$s, [t], \mathcal{M}$	$f(M_i)$	MGD	M	Φ
SMPD	s, t, M	$d_M(s, t)$	MCOUNT	M	$f(M)$

to match the nodes as well as their neighbors. We notice that the motif-instances which may be over counted are the ones who have multiple nodes V_s of the same label. Also, it happens only if some nodes in V_s are of the same *orbit*, which means that the switching of these nodes does not affect the isomorphism [12]. For example, for the clique composed by γ nodes with the same label (e.g., M_5 in Figure 4a), all nodes are in the same orbit, thus we only need to generate one permutation for matching, rather than the $\gamma!$ permutations, which is costly and leads to over-counting. Also, we find that some substructures are repeatedly enumerated when issuing a set of motifs to match. To reduce the matching time, we detect the substructures shared by these motifs, and then extend them to match different motif-instances. Since the shared substructures only need to be enumerated once, the matching process is boosted.

Advanced Data Analytic Tools. With the queries supported by M-Cypher in both syntax level and execution level, many motif-related queries of the state-of-the-art are supported, i.e., Table 2.

Personal Page Rank (PPR) is an algorithm that measures the influence of nodes in the graph, by iteratively distributing each node's score over its neighbors in a random walk manner. Recently, it is found that in many cases, PPR based on motifs (MPPR) can obtain better effectiveness [18]. By specifying the source node s, the motifs of interest $\mathcal{M}$, the iterations b and the damping factor δ, the function will return a rank of the nodes s_i according to the MPPR scores p_i. Motif Conductance (MCON) is the motif-generalized version of conductance, whose effectiveness is proven in graph clustering tasks [3]. By specifying the set of nodes S and the motif of interest M, the function will return the motif conductance $\varphi(S)$. Motif Feature Vector (MFV) is considered as an important feature for both node (e.g., for node classification) and edge (e.g., for link prediction) [2]. Given a set of motifs of interest $\mathcal{M}$ and the node s (or missing edge (s, t)) to be queried, MFV returns the number of motif-instances that contain s, e.g., $f_s(M_i)$ (or $f_{s,t}(M_i)$ for node pair (s, t)). We also develop Shortest Motif-Path Distance (SMPD) [12], which generalizes the shortest path [11] and calculate the

minimum number of motif-instances $d_M(s,t)$ that links nodes s and t; Motif Adjacency Matrix (MAM) [3, 18], which generalizes the adjacency matrix and returns matrix A where $A_{i,j} = f_{i,j}(M)$, $M \in \mathcal{M}$; Motif Clique (MCQ) [10], which generalizes the k-cliques [5] and returns the node set Q of the motif clique that contains the given node s; Motif Connected Component (MCC) [3], which generalizes connected component [4] and returns the node set Q of the motif connected component that contains s; Motif Graph Diameter (MGD) [12], which generalizes the graph diameter [13] and returns the diameter Φ of the higher-order graph.

These motif-related algorithms are summarized in Table 2 and callable in M-Cypher, with input (parameters to be specified in the `CALL` clause, with those surrounded by square brackets as the optional parameters) and output (results in the `YIELD` clause). For example, to answer the task Q_3 ="*find the motif feature vectors for Covid-19's potential drugs*", we use two motifs $\mathcal{M} = \{M_2, M_3\}$ from Figure 3, which focus on symptom information and protein information respectively. Then we can obtain the motif feature vector v_m for each possible drug candidate by the following M-Cypher query. A potential drug for covid-19 treatment is expected to have more motif-instances of both M_2 and M_3.

```
MATCH (s:Virus) WHERE s.name="SARS-CoV-2" MATCH (t:Drug)
CALL MFV(s,t,[M1,M2]) YIELD s, t, vm RETURN t.name, vm
```

Case Study. We engage the audience closely by a case study for drug re-purposing, e.g., Q_4 ="*what are the potential drugs for Covid-19 treatment*", because the existing drugs have potential effective pathways to Covid-19, with short development cycle and controllable side-effect, compared to develop new drugs [19]. And the user can use M-Cypher to answer this task. The user needs to connect a Neo4j database first where the Covid-19 knowledge graph is stored, e.g., Figure 6, then choose the suitable functions from Table 2. Here we use MPPR. Note that other algorithms can also be used here, e.g., the motif feature vectors generated in Q_3 can be further used by standard downstream link prediction modules in python libraries [2] to predict the existence probability of the missing link between each possible drug and Covid-19 virus.

Next, the user needs to choose the motifs of interest, or turn to MDIS for help. Here we use motif M_2 and M_3 from Figure 3, as well as triangle motifs composed by virus proteins (M_4) and host proteins (M_5) respectively, since triangles are proven significant in protein-protein interaction networks [3]. These motifs can be drawn from the user interface, e.g., Figure 4a. Based on the user's choice, the query can be written, with the input and output specified in `CALL` and `YIELD` clauses respectively. After clicking the "Run" button, it returns the drug lists with the MPPR scores (e.g., Figure 6), and logs the query time and entry count. We notice that many drugs ranked top are already recommended by the pharmacologists for Covid-19 treatment [19], while the ordinary PPR algorithm cannot answer this task on the knowledge graph.

To understand better about the knowledge graph, the user can preview the graph by issuing a query, e.g., `MATCH (n) RETURN n` and then click the "Visualize" button. The author may inspect the nodes of interest by specifying the labels in the `MATCH` clause and the constrains in the `WHERE` clause, e.g., `WHERE s.name CONTAINS "SARS"`. More using cases are introduced in [1].

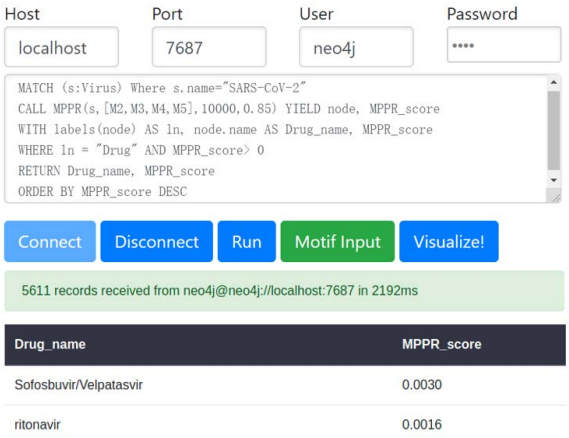


Figure 6: Case study of drug re-purposing by M-Cypher.

ACKNOWLEDGMENTS

R. Cheng, X. Li and X. Han were supported by HK RGC (17229116, 106150091, and 17205115), HKU (104004572, 102009508, and 104004129), and HKITF (MRP/029/18). K. Chang and H. Cao were supported by National Science Foundation IIS 16-19302 and IIS 16-33755, ZJU Research 083650, Futurewei Technologies HF2017060011 and 094013, UIUC OVCR CCIL Planning Grant 434S34, UIUC CSBS Small Grant 434C8U, and Advanced Digital Sciences Center Faculty Grant.

REFERENCES

[1] https://github.com/sheldon2016/covid19motif.
[2] G. AbuOda, G. D. F. Morales, and A. Aboulnaga. Link prediction via higher-order motif features. In *ECML PKDD*, pages 412–429. Springer, 2019.
[3] A. R. Benson, D. F. Gleich, and J. Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.
[4] Y. Fang, R. Cheng, X. Li, S. Luo, and J. Hu. Effective community search over large spatial graphs. *PVLDB*, 10(6):709–720, 2017.
[5] Y. Fang, Z. Wang, R. Cheng, X. Li, S. Luo, J. Hu, and X. Chen. On spatial-aware community search. *IEEE TKDE*, 31(4):783–798, 2018.
[6] N. Francis, A. Green, and P. Guagliardo et al. Cypher: An evolving query language for property graphs. In *ACM SIGMOD*, pages 1433–1445, 2018.
[7] X. Han, T. Grubenmann, R. Cheng, S. C. Wong, X. Li, and W. Sun. Traffic incident detection: A trajectory-based approach. In *IEEE ICDE*, pages 1866–1869, 2020.
[8] O. Kuchaiev and A. Stevanović et al. Graphcrunch 2: Software tool for network modeling, alignment and clustering. *BMC bioinformatics*, 12(1):1–13, 2011.
[9] J. Leskovec and R. Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM TIST*, 8(1):1–20, 2016.
[10] B. Li, R. Cheng, J. Hu, and Y. Fang et al. Mc-explorer: Analyzing and visualizing motif-cliques on large networks. In *IEEE ICDE*, pages 1722–1725, 2020.
[11] X. Li. Durs: A distributed method for k-nearest neighbor search on uncertain graphs. In *IEEE MDM*, pages 377–378, 2019.
[12] X. Li, T. N. Chan, and R. Cheng et al. Motif paths: A new approach for analyzing higher-order semantics between graph nodes. *HKU Technique Reports*, 3:4, 2019.
[13] X. Li, R. Cheng, Y. Fang, J. Hu, and S. Maniu. Scalable evaluation of k-nn queries on large uncertain graphs. *EDBT*, 2018.
[14] Y. Li, Z. Lou, Y. Shi, and J. Han. Temporal motifs in heterogeneous information networks. In *MLG Workshop@ KDD*, 2018.
[15] C. Ma, R. Cheng, L. V. Lakshmanan, T. Grubenmann, Y. Fang, and X. Li. Linc: a motif counting algorithm for uncertain graphs. *PVLDB*, 13(2):155–168, 2019.
[16] P. Pareja-Tobes, R. Tobes, and M. Manrique et al. Bio4j: a high-performance cloud-enabled graph-based data platform. *BioRxiv*, page 016758, 2015.
[17] T. Van Parys, I. Melckenbeeck, and M. Houbraken et al. A cytoscape app for motif enumeration with ismags. *Bioinformatics*, 33(3):461–463, 2017.
[18] H. Zhao, X. Xu, Y. Song, D. L. Lee, Z. Chen, and H. Gao. Ranking users in social networks with higher-order structures. In *AAAI*, 2018.
[19] Y. Zhou, Y. Hou, and J. Shen et al. Network-based drug repurposing for novel coronavirus 2019-ncov/sars-cov-2. *Cell discovery*, 6(1):1–18, 2020.