

ERMer: a serverless platform for navigating, analyzing, and visualizing *Escherichia coli* regulatory landscape through graph database

Zhitao Mao^{1,2,†}, Ruoyu Wang^{1,2,†}, Haoran Li^{1,2,†}, Yixin Huang³, Qiang Zhang⁴, Xiaoping Liao^{1,2,*} and Hongwu Ma^{1,2,*}

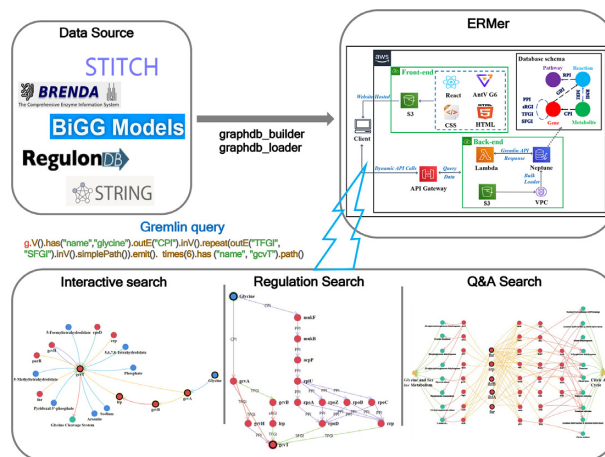
¹Biodesign Center, Key Laboratory of Systems Microbial Biotechnology, Tianjin Institute of Industrial Biotechnology, Chinese Academy of Sciences, Tianjin 300308, PR China, ²National Technology Innovation Center of Synthetic Biology, Tianjin 300308, PR China, ³AWS Professional Services, No.576 West Tianshan Road, Shanghai 200335, PR China and ⁴AWS Solution Architect Sector, Amazon Web Service Inc, Beijing 100016, PR China

Received February 20, 2022; Revised March 28, 2022; Editorial Decision April 06, 2022; Accepted April 26, 2022

ABSTRACT

Cellular regulation is inherently complex, and one particular cellular function is often controlled by a cascade of different types of regulatory interactions. For example, the activity of a transcription factor (TF), which regulates the expression level of downstream genes through transcriptional regulation, can be regulated by small molecules through compound–protein interactions. To identify such complex regulatory cascades, traditional relational databases require ineffective additional operations and are computationally expensive. In contrast, graph databases are purposefully developed to execute such deep searches efficiently. Here, we present ERMer (*E. coli* Regulation Miner), the first cloud platform for mining the regulatory landscape of *Escherichia coli* based on graph databases. Combining the AWS Neptune graph database, AWS lambda function, and G6 graph visualization engine enables quick search and visualization of complex regulatory cascades/patterns. Users can also interactively navigate the *E. coli* regulatory landscape through ERMer. Furthermore, a Q&A module is included to showcase the power of graph databases in answering complex biological questions through simple queries. The back-end graph model can be easily extended as new data become available. In addition, the framework implemented in ERMer can be easily migrated to other applications or organisms. ERMer is available at <https://ermer.biodesign.ac.cn/>.

GRAPHICAL ABSTRACT



INTRODUCTION

The graph is the natural way to model and represent connected data (1). This idea is not new, but has now become more viable via the introduction of scalable graph databases (2). Unlike traditional ways of managing data, such as relational databases, graph modelling allows for the real-world heterogeneity of data and efficiently handles complex deep queries (2,3). Graph databases have been widely adopted, particularly in areas including complex relationships such as social networks, financial services, and marketing (4,5). In recent years, several studies using graph databases for biodata storage and analysis have also been reported (6–11). For example, cyNeo4j connects Cytoscape and Neo4j, allowing users to use the Cypher query language to navigate and explore biological networks (6). Biochem4j shows the flexible integration and exploitation of biolog-

*To whom correspondence should be addressed. Tel: +86 22 24828735; Email: ma_hw@tib.cas.cn

Correspondence may also be addressed to Xiaoping Liao. Tel: +86 22 24828758; Email: liao_xp@tib.cas.cn

[†]The authors wish it to be known that, in their opinion, the first three authors should be regarded as joint First Authors

ical data sources from public databases and laboratory-specific experimental datasets using graph database (9). Reactome migrates the content from the relational database to a graph database for providing more efficient access to its high-quality curated reaction data (10). CKG describes a knowledge graph framework that allows clinically meaningful queries and advanced statistical analyses, enabling automated data analysis, knowledge mining, and visualization (11). One common problem of these tools is that the end-users need to write queries using specifically developed graph query languages to perform complex analyses. This makes them out of reach for most biologists unfamiliar with programming language. Another common problem is a lack of a public website/platform for end-users to easily navigate and interact with the graph databases.

In this study, by combining graph databases, serverless platform building architecture, and graph visualization tools, we proposed a new framework to store and analyze highly connected data, providing smoothly interactive user experiences for the end-users. As a showcase, we choose *E. coli* regulations as our study subjects for the following two reasons. First, cellular regulation is inherently highly complex and influenced by various types of interactions, such as protein activity regulation by small molecules or the regulation of gene expression by TF and regulatory sRNAs. One particular biofunction (e.g. a metabolic pathway for the synthesis of amino acid) is often regulated by feedback or feedforward loops consisting of different types of interactions. Whereas it is difficult for a biologist to identify and effectively modify such regulatory loops by focusing on one or two types of interactions closely related to the studied metabolites/proteins. Therefore it is helpful for biologists by bringing these different types of regulatory interactions together and providing some efficient ways to systematically search all possible regulatory cascades, thus better understanding the direct and indirect cause-effect relationships related to the biofunction interested. Second, as a model organism, *E. coli* has been studied for decades with enough data available to gain a more complete picture of the regulatory landscape. In contrast, regulation information for other organisms is often very patchy. Therefore *Escherichia coli* regulatory network is a suitable subject for our study.

Currently, there are some specialized databases for different types of regulation data. STRING (12) and STITCH (13) contain a large number of known and predicted protein-protein interactions (PPIs) and chemical-protein interactions (CPIs). The BRENDA database contains experimentally validated CPIs that affect enzyme activity (14). EcoCyc (15) and RegulonDB (16) contain curated data on transcriptional regulation and CPIs affecting TF activity, etc. EcoIN (17) is an *E. coli* integrated network that integrated the above regulatory information with the metabolic network. As these databases adopt a relational database strategy, three ways are available to search these regulatory paths between two entities. First, as Reactome describes (10), the join operation is required but this leads to degradation of performance and excessive response times. Second, as with ComiRNet (18), complex retrieval can be achieved using virtual tables and recursive queries. Third, merges tables and then uses other graph analysis tools such as NetworkX (19) and Pajek (20) for path search. On the other

hand, graph databases are purposefully designed to address such deep search problems and can be a suitable alternative to relational databases for storing and analyzing regulation data.

Here we presented ERMer, a cloud platform for mining complex regulatory cascades/patterns in the *E. coli* regulatory network using a graph database backend. We first implemented a graphdb_builder to automatically collect high-quality interaction data from various databases. A graphdb_loader module was then built to incorporate the data into an AWS Neptune graph database instance. Finally, we integrated the AWS lambda function and G6 graph visualization to provide three major functions for interacting with the whole regulatory landscape: (i) a interactive search function facilitates the extension of regulatory cascades by interactive exploration, (ii) a regulatory cascades retrieval function enables the mining of regulation paths, (iii) a question and answer (Q&A) system for retrieving key regulatory metabolites and regulatory factors across pathways. To the best of our knowledge, ERMer is the first cloud platform offering an overview of the regulatory landscape of *E. coli* based on a graph database approach. It enables effective interactive navigation and visualization, which can help researchers uncover the complete regulatory map and find complex regulatory cascades/patterns.

RESULTS

Graph database and cloud platform construction

We wrote a library of parsers with associated configurations for each source database to build the graph database (Supplementary materials). This library consists of three parts, namely data_downloader, data_filter, and data_integrator. The data_downloader module was used to get metabolic and regulatory interactions. Sigma factor regulations, TF regulations, and sRNA regulations were directly obtained from RegulonDB. PPIs in *E. coli* were obtained directly from the STRING database. CPIs were collected from three databases: BRENDA, RegulonDB and STITCH. We parsed the genome-scale metabolic model of *E. coli* iML1515 (21) downloaded from BiGG (22) to get the gene-reaction, reaction-metabolite and pathway-reaction relationships for pathway-based analysis. Then data_filter module filters data from different data sources using different processing flows. After that, all of these data are converted to tabular files using the data_integrator module. Finally, the graph loader module loaded the above data into the graph database, whose schema consists of nodes, edges, and properties (Supplementary materials). In ERMer, there are four types of nodes and nine types of edges (Figure 1 and Table 1), comprising 8421 nodes and 36331 edges (Table 1).

ERMer is built in an entire cloud-based serverless architecture (Figure 1), enabling high reliability, robustness and scalability (Supplementary materials). AWS Neptune is a fully-managed graph database service that was used as the database backend to store ERMer nodes and edges. When a user sends a request from the client, the request will be forwarded to the AWS Lambda function through API Gateway. Then the Lambda function invokes the corresponding gremlin API to query the requested data from

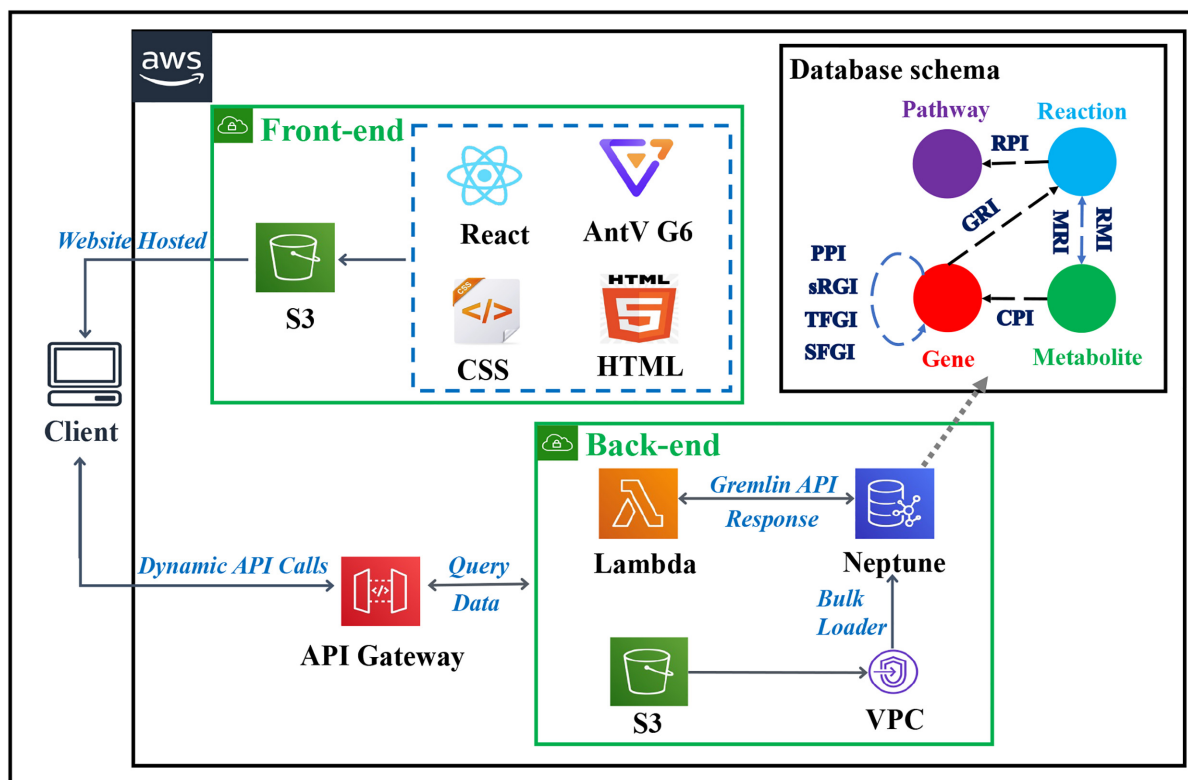


Figure 1. The graph database schema and architecture of ERMer.

Table 1. Various types of regulatory interactions in ERMer

Classification	Number	Sources
Chemical protein interaction (CPI)	7067	BRENDA, RegulonDB, and STITCH
Transcriptional factor regulation (TFGI)	4734	RegulonDB
Sigma factor regulation (SFGI)	2352	RegulonDB
sRNA regulation (sRGI)	145	RegulonDB
Protein-protein interaction (PPI)	9102	STRING
Reaction metabolite interaction (RMI)	3163	iML1515
Metabolite reaction interaction (MRI)	3096	iML1515
Pathway reaction interaction (PRI)	2375	iML1515
Gene reaction interaction (GRI)	4297	iML1515

the graph database. Finally, all the information, including nodes, edges and attributes, can be presented in a graph with the G6 graph visualization engine. By integrating the AWS Neptune graph database with the serverless AWS lambda function and frontend G6 graph visualization engine, ERMer facilitates end-users to search, visualize and navigate our graph database without the need to write any querying program.

Main functionality of ERMer

Interactive search. Interactive search enables interactive exploration of the regulatory landscape by recursive search of interacting partners of nodes. ERMer shows all relevant regulatory interactions for a query subject. Besides, the user

can choose a specific node in the graph and search its related interactions again. In this way, users can interactively explore the regulatory landscape. ERMer also provides the option to choose a subset of types of interactions in the search. This can be useful when the users want to exclude specific interactions without a clear function (e.g. PPI) in their search.

Taking Glycine as an example, when clicking the ‘Search’ button, the user can get the nodes directly interacting with it (Figure 2A). ERMer allows access to associated information on genes, metabolites, reactions, and pathways. The detailed information of a neighbor, for example, gene *gcvA*, can be obtained by clicking the node ‘*gcvA*’, such as Name (*gcvA*), BiGG ID (b2808), Swiss-Prot ID (P0A9F6), and detailed function. Right-clicking on a node to select ‘interactive search’ will bring up a hovering window for selecting the edge types (Figure 2B) for further navigation. Subsequently, the new interacting graph is presented to the user. Users can recursively select new nodes of interest to explore the complex regulatory relationships thoroughly. Figure 2C shows a path between Glycine and *gcvT* after ‘interactive search’ four times.

Regulation search. It is well known that many biosynthetic pathways contain feedback regulation mechanisms where the activity levels of the key enzymes are regulated by the corresponding end-products (23). Such regulatory mechanisms are essential for cells to be robust in varying environments. The regulation search function in ERMer enables the mining of regulatory cascades between any metabolite/gene and gene in two modes: shortest path or all paths.

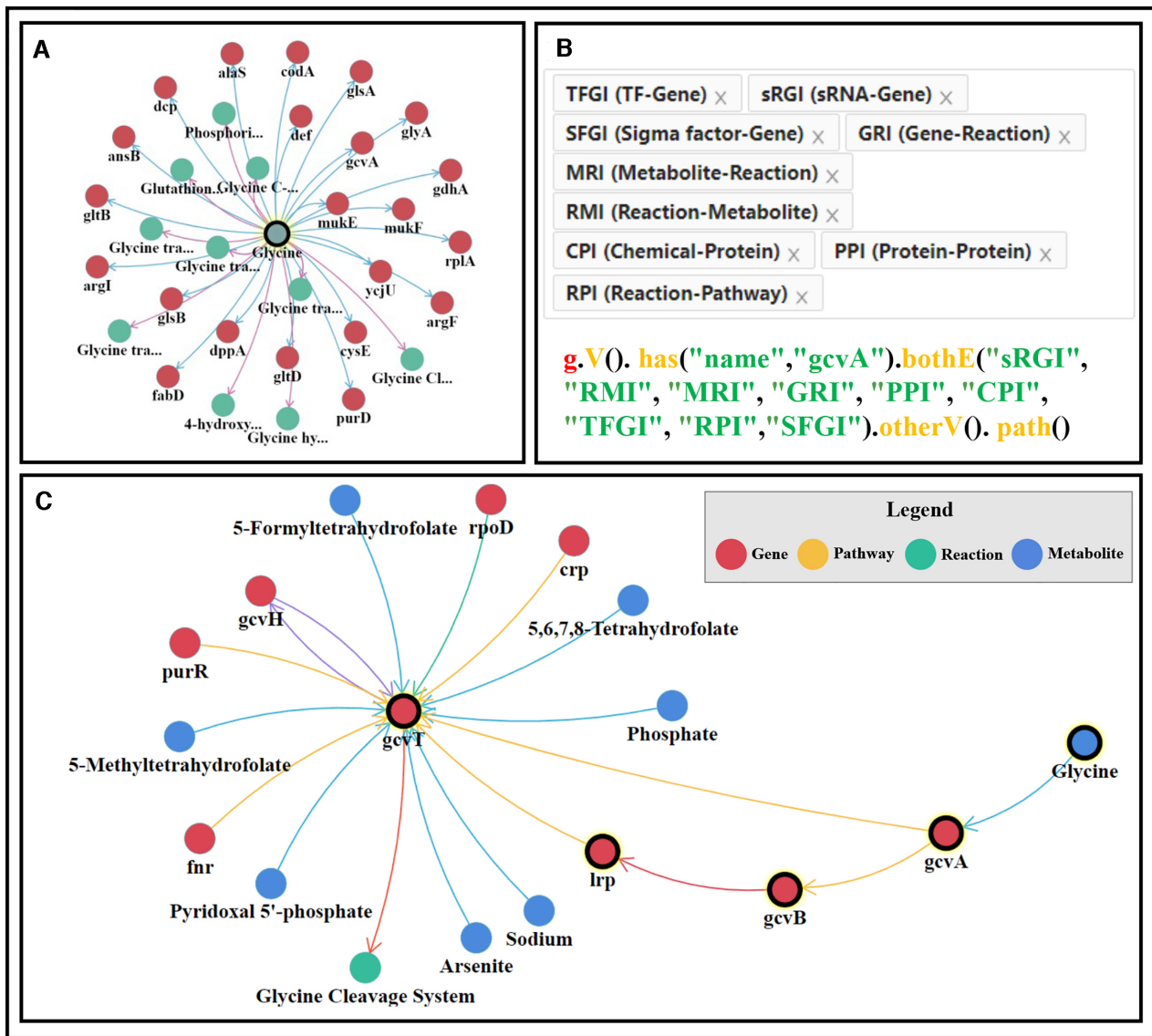


Figure 2. Interactive search. (A) The first order neighbors of glycine; (B) Interactive search setting and the corresponding gremlin query; (C) four times 'Interactive search' find a path between Glycine and gcvT.

A key advantage of ERMer is that it allows the users to easily find all complex regulatory cascades comprising different types of regulatory interactions. Taking glycine cleavage system as an example, in addition to the well-known regulatory cascade glycine-gcvA-gcvT, a new regulatory cascade glycine-gcvA-gcvB-lrp-gcvT can also be found in ERMer (Figure 3), which involves CPI, TFGI and sRGI. ERMer retrieves all regulatory cascades between glycine and gcvT for maximum 7 steps in a very straightforward way using the following gremlin query 'g.V().has('name','glycine').repeat(outE('CPI','TFGI','SFGI','sRGI','PPI').inV().simplePath()).emit().times(7).has('name','gcvT').path()'. More specifically, g.V().has('name','glycine') specifies the source node glycine, outE('TFGI','SFGI','sRGI','PPI') specifies the edge types to be included, times(7) stands for the maximum step. In the case of relational databases, the search is more

complex even with virtual tables and recursive search using PostgreSQL (Figure S1), and the response time is much longer than that of graph databases (840s versus 1.79s).

In addition to the glycine-gcvA-gcvB-lrp-gcvT cascade, other complex regulatory cascades can also be obtained (Figure 3). The existence of multiple feedback regulatory loops implies that just interrupting one loop may not be enough to break down the feedback control for improving the synthesis of the end product. Besides, these interactions can help users discover some new regulatory patterns that have not yet been studied, which can assist the design of genetic circuits (24) or enable researchers to design multiple specific dynamic regulatory systems (25). In addition, filtering based on various criteria is also provided. For example, the number of PPI or CPI, containing or not containing a particular node/edge, can be used for filtering. The filtered regulatory cascades will be shown in the table and the map

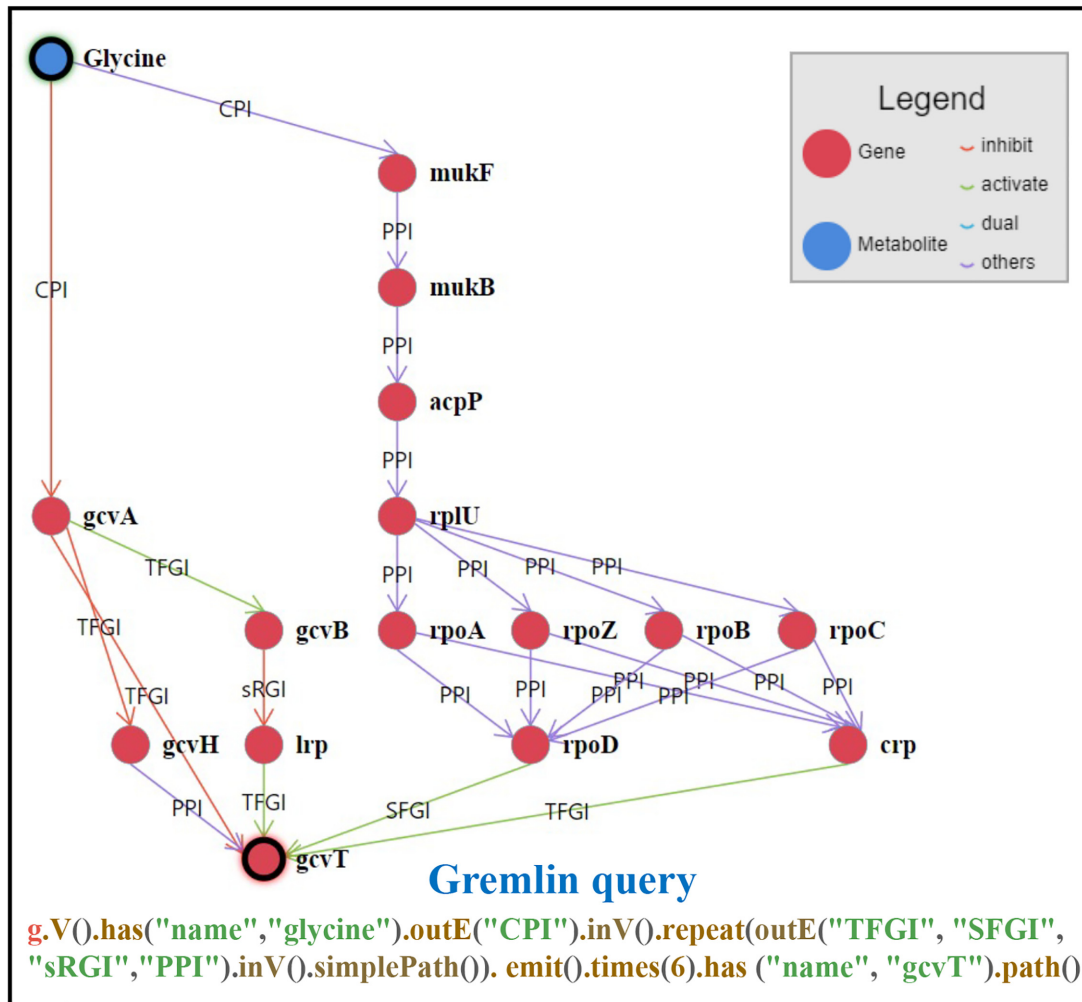


Figure 3. Regulatory cascades from glycine to gcvT using graph databases.

will be redrawn after clicking the ‘DRAW’ button (Figure S2).

Q&A search. Q&A is implemented to retrieve related information in a natural language way, which can provide answers to various biological questions. Several biological questions are predefined in ERMer to showcase the power of complex searches using graph databases. For example, for the question ‘What are the key TFs regulating genes in both pathways?’, if pathways ‘Glycine and Serine Metabolism’ and ‘Citric Acid Cycle’ are selected, this will invoke an intuitive gremlin query ‘g.V().has(‘name’, ‘Glycine and Serine Metabolism’).inE(‘RPI’).outV().inE(‘GRI’).outV().inE(‘TFGI’).outV().outE(‘TFGI’).inV().outE(‘GRI’).inV().outE(‘RPI’).inV().has(‘name’, ‘Citric Acid Cycle’).path()’. Five TFs can be found affecting genes expressed in both pathways (Figure 4). Although these five TFs are all global transcription factors, we found that the regulatory pattern differed. Take Crp as an example, for both pathways, activation is dominant, but the ratio of activation/repression in the ‘Glycine and Serine Metabolism’ pathway (6:1) was higher than that of the ‘Citric Acid Cycle’ (10:5) (Figure S3). In addition to TFs, we also provide searches for im-

portant sRNAs, metabolites, and Sigma factors affecting both pathways (supplementary material, Figures S4-S6).

ERMer can easily mine the regulatory relationships between multiple regulators and multiple pathways. Multiple pathways can be chosen for the question ‘What are the key TFs regulating genes across pathways?’. For example, if all pathways are chosen, it will invoke another gremlin query ‘g.V().has(‘label’, ‘pathway’).inE(‘RPI’).outV().inE(‘GRI’).outV().inE(‘TFGI’).outV().outE(‘TFGI’).inV().outE(‘GRI’).inV().outE(‘RPI’).inV().has(‘label’, ‘pathway’).path()’. By using has(‘label’, ‘pathway’), the search can be started and ended with many pathways, which is another key advantage of using graph databases. For this query, using traditional databases is very inefficient even with the aid of graph analysis tools as it requires nested for loops. To make the relationships between TFs and pathways clearer, a path between a specific TF and a pathway is aggregated to a direct edge between them in Figure S7. 137 TFs were found to affect genes expressed in at least two pathways. In addition to the familiar global TFs (e.g. Crp, Fnr, IHF, Fis, ArcA, and Lrp) (26,27), ERMer shows other TFs, such as PdhR, CpxR, Cra, Hns, and SoxS, are also very important

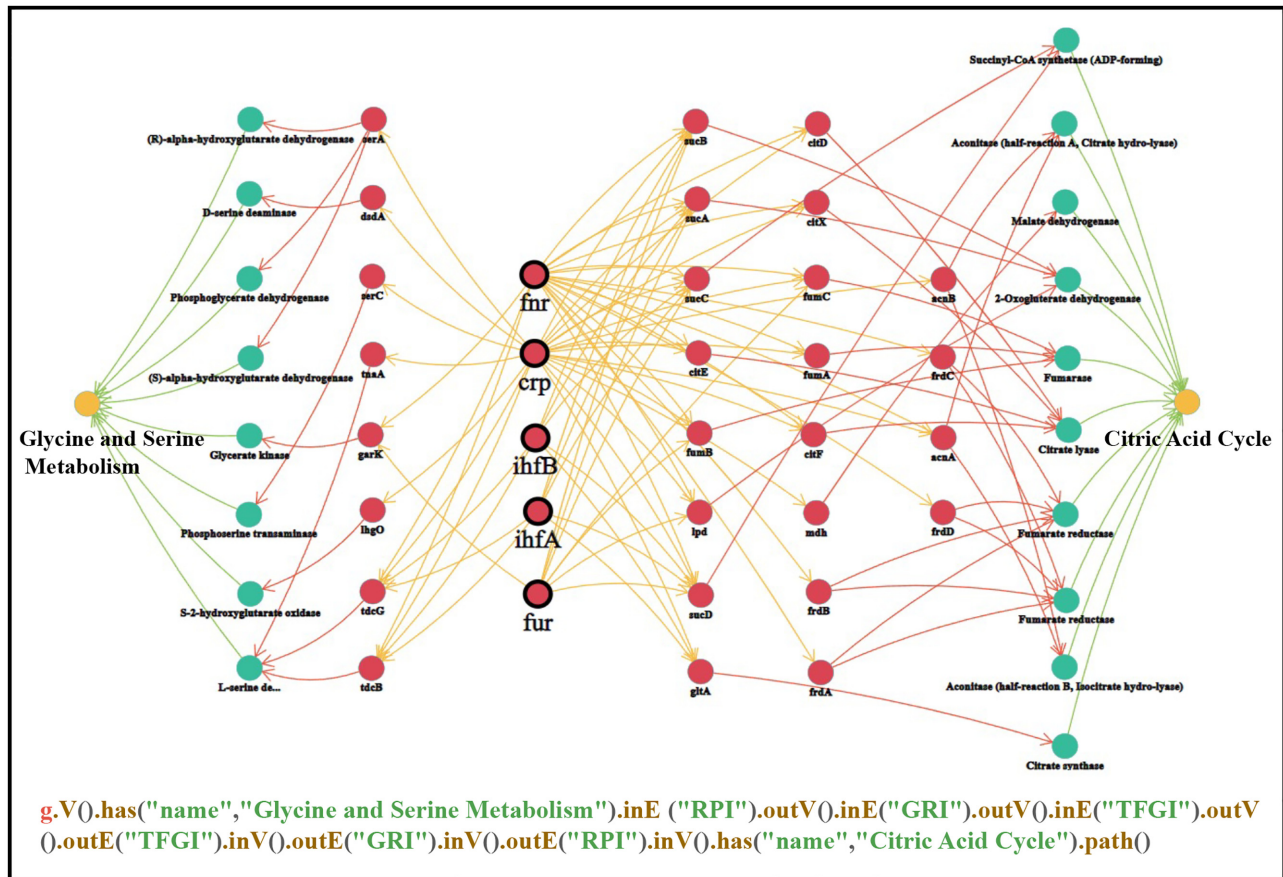


Figure 4. Retrieval of key TFs regulating both pathways.

as they affected genes expressed in at least 11 pathways. Besides, ERMer also provides the top 10 of TFs, sRNAs, and metabolites with the most regulatory targets in *E. coli* (Figure S8), and gives a hierarchical map of the *E. coli* TF–TF network (Figure S9).

DISCUSSION

ERMer is the first cloud platform offering an overview of the regulatory landscape of *E. coli* based on a graph database approach. The two modules, graphdb_builder, and graphdb_loader, enable ERMer to automatically acquire, process, and import them into the graph database. This approach ensures that ERMer is extensible for the data, and ERMer can automatically fetch and update the graph database when the source database is updated. ERMer offers an interactive way for the end-users to interactively explore the regulatory landscape of *E. coli*, which is one of our unique features. In addition, ERMer can rapidly mine regulatory cascades between metabolites (or genes) and genes in *E. coli* using efficient gremlin queries, which can help users discover new regulatory patterns and uncover meaningful regulatory strategies for developing production strains. Moreover, ERMer provides the end-users with other ways to interact with the graph databases using just human-readable biological questions. ERMer shows

for the first time a whole picture of the TF–pathways network, and some biological insight can be inferred (Figure S7). Overall, ERMer enables effective interactive navigation and visualization, which can help researchers uncover the complete regulatory map, showing the great potential by using the graph database.

Although ERMer was mainly designed for *E. coli* regulation mining, the framework implemented in ERMer is of general use and applicable to other applications or organisms. For example, designing transcription-factor-based biosensors (TFBs) with superior performance for applications in synthetic biology remains a significant challenge (28). To mine endogenous TFs in *E. coli*, we can use the same framework and slightly change the graph database and the function. More specifically, new modules of graphdb_builder are needed to collect promoters and related information. And the backend graph database schema should be modified to include the promoter node, TF–promoter, and promoter–gene edges. Finally, the gremlin API and functionality of ERMer (e.g. regulatory cascades retrieval function) need to be fine-tuned. Some other analyzing modules can also be incorporated. For example, Amazon Neptune ML, a new capability of Neptune that uses Graph Neural Networks (GNNs, a machine learning technique purpose-built for graphs), can be integrated into our platform in the future for tasks such as TF prediction

and TF–target predictions. We expect that others will be encouraged to adopt and further improve our framework for their applications.

DATA AVAILABILITY

The code and sample dataset of ERMer are available in the Github repository (<https://github.com/tibbdc/ermer>).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

FUNDING

National Key Research and Development Program of China [2020YFA0908300]; Tianjin Synthetic Biotechnology Innovation Capacity Improvement Project [TSBICIP-PTJS-001]; Youth Innovation Promotion Association CAS. Funding for open access charge: Tianjin Synthetic Biotechnology Innovation Capacity Improvement Project [TSBICIP-PTJS-001].

Conflict of interest statement. None declared.

REFERENCES

- Xia,F., Sun,K., Yu,S., Aziz,A., Wan,L., Pan,S. and Liu,H. (2021) Graph learning: a survey. *IEEE Trans. Artif. Intell.*, **2**, 109–127.
- Fernandes,D. and Bernardino,J. (2018) Graph databases comparison: allegrograph, ArangoDB, infinitegraph, Neo4J, and OrientDB. *Data*, **373–380**.
- Ahmadi,Z., Parand,F.-A. and Matinfar,F. (2022) A fuzzy logic-based approach for fuzzy queries over NoSQL graph database. *Concurr. Comp-Pract. E.*, **34**, e6542.
- Miller,J.J. (2013) Graph database applications and concepts with neo4j. *SAIS 2013 Proceedings*. Vol. **24**, <https://aisel.aisnet.org/sais2013/24>.
- Pivert,O., Smits,G. and Thion,V. (2015) Expression and efficient processing of fuzzy queries in a graph database context. *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. pp. 1–8.
- Summer,G., Kelder,T., Ono,K., Radonjic,M., Heymans,S. and Demchak,B. (2015) cyNeo4j: connecting neo4j and cytoscape. *Bioinformatics*, **31**, 3868–3869.
- Toure,V., Mazein,A., Waltemath,D., Balaur,I., Saqi,M., Henkel,R., Pellet,J. and Auffray,C. (2016) STON: exploring biological pathways using the SBGN standard and graph databases. *BMC Bioinf.*, **17**, 494.
- Balaur,I., Mazein,A., Saqi,M., Lysenko,A., Rawlings,C.J. and Auffray,C. (2017) Recon2Neo4j: applying graph database technologies for managing comprehensive genome-scale networks. *Bioinformatics*, **33**, 1096–1098.
- Swainston,N., Batista-Navarro,R., Carbonell,P., Dobson,P.D., Dunstan,M., Jervis,A.J., Vinaixa,M., Williams,A.R., Ananiadou,S., Faulon,J.L. *et al.* (2017) biochem4j: integrated and extensible biochemical knowledge through graph databases. *PLoS One*, **12**, e0179130.
- Fabregat,A., Korninger,F., Viteri,G., Sidiropoulos,K., Marin-Garcia,P., Ping,P., Wu,G., Stein,L., D’Eustachio,P. and Hermjakob,H. (2018) Reactome graph database: efficient access to complex pathway data. *PLoS Comput. Biol.*, **14**, e1005968.
- Santos,A., Colaco,A.R., Nielsen,A.B., Niu,L., Strauss,M., Geyer,P.E., Coscia,F., Albrechtsen,N.J.W., Mundt,F., Jensen,L.J. *et al.* (2022) A knowledge graph to interpret clinical proteomics data. *Nat. Biotechnol.*, <https://doi.org/10.1038/s41587-021-01145-6>.
- Szklarczyk,D., Gable,A.L., Nastou,K.C., Lyon,D., Kirsch,R., Pyysalo,S., Doncheva,N.T., Legeay,M., Fang,T., Bork,P. *et al.* (2021) The STRING database in 2021: customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res.*, **49**, D605–D612.
- Szklarczyk,D., Santos,A., von Mering,C., Jensen,L.J., Bork,P. and Kuhn,M. (2016) STITCH 5: augmenting protein-chemical interaction networks with tissue and affinity data. *Nucleic Acids Res.*, **44**, D380–D384.
- Chang,A., Jeske,L., Ulbrich,S., Hofmann,J., Koblitz,J., Schomburg,I., Neumann-Schaal,M., Jahn,D. and Schomburg,D. (2021) BRENDA, the ELIXIR core data resource in 2021: new developments and updates. *Nucleic Acids Res.*, **49**, D498–D508.
- Keseler,I.M., Gama-Castro,S., Mackie,A., Billington,R., Bonavides-Martinez,C., Caspi,R., Kothari,A., Krummenacker,M., Midford,P.E., Muniz-Rascado,L. *et al.* (2021) The ecocyc database in 2021. *Front. Microbiol.*, **12**, 711077.
- Santos-Zavaleta,A., Salgado,H., Gama-Castro,S., Sanchez-Perez,M., Gomez-Romero,L., Ledezma-Tejeida,D., Garcia-Sotelo,J.S., Alquicira-Hernandez,K., Muniz-Rascado,L.J., Pena-Loredo,P.Ishida-Gutierrez *et al.* (2019) RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in *E. coli* K-12. *Nucleic Acids Res.*, **47**, D212–D220.
- Mao,Z., Huang,T., Yuan,Q. and Ma,H. (2022) Construction and analysis of an integrated biological network of *Escherichia coli*. *Syst. Microbiol. Biomanuf.*, **2**, 165–176.
- Pio,G., Ceci,M., Malerba,D. and D’Elia,D. (2015) ComiRNet: a web-based system for the analysis of miRNA-gene regulatory networks. *BMC Bioinf.*, **16**, S7.
- Hagberg,A., Swart,P. and S Chult,D. (2008) Exploring network structure, dynamics, and function using networkX. In: Varoquaux,G., Vaught,T. and Millman,J. (eds) *Proc. SciPy 2008*. pp. 11–16.
- Mrvar,A. and Batagelj,V. (2016) Analysis and visualization of large networks with program package pajek. *Complex Adapt. Syst. Model.*, **4**, 6.
- Monk,J.M., Lloyd,C.J., Brunk,E., Mih,N., Sastry,A., King,Z., Takeuchi,R., Nomura,W., Zhang,Z., Mori,Feist *et al.* (2017) iML1515, a knowledgebase that computes *Escherichia coli* traits. *Nat. Biotechnol.*, **35**, 904–908.
- King,Z.A., Lu,J., Drager,A., Miller,P., Federowicz,S., Lerman,J.A., Ebrahim,A., Palsson,B.O. and Lewis,N.E. (2016) BiGG models: a platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res.*, **44**, D515–D522.
- Park,J.H., Lee,K.H., Kim,T.Y. and Lee,S.Y. (2007) Metabolic engineering of *escherichiacoli* for the production of L-valine based on transcriptome analysis and in silico gene knockout simulation. *Proc. Natl. Acad. Sci. U.S.A.*, **104**, 7797–7802.
- Nielsen,A.A., Der,B.S., Shin,J., Vaidyanathan,P., Paralanov,V., Strychalski,E.A., Ross,D., Densmore,D. and Voigt,C.A. (2016) Genetic circuit design automation. *Science*, **352**, aac7341.
- Zhang,F., Carothers,J.M. and Keasling,J.D. (2012) Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids. *Nat. Biotechnol.*, **30**, 354–359.
- Martinez-Antonio,A. and Collado-Vides,J. (2003) Identifying global regulators in transcriptional regulatory networks in bacteria. *Curr. Opin. Microbiol.*, **6**, 482–489.
- Kargeti,M. and Venkatesh,K.V. (2017) The effect of global transcriptional regulators on the anaerobic fermentative metabolism of *escherichiacoli*. *Mol. Biosyst.*, **13**, 1388–1398.
- Ding,N., Zhou,S. and Deng,Y. (2021) Transcription-Factor-based biosensor engineering for applications in synthetic biology. *ACS Synth. Biol.*, **10**, 911–922.