Supporting Data Foragers in Scientific Computing Community Ecosystems for Life Sciences

Hasan M. Jamil

Department of Computer Science, University of Idaho, USA
jamil@uidaho.edu

Abstract. Biology today is heavily data-driven and knowledge-centric that are stored across the linked open web in numerous heterogeneous deep web databases. To improve searching, finding, accessing, and interoperating among these diverse information sources to increase usability, the FAIR data principle has been proposed. Unfortunately, FAIR compliance is extremely low and linked open data does not guarantee FAIRness, leaving biologists on a solo hunt for information on the open network. In this paper, we propose *SoDa*, for intelligent data foraging on the internet. SoDa helps biologists discover resources based on analysis requirements, generate resource access plans, and store cleaned data and knowledge for community use. A secondary search index is also supported for community members to find archived information conveniently.

Keywords: Large language model, intelligent user interface, FAIR, wrapper generation, interoperability, ecosystem.

1 Introduction

The techniques needed to gather information from deep web databases are significantly different from browsing web pages on the internet. The standard tools used to access deep web data are wrappers and more recently, RESTful APIs. Word-of-mouth [12] and web crawlers [9] also play significant roles in the generation of resource indices such as MBDL [2] or Pathguide [1]. While the compiled resource indices help with the discovery and identification of interesting data sources and analysis tools users need, the process is completely manual, sluggish, and limited in scope. Updates are slow and do not serve unique needs users may have. In particular, they do not actively find resources of interest that have not already been found and indexed in the resource list.

Linked Open Data (LOD), on the other hand, has emerged as a promising approach to provide interconnected, machine-readable datasets, but its effectiveness is often hampered by issues related to data quality and accessibility. While LOD emphasizes openness, the FAIR (Findable, Accessible, Interoperable, and Reusable) principles [13] extend this concept, revealing significant gaps in compliance among LOD resources [11]. However, widespread adoption of FAIR principles remains a challenge, hindering the full realization of LOD's potential in supporting scientific inquiry and data-driven research.

In this paper we introduce the concept of data foraging and sharing within a community resource ecosystem that will undergo continuous curation and evolve over time [5]. We present a bird's eye view of the architecture and functionalities of the proposed system called *SoDa* (which stands for Solo Data Forager). In SoDa, we integrate four basic subsystems – a resource recommender system, a data access protocol design or wrapper system, a query processing system with the help of a schema matcher to support interoperability, and a curation system for knowledge evolution.

2 SoDa Architecture

As a preamble to the discussion of the architecture of SoDa, we introduce a potential scientific expedition a biologist might want to do using our cloud-based analysis platform for open science over linked open data.

2.1 Scientific Inquiry

In their quest to establish a definitive link between defects in sperm and male infertility [3], a biologist might want to follow the steps below once semen samples from both fertile and infertile men are collected and sequenced (among several other steps before and after).

1. Perform differential expression analysis to identify RNAs that are significantly differentially expressed between fertile and infertile men using commonly used tools such as DESeq2, edgeR, or Limma.
2. Use tools like DAVID, Enrichr, or clusterProfiler to perform Gene Ontology (GO) and pathway enrichment analysis on the differentially expressed RNAs.
3. Identify biological processes, molecular functions, and pathways that are significantly associated with the differentially expressed RNAs.

There are multiple different levels at which this project can be approached depending on the biologist or their expertise. For example, a more established researcher in fertility research may already have collected sperm samples from both fertile and infertile men, isolated RNA from these sperm samples, identified and quantified the RNA molecules present in the sperm samples using RNA sequencing and learned about the RNA profiles of fertile and infertile men before entering differential expression analysis.

An alternative to come to this stage is to use publicly available data at GEO [4] or SRA [8] databases, and start right away. However, the querying abstraction levels can be highly varied. In a cognate research in ProAb [7], we have explored the possibility of asking this query at the highest possible abstraction level in natural English, likely as follows:

> Is there a link between human spermatozoal RNA and infertility? Could I establish the link computationally?

and developing a candidate workflow that could be executed fully automatically. However, as demonstrated in BioNursery [6], significant technological and knowledge gaps exist to fully achieve such an approach, and often, substantial human involvements are necessary. Abstractions supported in SoDa help find relevant data and tools automatically but require users' involvement in selecting, isolating and generating a computational information extraction procedure that can be used in automation. Once collected and archived, and users have a well articulated computational process in mind, they are able to develop a computational pipeline to implement a scientific inquiry using the smart SoDa GUI.

2.2 Components of SoDa

Fig 1 shows the conceptual model of SoDa and its four basic subsystems. We briefly discuss these components of SoDa in the sections below.

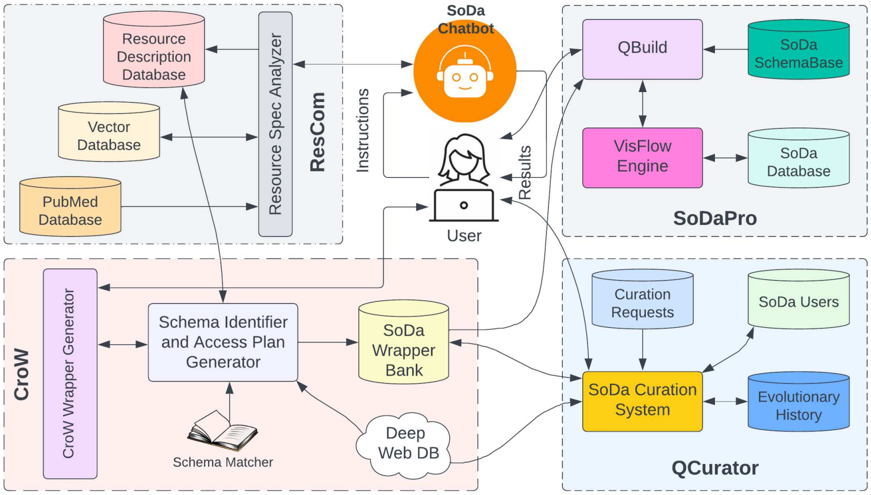

Fig. 1. SoDa Architecture.

Resource Recommendation using ResCom Existing resources in SoDa can be browsed from a searchable index, or a specification in the form of a paragraph for the resource needs can be used as a search key in natural English. For example,

> Need to find normalized sperm RNA-seq expression data for differential expression analysis (using DESeq2, edgeR, or Limma).

The Resource ReCommender system ResCom accepts this request and suggests a ranked list of databases prioritizing the most relevant at the top. Additionally, it generates a possible scheme of the table that can be accessed at an internet location, i.e., a URL.

In response to the above request, for example, ResCom will first search for a matching data set in the SoDa archive. In SoDa, all table schemes are semantically described in some detail using natural language, and their possible usage is also included in these descriptions. These descriptions are stored in SoDa's vector database for a possible linguistic analysis using a LLM to ascertain query relevance. When an external search for a table is warranted, ResCom uses PubMed abstracts as the first level of descriptors of data to find a match. If a sufficient match is found, the list of abstracts is organized in a decreasing order ranked list and also vectorized for semantic matching. HTTP links found in the abstracts are explored exhaustively using link hoping to identify a database with the closest match and presented to the user for review with two options – accept or search next. The search ultimately ends in a success or failure.

Accessing Resources and Information Extraction using CroW Once a resource external to SoDa is identified, a wrapper needs to be generated to facilitate real time online access and ensure successful extraction. In SoDa, resources are of two types – a deep web database[1], or an online analysis tool that returns a table on appropriate submission of input parameters, again likely using forms. In SoDa, the underlying data model is relational, and thus all its operations are conceived using a tabular representation of data, although the engine is capable of processing, TXT, CSV, XML, and JSON formatted data.

However, developing an access protocol for such resources is mostly a manual process, largely because each one is unique. We developed a GUI enabled public wrapper generation system called *CroW* (which stands for Crowd Wrapper Generator) for this purpose. CroW supports visual tools and functions to help users assist CroW in learning the input form behavior and data layouts so that a formula can be learned using which CroW will be able to recreate the access plan for a site u when requested by a query in real time. Once a wrapper is generated and tested, it can be archived in a searchable wrapper bank for community use.

Application Design using SoDaPro Application design in SoDa always involves registered resources, and never resources for which it has no access plans generated in advance. SoDa supports a graphical query builder and allows fairly complex query generation, and computed view materialization, either temporarily or indefinitely. The application designer is called *QBuild* (stands for Query Builder). Except for the directly stored tables (extensional tables), all references to a table are virtual, which means, a reference to a table on the internet is made through the use of a wrapper available in SoDa wrapper database.

QBuild uses BioFlow language construct `extract` [10] to access online virtual tables in real time and approached the deep web resource querying problem

[1] A database that can only be accessed via query form submission or through an API.

declaratively. In this approach, the deep web database at a URL φ is treated as a black-box, to which an input relation r_i is sent and in return, an output relation r_o is expected over the schemes S_i and S_o, respectively. SoDa supports two types of queries – extract statement for deep web data extraction, and select statement for querying tables, and workflow orchestration using a list of these two query types. A GUI for query construction using Extract and Select queries and workflow construction is supported for the so called naive users.

Crowd Enabled Curation of Workflows when Pipelines Break SoDa's crowd curation system, called *QCurator*, takes an active support approach for error tracing and bug fixing using a crowd computing approach. SoDa users have the option to push a code segment or query to a community discussion board for help. The discussion board has a special notification system which alerts relevant users of an available help request as a ticket. Users may sign up as a participant crowd in the research area or topic of choice. They are able to participate in the discussion to fix the errors, execute the code fragments to see if it worked, or opt out. Until they explicitly exit the ticket, it stays active in their notification queue. The ticket is resolved until the user who initiated the ticket exits it, or all active participants do so. Note that the ticket is also available for all members of the community in the general discussion board. However, the general discussion board notification goes off only when the initiating user exits the ticket. Finally, the discussion with and solution by the special users are also visible in the general discussion board.

The look and feel of the QCurator discussion board is not much different than the Stack Overflow or GitHub discussion boards, but the difference is more critical when it comes to the notification and debugging approaches. Notifications are owned by the users to whom these are specifically sent to, and the notification to the general discussion board is owned by the ticket initiating user. Therefore, when all of the owners exit the ticket, it stays active in each of the owners' dashboards. Finally, every user in the community has access to the discussion board and is able to debug, modify and execute the code fragments on the forum directly from their dashboards.

3 Conclusion

SoDa primarily aims to support biologists to gather data from online resources in a searchable repository so that other biologists can also use the data readily. The search for resources is truly flexible using text analysis of resource needs. As opposed to fully automatic ProAb [7], SoDa is manual with a human-in-the-loop principle, and offers relatively lower failure possibilities. The downside is that users must know exactly what they want to compute, but not necessarily how to do it, and thus SoDa supports a declarative way of computing internet workflow queries involving heterogeneous resources. The current edition of SoDa is experimental and does not support user data archival for guest users. Future editions of SoDa will support user accounts and allow data archival options along with advanced query building options, likely using LLMs.

Acknowledgement

Research was supported in part by a National Institutes of Health Institutional Development Award (IDeA) #P20GM103408, a National Science Foundation CSSI grant OAC 2410668, and a US Department of Energy grant DE-0011014.

References

1. G. D. Bader, M. P. Cary, and C. Sander. Pathguide: a pathway resource list. *Nucleic Acids Res.*, 34(Database-Issue):504–506, 2006.
2. C. Burks. Molecular biology database list. *Nucleic acids research*, 27 1:1–9, 1999.
3. R. B. Burl, S. Clough, E. Sendler, M. Estill, and S. A. Krawetz. Sperm rna elements as markers of health. *Systems Biology in Reproductive Medicine*, 64(1):25–38, 2018.
4. E. Clough, T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, H. Lee, N. Zhang, N. Serova, L. Wagner, V. Zalunin, A. Kochergin, and A. Soboleva. NCBI GEO: archive for gene expression and epigenomics data sets: 23-year update. *Nucleic Acids Research*, 52(D1):D138–D144, Nov. 2023.
5. D. Gendarmi, F. Abbattista, and F. Lanubile. Fostering knowledge evolution through community-based participation. In *(CKC 2007)@(WWW2007) Banff, Canada, May 8, 2007*, volume 273. CEUR-WS.org, 2007.
6. H. Jamil, S. A. Krawetz, and A. Gow. Knowledge synthesis using large language models for a computational biology workflow ecosystem. In *SAC 2024, Avila, Spain, April 8-12, 2024*, pages 523–530. ACM, 2024.
7. H. M. Jamil. Smart science needs linked open data with a dash of large language models and extended relations. In *aiDM@SIGMOD 2024, Santiago, Chile, 14 June 2024*, pages 1:1–1:11. ACM, 2024.
8. K. Katz, O. Shutov, R. Lapoint, M. Kimelman, J. R. Brister, and C. O'Sullivan. The Sequence Read Archive: a decade more of explosive growth. *Nucleic Acids Research*, 50(D1):D387–D390, 11 2021.
9. Y. Li, Y. Wang, and E. Tian. A new architecture of an intelligent agent-based crawler for domain-specific deep web databases. In *WI 2012, Macau, China, December 4-7, 2012*, pages 656–663. IEEE Computer Society, 2012.
10. X. Mou and H. M. Jamil. Visual life sciences workflow design using distributed and heterogeneous resources. *IEEE/ACM TCBB*, 17(4):1459–1473, 2020.
11. F. Pattyn, B. Wulbrecht, K. Knecht, and H. Constandt. Assessment of fairness of open data sources in life sciences. In *(SWAT4LS 2017), Rome, Italy, December 4-7, 2017*, volume 2042. CEUR-WS.org, 2017.
12. T. Rodrigues, F. Benevenuto, M. Cha, P. K. Gummadi, and V. A. F. Almeida. On word-of-mouth based discovery of the web. In *ACM SIGCOMM IMC '11, Berlin, Germany, November 2-, 2011*, pages 381–396. ACM, 2011.
13. M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.