BioDataFuse: Enhancing Data Interoperability through Modular Queries and Knowledge Graph Construction

Tooba Abbassi-Daloii[1,*,†], Yojana Gadiya[2,3,4,*,†], Ammar Ammar[1], Egon Willighagen[1], Ana Claudia Sima[5] and Hasan Balci[6]

[1]Dept of Bioinformatics - BiGCaT, NUTRIM, FHML, Maastricht University, Maastricht, The Netherlands
[2]Fraunhofer Institute for Translational Medicine and Pharmacology (ITMP), Hamburg, Germany
[3]Fraunhofer Cluster of Excellence for Immune-Mediated Diseases (CIMD), Frankfurt, Germany
[4]Bonn-Aachen International Center for Information Technology (B-IT), University of Bonn, Bonn, Germany
[5]SIB Swiss Institute of Bioinformatics, Lausanne, Switzerland
[6]Maastricht Centre for Systems Biology (MaCSBio), Maastricht University, Maastricht, The Netherlands

Abstract

In biological research, integrating experimental data with publicly available resources is pivotal for understanding complex biological mechanisms. However, this process is often intricate and time-consuming due to the complexity and diversity of data. Furthermore, the lack of consistent harmonization across different data types complicates the management of disparate data formats and sources. Addressing this, we introduce BioDataFuse, a query-based Python tool for seamless integration of biomedical data resources. BioDataFuse establishes a modular framework for efficient data wrangling, enabling context-specific knowledge graph creation and supporting graph-based analyses. With a user-friendly interface, it enables users to dynamically create knowledge graphs from their input experimental data. Supported by a robust Python package, pyBiodatafuse, this tool excels in data harmonization, aggregating diverse sources through modular queries. Moreover, BioDataFuse provides plugin capabilities for Cytoscape and Neo4j, allowing local graph hosting. Ongoing refinements enhance the graph utility through tasks like link prediction, making BioDataFuse a versatile solution for efficient and effective biological data integration.

Keywords

Biomedical Data Source, Context-specific Knowledge Graph, Data Wrangling, Graph Analysis

1. Introduction

BioDataFuse is an innovative solution facilitating the seamless integration of diverse data resources on the fly. It establishes a modular framework, adhering to the "what-you-see-is-what-you-get (WYSIWYG)" principle, granting users control over the graph creation. The adaptable

SWAT4HCLS 2024: Bridging Life Sciences and Technology, February 26-29, Leiden, The Netherlands
*Corresponding authors.
†These authors contributed equally.
t.abbassidaloii@maastrichtuniversity.nl (T. Abbassi-Daloii); yojana.gadiya@itmp.fraunhofer.de (Y. Gadiya)
0000-0002-4904-3269 (T. Abbassi-Daloii); 0000-0002-7683-0452 (Y. Gadiya); 0000-0002-8399-8990 (A. Ammar); 0000-0001-7542-0286 (E. Willighagen); 0000-0003-3213-4495 (A. C. Sima); 0000-0001-8319-7758 (H. Balci)

backend and user-friendly interface offer a solution to address data integration challenges. The backend, implemented through the Python package *pyBiodatafuse* (GitHub repo), supports various tool functionalities. Concurrently, the front-end interface improves accessibility for non-programmers. The BioDataFuse aims to develop into a comprehensive toolkit that enables users to seamlessly explore, interpret, and visualize biomedical data across a diverse range of resources, fostering interoperability and facilitating effortless navigation for extraction of meaningful insights from context-specific biomedical graphs.

2. BioDataFuse framework

The BioDataFuse framework is structured around five primary components:

1. Data harmonizer ensures unique persistent identifiers for diverse biomedical modalities using BridgeDb framework [1]. This component supports various input data, including genes, metabolites, and tables from differential expression analysis.

2. Data annotators empower modular queries across diverse resources through **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage (SPARQL). Annotators, including Wikidata [2], Bgee [3], Molecules on Membranes Database (MolMeDB) [4], Open Targets [5], DisGeNET [6], WikiPathways [7], and STRING [8], enrich gene-related metadata, gene-disease relationships, expression profiles, pathway information, and protein-protein interactions.

3. Graph generator creates knowledge graphs from annotated data using NetworkX [9], offering a clear and structured representation. The Python module in pyBiodatafuse constructs graphs exportable to Cytoscape [10] and Neo4j [11].

4. Graph analyzer employs Python packages like Matplotlib [12], Seaborn [13], and Plotly for basic plots and network-specific summaries, facilitating a quick understanding of the data and its interconnections.

5. User interface is developed using Streamlit, enabling users to input gene lists, select identifier types, and annotate data from chosen sources.

3. Future work

Planned future directions for BioDataFuse include continued attention to the graph analyzer component, emphasizing the importance of refining and expanding its capabilities. Additionally, the future work includes supporting additional input data types, integrating annotators from drug databases and larger repositories, and ensuring the continuous updating of pyBiodatafuse at PyPi. Exploration of migrating the user interface to frameworks like Shiny or Dash is underway, aiming to improve functionality and provide an enhanced user experience. These efforts align with our dedication to advancing BioDataFuse for improved data interoperability.

4. Acknowledgments

We acknowledge the support from ELIXIR BioHackthon 2023, resulting in a preprint [14].

References

[1] M. P. van Iersel, A. R. Pico, T. Kelder, et al., The BridgeDb framework: standardized access to gene, protein and metabolite identifier mapping services, BMC Bioinformatics 11 (2010) 5. doi:10.1186/1471-2105-11-5.

[2] A. Waagmeester, G. Stupp, S. Burgstaller-Muehlbacher, B. M. Good, M. Griffith, O. Griffith, K. Hanspers, H. Hermjakob, T. Hudson, K. Hybiske, S. M. Keating, M. Manske, M. Mayers, D. Mietchen, E. Mitraka, A. R. Pico, T. E. Putman, A. Riutta, N. Q. Rosinach, L. Schriml, T. Shafee, D. Slenter, R. Stephan, K. Thornton, G. Tsueng, R. Tu, S. Ul-Hasan, E. Willighagen, C. Wu, A. I. Su, Wikidata as a knowledge graph for the life sciences, eLife 9 (2020).

[3] F. B. Bastian, J. Roux, A. Niknejad, A. Comte, S. S. Fonseca Costa, T. M. de Farias, S. Moretti, G. Parmentier, V. R. de Laval, M. Rosikiewicz, J. Wollbrett, A. Echchiki, A. Escoriza, W. H. Gharib, M. Gonzales-Porta, Y. Jarosz, B. Laurenczy, P. Moret, E. Person, P. Roelli, K. Sanjeev, M. Seppey, M. Robinson-Rechavi, The bgee suite: integrated curated expression atlas and comparative transcriptomics in animals, Nucleic Acids Research 49 (2020) D831–D847. doi:10.1093/nar/gkaa793.

[4] J. Juračka, M. Šrejber, M. Melíková, V. Bazgier, K. Berka, Molmedb: Molecules on Membranes Database, Database 2019 (2019).

[5] D. Ochoa, A. Hercules, M. Carmona, D. Suveges, J. Baker, C. Malangone, I. Lopez, A. Miranda, C. Cruz-Castillo, L. Fumis, M. Bernal-Llinares, K. Tsukanov, H. Cornu, K. Tsirigos, O. Razuvayevskaya, A. Buniello, J. Schwartzentruber, M. Karim, B. Ariano, R. Martinez Osorio, J. Ferrer, X. Ge, S. Machlitt-Northen, A. Gonzalez-Uriarte, S. Saha, S. Tirunagari, C. Mehta, J. Roldán-Romero, S. Horswell, S. Young, M. Ghoussaini, D. Hulcoop, I. Dunham, E. McDonagh, The next-generation Open Targets Platform: reimagined, redesigned, rebuilt, Nucleic Acids Research 51 (2022) D1353–D1359. doi:10.1093/nar/gkac1046. arXiv:https://academic.oup.com/nar/article-pdf/51/D1/D1353/48441188/gkac1046.pdf.

[6] N. Queralt-Rosinach, J. Piñero, A. Bravo, F. Sanz, L. I. Furlong, Disgenet-rdf: Harnessing the innovative power of the semantic web to explore the genetic basis of diseases, Bioinformatics (2016). doi:10.1093/bioinformatics/btw214.

[7] A. Agrawal, H. Balcı, K. Hanspers, S. L. Coort, M. Martens, D. N. Slenter, F. Ehrhart, D. Digles, A. Waagmeester, I. Wassink, T. Abbassi-Daloii, E. N. Lopes, A. Iyer, J. M. Acosta, L. G. Willighagen, K. Nishida, A. Riutta, H. Basaric, C. T. Evelo, E. L. Willighagen, M. Kutmon, A. R. Pico, WikiPathways 2024: next generation pathway database, Nucleic Acids Res. (2023).

[8] D. Szklarczyk, R. Kirsch, M. Koutrouli, K. Nastou, F. Mehryary, R. Hachilif, A. L. Gable, T. Fang, N. T. Doncheva, S. Pyysalo, P. Bork, L. J. Jensen, C. von Mering, The STRING database in 2023: protein-protein association networks and functional enrichment analyses for any sequenced genome of interest, Nucleic Acids Res. 51 (2023) D638–D646.

[9] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using networkx, in: G. Varoquaux, T. Vaught, J. Millman (Eds.), Proceedings of the 7th Python in Science Conference, Pasadena, CA USA, 2008, pp. 11 – 15.

[10] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, T. Ideker, Cytoscape: A software environment for integrated mod-

els of biomolecular interaction networks, Genome Research 13 (2003) 2498–2504. doi:10.1101/gr.1239303.
[11] Neo4j, Neo4j - the world's leading graph database, 2012. URL: http://neo4j.org/.
[12] J. D. Hunter, Matplotlib: A 2d graphics environment, Computing in Science & Engineering 9 (2007) 90–95. doi:10.1109/MCSE.2007.55.
[13] M. L. Waskom, seaborn: statistical data visualization, Journal of Open Source Software 6 (2021) 3021. doi:10.21105/joss.03021.
[14] Y. Gadiya, A. Ammar, E. Willighagen, D. Martinat, A. C. Sima, H. Balci, T. Abbassi-Daloii, Biohackeu23 report: Extending interoperability of experimental data using modular queries across biomedical resources, 2023. doi:10.37044/osf.io/mhsqp.