

DORMAN: Database of Reconstructed Metabolic Networks

Furkan Ozden, Metin Can Siper^{ID}, Necmi Acarsoy, Tugrulcan Elmas^{ID}, Bryan Marty, Xinjian Qi, and A. Ercument Cicek^{ID}

Abstract—Genome-scale reconstructed metabolic networks have provided an organism specific understanding of cellular processes and their relations to phenotype. As they are deemed essential to study metabolism, the number of organisms with reconstructed metabolic networks continues to increase. This everlasting research interest lead to the development of online systems/repositories that store existing reconstructions and enable new model generation, integration, and constraint-based analyses. While features that support model reconstruction are widely available, current systems lack the means to help users who are interested in analyzing the topology of the reconstructed networks. Here, we present the Database of Reconstructed Metabolic Networks - DORMAN. DORMAN is a centralized online database that stores SBML-based reconstructed metabolic networks published in the literature, and provides web-based computational tools for visualizing and analyzing the model topology. Novel features of DORMAN are (i) interactive visualization interface that allows rendering of the complete network as well as editing and exporting the model, (ii) hierarchical navigation that provides efficient access to connected entities in the model, (iii) built-in query interface that allow posing topological queries, and finally, and (iv) model comparison tool that enables comparing models with different nomenclatures, using approximate string matching. DORMAN is online and freely accessible at <http://ciceklab.cs.bilkent.edu.tr/dorman>.

Index Terms—Genome-scale reconstructed metabolic networks, metabolism, metabolomics, online workbench

1 INTRODUCTION

WITH the advancements in the omics platforms and the availability of affordable high throughput data, researchers have been able to capture the genome-scale chemical composition of the cell and integrate this knowledge into genome-scale reconstructed metabolic networks of organisms [1]. Reconstructed models have proven to be indispensable tools for understanding the metabolism in a diverse spectrum of applications such as: (i) metabolic engineering, (ii) model-directed discovery, (iii) interpretations of phenotypic screens, (iv) analysis of network properties, and (v) studies of evolutionary processes [2]. Also many studies have made use of the topology of the networks for understanding disease mechanisms [3], [4], [5], [6], [7], [8], [9], [10]. Ongoing interest in network reconstruction and analyses comes with the need for computational tools to work on the resulting models. There are several online systems, which store existing reconstructions and provide interfaces for

model generation, integration and constraint-based analyses. GSMNDB is a legacy repository with a basic HTML interface that stored published models in the literature (system is offline and not maintained anymore). BiGG database is one of the first online repositories, which contains 10 models [11]. Basic browsing of models, SVG-based visualization of pathways and export functions are provided. BiGG database is upgraded to BiGG Models, which now includes standardized identifiers for reactions and metabolites [12]. This new system comes with Escher pathway maps-based visualization and an application programming interface (API) for data access. Model SEED is a contemporary of the early BiGG database. In addition to being a model repository for a small number of seed models, it is also a workbench for model generation (on top of the seeds), model analysis (e.g., gene essentiality) and model optimization [13]. Model SEED was followed by MemoSys [14]. MemoSys is a storage and model generation platform that provides basic browsing, model comparison and version control functionalities.

One shortcoming of earlier repositories was the lack of standardized metabolite and reaction naming in models. When also accompanied with lack of annotation of entities (i.e., metabolite names not annotated with KEGG compound ids), comparison of networks and reuse of existing information was almost impossible. Next line of tools came with approaches to also address this issue. MetRxn is a curated system with browsing/search functionalities and it provides a knowledgebase that standardizes metabolite and reaction names [15]. MetaNetX is another system whose goal is enabling automated reconciliation of model

- F. Ozden, M. C. Siper, N. Acarsoy, and T. Elmas are with the Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey. E-mail: furkan.ozden@bilkent.edu.tr, metincansiper@gmail.com, necmi.acarsoy@gmail.com, tugrulcan.elmas@epfl.ch.
- B. Marty and X. Qi are with the EECS Department, Case Western Reserve University, Cleveland, OH 44106 USA. E-mail: bryanmarty@gmail.com, xxq21@case.edu.
- A. E. Cicek is with the Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey, and also with the Computational Biology Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA. E-mail: cicek@cs.bilkent.edu.tr.

Manuscript received 14 Dec. 2018; revised 30 Aug. 2019; accepted 26 Sept. 2019. Date of publication 1 Oct. 2019; date of current version 6 Aug. 2021.

(Corresponding author: A. E. Cicek.)

Digital Object Identifier no. 10.1109/TCBB.2019.2944905

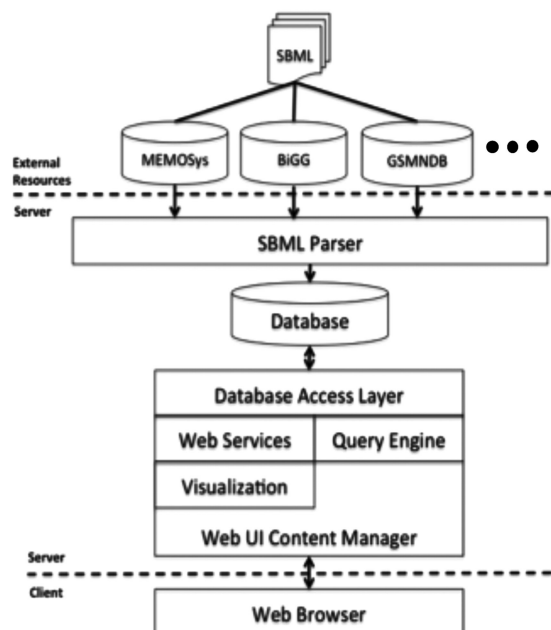


Fig. 1. Architecture of DORMAN is shown. SBML files of reconstructed metabolic networks are gathered from external resources to populate the database using the SBML parser. Two-tiered system with a thin client pushes most of the computation to the server-side (i.e., query processing etc.). Browser interfaces with the Web UI Content manager on the server side via web services and mostly with AJAX calls. The database is accessed via a Database Access Layer that uses wrapper classes to model the database at the program level.

metabolites and reactions [16], [17], [18]. They provide a namespace called MNXRef [18] for this purpose. System provides computational tools like Flux Balance Analysis (FBA), Flux Variability Analysis (FVA), Gene/Reaction/Peptide knock-out analysis and creation/modification of models.

Above-mentioned systems have been frequently used by researchers (i) as model repositories and (ii) as workbenches for building and then for analyzing models using constraint-based analyses (i.e., FBA, FVA etc.) So, they are not designed for users who are interested in the topology of networks despite the fact that there are also numerous studies that focus on that aspect. For instance, [19] shows that genes that are related to highly connected reactions in metabolic networks are subject to stricter evolutionary constraints. Another example is [20] that uses motifs in a metabolic network to study the evolutionary origin of six Eukaryotic organelles. Finally, *Reporter Algorithm* [3] detects biomarker metabolites, which have the highest dysregulation of gene expression in their direct neighborhood with respect to a phenotype. Users who are using reconstructions for such purposes would naturally like to explore the subnetworks of interest via flexible browsing, querying and visualization functionalities.

In this paper, we present the Database of Reconstructed Metabolic Networks (DORMAN); a central database that collects available genome-scale reconstructed metabolic networks from the literature and provides a user friendly and efficient platform for accessing, visualizing and querying the models with multiple interfaces. While above mentioned systems are mostly focused on the integration process as explained above, DORMAN provides novel complementary

features, which lets users to analyze the topology of the networks. First of all, current systems provide either no or only pathway-level static visualization (i.e., Escher maps of BiGG) while DORMAN's visualization tool can render the full network (not just pathways) with compartments and lets users to interact with the model (e.g., zoom in/out; move, add or remove nodes in the network). Unlike its counterparts, hierarchical navigation feature of DORMAN allows users to efficiently browse through connected entities in the model, level-by-level. For instance users can start browsing the reactions of a model, and then can list and select the metabolites of a selected reaction, and finally, can list the compartments associated with the selected metabolites. The browser interface is integrated with KEGG data and can be used to navigate whenever KEGG pathway or KEGG molecule data is made available by the model. External databases such as UniProt, ChEBI and KEGG are linked when the information is provided in the reconstruction. DORMAN's built-in graph queries can be used to search for topologically related entities in the graph such as searching for entities in k-hop neighborhoods of reactions or metabolites. DORMAN also provides an interface to compare models that do not follow the same nomenclature, using approximate name matching of the entities. Currently, the database contains 199 SBML-based models obtained from external repositories and is continuously updated by screening the literature and available repositories.

Rest of the manuscript is organized as follows: In Section 2, we give details about the implementation; in Section 3, interfaces of the system are explained. In Section 4, we discuss the limitations of the system and finally, in Section 5, we conclude.

2 IMPLEMENTATION

The system uses two-tiered client-server software architecture. An overview of the architecture is shown in Fig. 1. On the server side, we have (i) the database, (ii) the SBML Parser that populates the database, and (iii) the web server. The client side can be considered as a thin client that can be accessed via any up-to-date web browser.

The database is populated using the SBML parser that relies on libSBML library [21]. The models are obtained from other third party repositories: BiGG Models, Model Seed, MetRxn, MemoSys, MetaNetX, and legacy GSMNDB site. The parser is implemented in C# language, and in .NET environment. For the system to scale to a large number of reconstructed networks and to avoid human errors, we do not curate original files, and population of the database is fully automated. We use Microsoft SqlServer to host the database.

The browsing interface is also implemented in C# language, using ASP.NET. Web Content UI Manager handles all synchronous and asynchronous requests coming from users and prepares the response files by interacting with the back-end systems. The browser is based on the PathCase library [22], [23], [24], [25]. It now provides 6 new reconstructed-metabolic-network-specific interfaces: model comparison, network (topological) properties, species and reaction locator, Flux Balance Analysis (FBA), Flux Variance Analysis (FVA) and scalable Javascript based visualization tool.

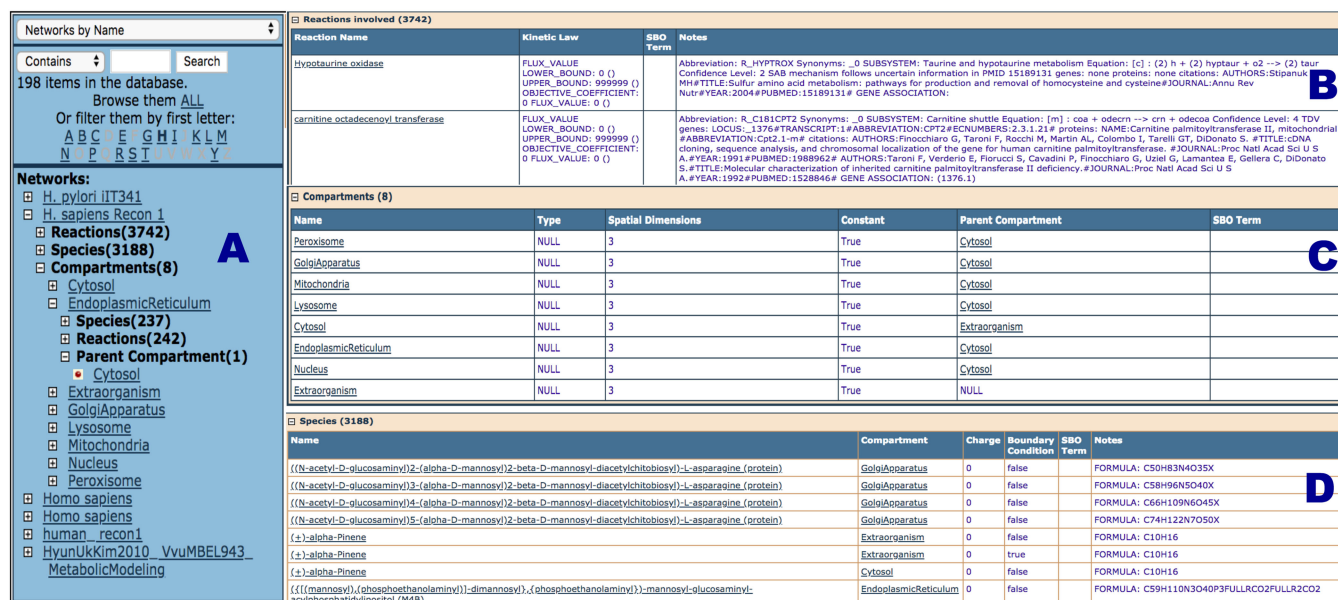


Fig. 2. Browser interface of DORMAN. The navigation bar shows available models in the database. On top, users can search the database by text, and select networks by the initial letters. In this example, H. Sapiens Recon 1 model is selected and the hierarchy of related entities is shown underneath: Compartments (in H. Sapiens Recon 1) > Endoplasmic Reticulum (of H. Sapiens Recon 1) > Parent Compartment (of Endoplasmic Reticulum of H. Sapiens Recon 1). Part B. Once the network is clicked, all model related interfaces (panels) are loaded onto the main-frame. This example shows the panel that lists the reactions of H. Sapiens Recon 1. Part C. List of compartments for H. Sapiens Recon 1 is shown on the main frame. Part D. List of metabolites for of H. Sapiens Recon 1 are shown in this panel.

For the visualization of the networks, we implemented a Javascript-based interface that uses Cytospace.js library in the background [26]. Rendering is done on the client side, which again enables the system to scale to a large number of service requests. The data is passed to the visualization interface using the Web Services sub-component that responds in XML format. Query Engine is responsible for processing of the built-in queries and acts as an add-on to the Database Access Layer (DAL). DAL handles all database interactions via wrapper classes and receives the requests from the Web Services layer.

3 RESULTS

3.1 Browsing the Models via the Web Interface

The web interface has two parts: The navigation bar on the left (as seen on Fig. 2 Part A), and the main frame on the right (Fig. 2, Parts B-D that show partial screenshots of the main frame). Using the navigation bar, users can (1) browse the available networks by name, (2) browse reactions by name and EC Numbers, (3) browse metabolites by name, and finally, (4) browse compartments by name.

Navigation bar links all related entities together. This hierarchical and flexible tree structure lets users to browse through related entities with ease. For instance, once H. Sapiens Recon 1 model is toggled as shown in Fig. 2 (Part A), all reactions, metabolites and compartments in this model are listed below, with corresponding counts in parentheses. Users can then browse in the hierarchy. For instance, in the same figure, the *compartments* item is toggled to find all compartments of H. Sapiens Recon 1 model. When *Endoplasmic Reticulum* (ER) compartment is toggled, metabolites, reactions, and the parent compartment of ER are displayed. Similar hierarchies are provided for reactions, metabolites and compartments as well. Toggling an item initiates

asynchronous calls to the server. Hence, only the data for the desired item is fetched; thus, the users only need to wait for the data they request; and they can continue to interact with other parts of the page.

Whenever an item (i.e., model, reaction etc.) is clicked on the navigation bar, all associated interfaces (i.e., visualization) available for that item is loaded onto the main frame in separate panels. Each interface has its own independent panel and this modular design enables DORMAN to easily incorporate new features without affecting the previous design or functionality.

In Fig. 2, parts B-D show three collapsible panels loaded to the main frame for the H. Sapiens Recon 1 model after it is clicked on the navigator. For models, the following panels are loaded to the main frame: (1) visualization panel, (2) reactions list panel as shown in Fig. 2 - Part B, (4) compartments list panel as shown in Fig. 2 - Part C, (5) species list panel as shown in Fig. 2 - Part D, (6) a panel that lists the defined units in the network, (7) a panel that lists the FBA result for the model (if applicable), (8) a panel that provides the FVA result for the model, (9) a panel that lists topological properties of the model, and finally (10) a panel that lists all available network related queries. For a reaction, system loads various collapsible panels such as: kinetic laws, stoichiometry equation, EC number, SBO Term, list of metabolites playing a role in this reaction, and reaction related queries. Species and compartments have similar content in their details pages. Visualization interface is available only for the models (see Section 3.2 for details).

3.2 Interactive Visualization Interface

For large network such as genome-scale reconstructed metabolic networks, full-scale visualization can become busy with thousands of metabolites and reactions rendered. This

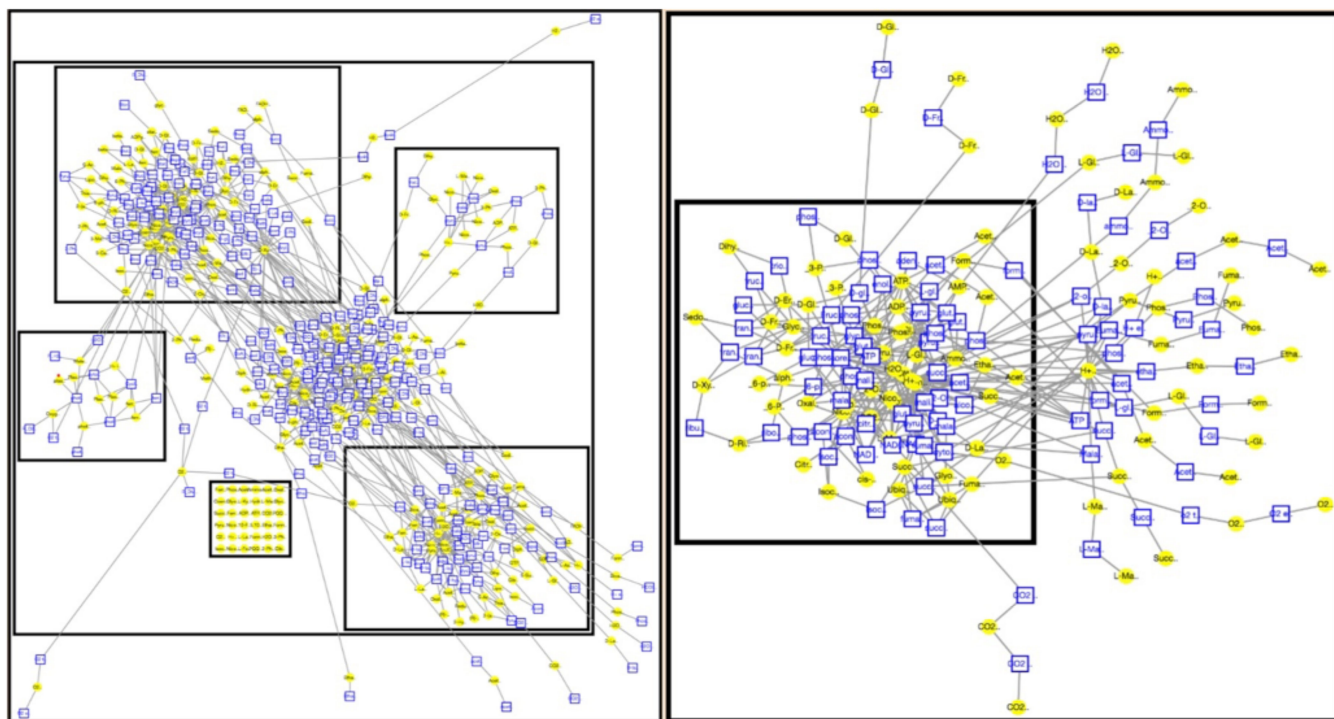


Fig. 3. Visualizations of the iAM303 model (left) and E.Coli textbook model (right) using the Javascript-based visualization interface. The layout is computed using the BilkentCoSE layout algorithm. Interface displays the full model with the compartment hierarchy whenever provided. Yellow circles represent metabolites and blue rectangles represent metabolic reactions. Users can move nodes/compartments over and isolate subnetworks of interest. They can also add/remove reactions/species and export the underlying model with the button above the interface.

is mainly because the layout algorithms are not designed to scale up to the size of genome scale metabolic networks. Among the counterparts of DORMAN, only BiGG Models provides a visualization interface and it opts for pathway-only (Escher map based) visualization, which is pre-rendered and static. However, this is restrictive and user cannot see the full picture of the model. We argue that despite the size, a full-scale visualization metabolic pathways is still valuable. One successful example is Roche's wall-sized *biochemical pathways*, which have been used by researchers for 50 years in hard copy. It is now available online at <http://biochemical-pathways.com>.

For the first time, DORMAN provides a visualization interface for reconstructions, which renders the complete model (not just pathways) with compartments. System also allows interaction and lets users to zoom in/out, move nodes (reactions, compartments, metabolites, edges). Users can add/remove reactions and metabolites to the currently displayed network and then save the updated network to their disk in SBML format. This data is not saved on the server side and lost if user does not save the SBML. Fig. 3 shows the visualizations of the *iAM303* and *E. Coli Textbook* models with this interface. Bilkent CoSE algorithm is used for calculating the layout [27], which, in our experience, provides the best layout for large scale networks.

3.3 Query Interface

DORMAN comes with built-in queries that enables users to search models for certain metabolites or as topological questions like finding reactions/metabolites that are in n-hop distance to other reactions/metabolites. The query interface is empowered with dynamic calls to the server, so that

whenever user picks a category (e.g., network) from a dropdown box, the items in the next dropdown box are populated based on the previous selection (e.g., if a reaction is to be picked, only reactions in that network is provided). The system is also integrated with the KEGG data. Whenever the information is provided in the model, entities in the model are linked to KEGG entities (i.e., species) and queries related to only KEGG pathways can be answered. Specifically, the following queries are available in the system.

Network related queries:

1. Find reconstructed networks having metabolite in a given domain (compartment, organism)
2. Find reconstructed networks and their compartments containing a given metabolite in a given organism

Reaction related queries:

1. Find reactions (and their kinetic equations if exist) that are n-step downstream/upstream from the reaction of in a given network
2. Find reactions within a given number of steps from a reaction in a KEGG pathway
3. Find reactions within a given number of steps from a metabolite in a KEGG pathway
4. Find reactions within a given number of steps from a reaction in a given metabolic network (model)
5. Find reactions sharing activators and inhibitors with a reaction in a pathway
6. Find reactions with the given number of molecules in a specific use
7. Find reactions involving exactly one substrate and one product

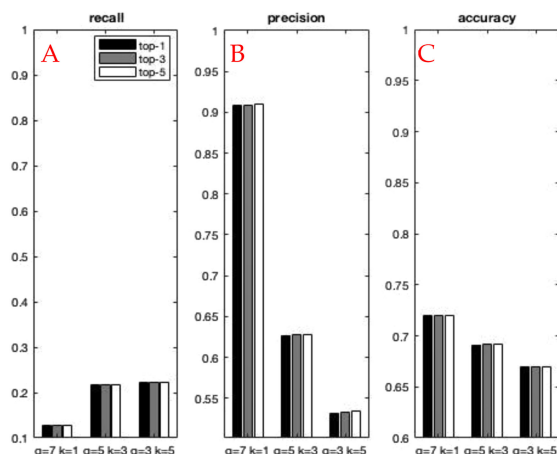


Fig. 5. Panels A, B, and C show the performance of the Compare Networks tool on the following four models: IRC1080, E.coli iAF1260, E.coli iJR904 and H. sapiens Recon 1. MetaNetX system has name-standardized versions of these four models. We compare these models (pairwise, six comparisons) using the Compare Models tool and measure the performance using the ground truth matches in MetaNetX. Three parameter configurations are used to tune the strictness of our tool. In the strictest setting, only exact matches are returned. As parameters are relaxed, less similar pairs are also matched by the system: $q = 7, k = 1$ (the strictest setting); $q = 5, k = 3$ and $q = 3, k = 5$ (the most relaxed setting). Recall (Panel A), precision (Panel B), and accuracy (Panel C) results for top-1, top-3 and top-5 matches are shown. In a top-k setting, if the ground truth match is within the top-k results returned by the system, it is counted as a true positive.

relaxed. A match is a true positive if the MetaNetX mapped reaction ids of source reaction and target reaction match. A match is a false positive if none of the listed prediction pairs' MetaNetX mapped reaction ids match.

We show the performance of the system in Fig. 5, when user requests top-1, top-3, and top-5 matches. Results show that as the parameters are relaxed the accuracy and precision go down while the recall increases as expected. In all settings, the system provides high accuracy and precision while the recall is relatively low. This means the system is successful at avoiding incorrect matches and the listed pairs frequently provide a true match in these settings. However, it cannot recover all true matches as naming of two entities might differ too much. One example is the reaction pair “_4-aminobutyrate transport via diffusion (extracellular to periplasm)” and “4-aminobutanoate transport via diffusion, chloroplast”. These two reactions match according to MetaNetX's MNXRef ids but their names are quite distinct and cannot be matched by our tool.

3.6 Locating Similar Entities in Other Models

Given a metabolite/reaction name, DORMAN provides a tool for locating similar species/reactions in DORMAN's database. The system calculates a similarity score between the input species/reactions and all species/reactions in the database by applying the q-gram based approximate string matching technique as described in Section 3.5. In this tool, the user picks a similarity threshold in addition to setting the edit distance and q gram length parameters.

For example, when the user is looking for metabolites similar to “M_glyc_DASH_R_c” in DORMAN's database, and selects $k = 5, q = 3$ and similarity score threshold = 0.7, system locates “glyc_R[c]” in *C_beijerinckii_iCM926* model with

a similarity score 0.95. “M_glyc_DASH_R_c” and “glyc_R[c]” are the same metabolite with different naming conventions and DORMAN successfully matches these strings despite such poor character matching. Another example is the search for “M_fe2_c” metabolite with a more relaxed setting ($k = 3, q = 3$ and similarity score threshold = 0.7). DORMAN returns metabolites “fe2_c”, “fe2_m”, and “fe2_e” in the *Pichia_pastoris* model. “M_fe2_c” and “fe2_e” are the same metabolite in different compartments (different metabolite pools) and “M_” is a prefix, which represents metabolite. Thus, the users are able to fine-tune the number of results they get back by playing with the similarity threshold parameter, and they can locate models involving matching entities which use different nomenclature.

4 DISCUSSION AND FUTURE WORK

Reconstructed model development process requires integration of diverse data sources such as the annotated genome of the organism, existing reconstructions, and information about the individual reactions and metabolites from the literature. As the field progressed, online systems which act as seed model repositories as well as online workbenches have emerged to help out this rigorous reconstruction process and also provided constraint based analyzing capabilities. DORMAN is not designed with the goal of replacing these systems or functionalities, which have been proven very useful for the research community. Rather, the goal of DORMAN is to complement them by providing efficient and flexible browsing, visualization and querying capabilities to enable network topology analyses for researchers, which is not the primary goal of the systems in the literature.

The standardization of the nomenclature has been one of the major bottlenecks of the research community and annotation is ignored especially in earlier versions of models. Instead of annotating the data according to SBML specifications (using <annotation> tag), authors opted to include all other details in the <notes> tag for reactions and metabolites. Therefore, extracting such data in the <notes> tag becomes yet another specialized text-parsing task. While it is possible to manually fix some of these discrepancies, in DORMAN, we avoid manual intervention for two reasons: First, as the number and size of the modes increase (which is already large), manual effort would not scale. Second, interfering with domain scientists' is an error-prone process, which means it is possible to input errors while trying to fix the problems.

Another challenge is the visualization of the models. Many reconstructed networks have thousands of reactions, and their full visualizations are quite complex. While we stress the importance of such a model-wide visualization interface is an important feature for researchers current layout algorithms are designed for small pathways and mostly do not scale very well for the genome-scale reconstructed metabolic networks. Currently, we applied Bilkent-CoSE layout algorithm [24] to all networks and we are now working on automated visualization techniques that provide better visualizations for our application. Finally, at the current stage, the system provides FBA/FVA on the original network in an offline manner. One other future direction for DORMAN is to enable online analysis which enables modification of the inputs by the user.

5 CONCLUSIONS

DORMAN is up and running and freely available for the use of the research community. Ever-increasing research focus on building and analyzing genome-scale reconstructed metabolic networks require systems that enable users to browse, visualize, query and compare reconstructions for topological analyses. DORMAN fills the gap in the research field as an online and centralized database of SBML-based reconstructed metabolic networks literature and providing above-mentioned capabilities over the models.

ACKNOWLEDGMENTS

The authors would like to acknowledge the efforts of all past CWRU-DBLAB members, with special thanks to Meral Ozsoyoglu and Gultekin Ozsoyoglu. They would also like to acknowledge the contributions of all current and past Bilkent CicekLab members. This work has been supported by the Scientific and Technological Research Council of Turkey – TUBITAK grant #115C128 to AEC. F. Ozden and M.C. Siper have contributed equally.

REFERENCES

- [1] B. O. Palsson, et al., “*Systems Biology: Properties of Reconstructed Metabolic Networks*,” Cambridge, UK: Cambridge University Press, 2006
 - [2] A. M. Feist and B. O. Palsson, “The growing scope of applications of genome-scale metabolic reconstructions using *Escherichia coli*,” *Nat. Biotechnology*, vol. 26, no. 6, pp. 659–667, 2008
 - [3] K. R. Patil, et al., “Uncovering transcriptional regulation of metabolism by using metabolic network topology,” *Proc. Nat. Acad. Sci. United States America*, vol. 102, pp. 2685–2689, 2005
 - [4] H. David, et al., “Metabolic network driven analysis of genome-wide transcription data from *Aspergillus nidulans*,” *Genome Biol.*, vol. 7, no. 11, 2006, Art. no. R108
 - [5] D. S. Lee, et al., “The implications of human metabolic network topology for disease comorbidity,” *PNAS*, vol. 105, no. 29, pp. 9880–9885, 2008
 - [6] A. Zelezniak, et al., “Metabolic network topology reveals transcriptional regulatory signatures of type 2 diabetes,” *PLoS Comput. Biol.*, vol. 6, no. 4, 2010, Art. no. e1000729
 - [7] T. Hancock, et al., “Identifying neighborhoods of coordinated gene expression and metabolic profiles,” *PLoS One*, vol. 7, no. 2, 2012, Art. no. e31345.
 - [8] A. Cakmak, et al., “A new metabolomics analysis technique: Steady state metabolic network dynamics analysis,” *J. Bioinf. Comput. Biol.*, vol. 10, no. 1, 2012, Art. no. 1240003
 - [9] A. E. Cicek and G. Ozsoyoglu, “Observation conflict resolution in steady state metabolic network dynamics analysis,” *J. Bioinf. Comput. Biol.*, vol. 10, no. 1, 2012, Art. no. 1240004
 - [10] A. E. Cicek, et al., “ADEMA: An algorithm to determine expected metabolite level alterations using mutual information,” *PLoS Comput. Biol.*, vol. 9, 2013, Art. no. e1002859
 - [11] J. Schellenberger, et al., “BiGG: A biochemical genetic and genomic knowledgebase of large scale metabolic reconstructions,” *BMC Bioinf.*, vol. 11, 2010, Art. no. 213.
 - [12] Z. A. King, et al., “BiGG models: A platform for integrating, standardizing, and sharing genome-scale models,” *Nucl. Acids Res.*, vol. 44, no. D1, pp. D515–D522, 2016.
 - [13] C. S. Henry, et al., “High-throughput generation, optimization and analysis of genome-scale metabolic models,” *Nature Biotechnol.*, vol. 28, no. 9, pp. 977–982, 2010
 - [14] S. Pabinger, et al., “MEMOSys: Bioinformatics platform for genome-scale metabolic models,” *Syst. Biol.*, vol. 5, 2011, Art. no. 20.
 - [15] A. Kumar, et al., “MetRxn: A knowledgebase of metabolites and reactions spanning metabolic models and databases,” *BMC Bioinf.*, vol. 13, 2012, Art. no. 6.
 - [16] T. Bernard, et al., “Reconciliation of metabolites and biochemical reactions for metabolic networks,” *Briefings Bioinf.*, vol. 15, no. 1, pp. 123–135, 2012.
 - [17] M. Ganter, et al., “MetaNetX.org: A website and repository for accessing, analysing and manipulating metabolic networks,” *Bioinf.* vol. 29, no. 6, pp. 815–816, 2013.
 - [18] S. Moretti, et al., “MetaNetX/MNXref—reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks,” *Nucl. Acids Res.*, vol. 44, no. D1, pp. D523–D526, 2015
 - [19] D. Vitkup, et al., “Influence of metabolic network structure and function on enzyme evolution,” *Genome Biol.*, vol. 7, no. 5, 2006, Art. no. R39
 - [20] E. R. Shellman, et al., “Metabolic network motifs can provide novel insights into evolution: The evolutionary origin of Eukaryotic organelles as a case study,” *Comput. Biol. Chem.*, vol. 53PB, pp. 242–250, 2014.
 - [21] B. J. Bornstein, et al., “LibSBML: An API library for SBML,” *Bioinf.*, vol. 24, no. 6, pp. 880–881, 2008.
 - [22] S. A. Coskun, et al., “PathCase-SB: Integrating data sources and providing tools for systems biology research,” *BMC Syst. Biol.*, vol. 6, no. 1, 2012, Art. no. 67.
 - [23] A. Cakmak, et al., “PathCase-SB architecture and database design,” *BMC Syst. Biol.*, vol. 5, no. 1, 2011, Art. no. 188.
 - [24] A. E. Cicek, et al., “An online system for metabolic network analysis,” *Database bau091*, vol. 2014, 2014. [Online]. Available: <https://doi.org/10.1093/database/bau091>
 - [25] S. A. Coskun, et al., “An online model composition tool for system biology models,” *BMC Syst. Biol.*, vol. 7, no. 1, Art. no. 88.
 - [26] M. Franz, et al., “Cytoscape.js: A graph theory library for visualisation and analysis,” *Bioinf.*, vol. 15, no. 32(2), pp. 309–311, 2016.
 - [27] E. Dogrusoz, et al., “A layout algorithm for undirected compound graphs,” *Inf. Sci.*, vol. 179, pp. 980–994, 2009.
 - [28] S. A. Becker, et al., “Quantitative prediction of cellular metabolism with constraint-based models: the COBRA toolbox,” *Nature Protocols*, vol. 2, pp. 727–738, 2007.
 - [29] L. Gravano, et al., “Approximate string joins in a database (almost) for free,” in *Proc. Int. Conf. Very Large Databases*, Sep., 11–14, 2001, pp. 491–500.
- Furkan Ozden** received the BS degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2019. He is currently working toward the MS degree and is a research assistant at CicekLab, in the Computer Engineering Department of Bilkent University.
- Metin Can Siper** received the BS and MS degrees in computer engineering from Bilkent University, Ankara, Turkey in 2015 and 2017, respectively. He is currently working as a research software engineer with Oregon Health and Science University, Portland, OR.
- Necmi Acarsoy** received the BS degree in computer engineering from the Izmir Institute of Technology (IZTECH), Izmir, Turkey, in 2017. He is currently working toward the MS degree in the same field at IZTECH.
- Tugrulcan Elmas** received the BS degree in computer engineering from Bilkent University, Ankara, Turkey, in 2017. He is currently working toward the PhD degree in computer science with EPFL, Laussane, Switzerland.
- Bryan Marty** received the BS degree in computer engineering from Case Western Reserve University, Cleveland, OH, in 2013. He is currently working as a technical lead with Yelp, San Francisco, CA.
- Xinjian Qi** received the PhD degree in computer science from Case Western Reserve University, Cleveland, OH, in 2013. His research focuses on computational analysis, visualization and entity identification of metabolic networks.
- A. Ercument Cicek** received the BS and MS degrees in computer science and engineering from Sabanci University, in 2007 and 2009, respectively. He received the PhD degree in computer science from Case Western Reserve University, in 2013. Then, he worked as a Lane fellow in computational biology at Carnegie Mellon University until 2015. Since then, he has been an assistant professor with the Computer Engineering Department of Bilkent University and is an adjunct faculty member at the Computational Biology Department of Carnegie Mellon University.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.