Creating and Exploiting the Intrinsically Disordered Protein Knowledge Graph (IDP-KG)

Alasdair J. G. Gray[1][0000−0002−5711−4872], Petros Papadopoulos[1][0000−0002−8110−7576], Imran Asif[1][0000−0002−1144−6265], Ivan Mičetić[2][0000−0003−1691−8425], and András Hatos[2][0000−0001−9224−9820]

[1] Department of Computer Science, Heriot-Watt University, Edinburgh, UK
[2] Department of Biomedical Sciences, University of Padua, Padova, Italy

Abstract. There are many data sources containing overlapping information about Intrinsically Disordered Proteins (IDP). IDPcentral aims to be a registry to aid the discovery of data about proteins known to be intrinsically disordered by aggregating the content from these sources. Traditional ETL approaches for populating IDPcentral require the API and data model of each source to be wrapped and then transformed into a common model.

In this paper, we investigate using Bioschemas markup as a mechanism to populate the IDPcentral registry by constructing the Intrinsically Disordered Protein Knowledge Graph (IDP-KG). Bioschemas markup is a machine-readable, lightweight representation of the content of each page in the site that is embedded in the HTML. For any site it is accessible through a HTTP request. We harvest the Bioschemas markup in three IDP sources and show the resulting IDP-KG has the same breadth of proteins available as the original sources, and can be used to gain deeper insight into their content by querying them as a single, consolidated knowledge graph.

Keywords: Knowledge Graphs · Schema.org · Bioschemas · Findable · Intrinsically Disordered Proteins

1 Introduction

One of the goals of the ELIXIR Intrinsically Disordered Protein (IDP) community is to create a centralised registry for IDP data to support the community in their data analyses. The registry will aggregate data contained in the community's numerous specialist data sources, such as DisProt [8], MobiDB [11], and Protein Ensemble Database (PED) [9], that contain overlapping but complimentary data about IDPs. Users of the registry should be able to search for IDPs and be presented with summary details of the protein and how it is known to be disordered; with the specialist source consulted for more detailed data.

Bioschemas is a community effort to provide machine-readable markup within life sciences resources to increase their discoverability [6]. The community have

developed extensions to the core Schema.org vocabulary [7] to enable the representation of life science concepts such as proteins. Deployments of this markup have been made in several life sciences resources, including DisProt, MobiDB, and PED. Bioschemas also provide usage profiles that recommend which properties should be present in the markup to represent a specific resource. The purpose of these profiles is to simplify the consumption and use of the markup.

In this paper, we demonstrate that Bioschemas markup can be harvested to create a central repository of IDPs by creating the Intrinsically Disordered Proteins Knowledge Graph (IDP-KG[3]). It is not sufficient to simply harvest all the markup into a single data repository. The concepts within the markup need to be identified and reconciled since the sources contain overlapping, and potentially conflicting, information about proteins but use different identifiers for the proteins. Therefore, the provenance of each statement should be tracked so that users of the registry can retrieve full details from the original data source.

2 Background

We will now discuss the background material for our work. Note that throughout this paper we will use CURIEs to link to items in databases. These can be resolved using Identifiers.org. Similarly, ontology terms will be given as CURIEs that correspond to the widely used prefixes given in https://prefix.cc.

2.1 Schema.org and Bioschemas

Schema.org provides a way to add semantic markup to web pages to enable those web pages to become more understandable by the search engines that index them, and therefore to improve search results [7]. Markup is increasingly being applied to web pages as it boosts a site's ranking in search results. The markup in web pages also enhances the search experience for end users, e.g. enabling them to make more informed decisions when deciding between two search results, or by providing dedicated search portals such as Google Dataset Search [2] or ELIXIR's training portal TeSS [1].

The Schema.org vocabulary provides *types* which correspond to the things we can describe, and *properties* which capture the characteristics of those things. The majority of the vocabulary is focused on generic web search, e.g. books, movies, or places, but it also includes types relevant to science, e.g. `Dataset`, and most recently types have been added for Bioinformatics[4] such as `Gene`, `Protein`, and `Taxon`. A major benefit of this approach is that the markup is accessible to all through a common API, i.e. HTTP Get requests, there is no need to learn and code for the REST API of each individual source.

The Bioschemas community (Bioschemas.org) promotes the use of Schema.org markup within life sciences web resources to improve their Findability, and provide lightweight Interoperability (c.f. the FAIR Data Principles) [6]. The community achieve this by:

[3] `https://alasdairgray.github.io/IDP-KG/` accessed 30 Sept 2021
[4] Schema.org v13.0 `https://schema.org/version/13.0` accessed 30 Aug 2021.

1. Proposing extensions to the Schema.org vocabulary to include types and properties relevant for life sciences resources; and
2. Providing recommended usage profiles over Schema.org types.

Seven types covering key life sciences areas have been included in the Schema.org pending vocabulary. The Bioschemas community continue to work to add more types, e.g. the annotation of genes or proteins using a `SequenceAnnotation` type. The goal is not to replace existing life sciences ontologies, but to provide a lightweight vocabulary to aid discovery of resources. Once discovered it is expected that detailed biological models, captured with rich Interoperable ontologies, will be used to accurately describe the data.

For any given type in Schema.org, there can be a large number of properties available to use, many of which can be inherited from parent types. For example, the `Dataset` type has over 100 properties due to the inheritance from `CreativeWork` and `Thing`. This can make it difficult for developers of markup to know which properties to use, and certainly they are unlikely to use all. Bioschemas profiles[5] provide usage guidelines for Schema.org types; identifying the most critical properties to aid search, and important properties for disambiguation; presented as *minimal* and *recommended* properties respectively. This provides a much smaller pool of properties for types relating to the life sciences.

2.2 Intrinsically Disordered Protein Data Sources

The ELIXIR IDP community[6] curates and maintains many data resources that function as the basis of the IDPcentral registry. These specialist data sources are built around a subset of proteins having the interesting property of being unstructured or structurally disordered. The structural and functional aspects of such proteins are covered in three distinct resources.

DisProt [8] is a manually curated database of IDPs where structural disorder and functional annotation is recorded directly from evidence in scientific publications. For each protein, Bioschemas markup is exposed describing all disordered regions and their functions. These are represented as a `SequenceAnnotation` that identify a region of the protein sequence using a `SequenceRange` and associating it with a defined term from the IDPOntology [8].

MobiDB [11] is a comprehensive database with experimental and predicted protein disorder for all known protein sequences. Although all MobiDB entries are marked up with Bioschemas, only the most interesting subset of entries appears in the sitemap index. This subset contains ˜2k entries out of 189M entries in the complete MobiDB. A set of `SequenceAnnotation` types is exposed for each `Protein` identifying all consensus predicted disordered regions, with the range of the region captured as a `SequenceRange`.

The Protein Ensemble Database (PED) [9] is a primary database for the deposition of protein structural assemblies which include intrinsically disordered

[5] Bioschemas profiles `https://bioschemas.org/profiles/` accessed 30 Aug 2021.

[6] `https://elixir-europe.org/communities/intrinsically-disordered-proteins` accessed Sept 2021

proteins. A database entry in PED consists of an ensemble of proteins, in contrast to the other two resources where an entry describes a single protein. At the protein level, the description is comparable to DisProt and MobiDB with individual proteins annotated with a series of `SequenceAnnotation` types having defined terms describing the detection method used to obtain structural information connected to a specific `SequenceRange` region.

3 Knowledge Graph Generation

The creation of the IDP-KG requires two steps. First we must harvest the markup from each of the data sources. Second we need to transform the source markup into the model for the knowledge graph, reconciling the multiple identifiers for a specific protein into a single concept.

3.1 Data Harvesting

The markup was extracted from the three data sources using the Bioschemas Markup Scraper and Extractor (BMUSE). Markup is extracted using HTTP Get requests which means that resource specific APIs do not need to be coded for. To verify the correctness of the harvesting, we developed three datasets.

BMUSE. The Bioschemas Markup Scraper and Extractor (BMUSE[7]) is a data harvester developed specifically to extract markup embedded within web pages. BMUSE has been developed to extract markup embedded as either JSON-LD or RDFa, and also supports the use of both in the same page. The pages to be harvested can be static, or be single page applications (dynamic) that require JavaScript processing on the client side to generate the page content. BMUSE harvests data from a given list of URLs or sitemaps; it does not perform web crawling by following links embedded within pages. A maximum number of pages to harvest per sitemap is also required.

For each page extracted, BMUSE generates an n-quad file containing:

1. The extracted markup stored in an RDF named graph with the IRI of the named graph being uniquely constructed based on the date of the scrape and the page visited. Where the markup does not contain a subject IRI for the data, i.e. the JSON-LD markup does not include an `@id` attribute, BMUSE substitutes in the page URL to avoid the use of blank nodes.
2. Provenance data about the data harvesting. This is stored in the default graph and describes the named graph in which the data is stored. The provenance data provided is:
   - URL of the page visited using `pav:retrievedFrom`
   - Date of extraction using `pav:retrievedOn`
   - The version of BMUSE used to harvest the data using `pav:createdWith`

[7] `https://github.com/HW-SWeL/BMUSE` accessed Sept 2021

The `pav:retrievedFrom` property can be used to provide the links back from individual pieces of data to the source from which it came. The other two properties are primarily used for debugging purposes, although the retrieval date can also be used to ensure that the most up to date data is available in the generated IDP-KG.

Harvested Data. To develop and test the data processing pipeline to be applied to the harvested data, we used a series of test datasets. These correspond to data harvested from the three data sources on 28 September 2021. We note that in the initial run of BMUSE 13 pages produced errors due to timeouts. These pages were harvested in a second run with just those pages listed as targets.

Test-8: This dataset consists of eight sample pages that correspond to those used in [5]. In constructing this test dataset, we ensured that there was at least one protein (`uniprot:P03265`) that was present in all three datasets. Two additional pages have been added since the previous work which correspond to the DisProt homepage and another page that exists in the DisProt sitemap but contains no markup. These were added to ensure that the pipeline would work with pages not corresponding to protein information.

Sample-25: This dataset contains the first 25 pages harvested from each of the sitemaps of the source databases. This corresponds to 5 to 9 pages of site structure and then first 25 protein pages per source[8]. This dataset allowed us to check the pipeline would scale up.

Full: This dataset contains all pages that could be harvested from the sitemaps of the three data sources. This contains the 5 to 9 pages of site structure per data source and all protein pages listed in the sitemap. This dataset is used to construct the IDP-KG.

3.2 Data Transformation

After the data has been harvested, it is processed so that information about a particular protein, which can come from multiple sources, is consolidated into a single concept for the protein, with links back to where each piece of data originated. The data transformation process is available as a Jupyter Notebook[9]. This is an extended version of the notebook presented in [5], containing bug fixes and the ability to extract markup corresponding to more Bioschemas profiles.

The notebook uses SPARQL CONSTRUCT queries to extract the data from the harvested pages and convert them into the IDP-KG model, based on the Bioschemas vocabulary. While the queries are based on the properties listed in the corresponding Bioschemas profile, they make extensive use of `OPTIONAL` clauses since the data does not always exactly correspond to the profile.

[8] The sitemap of each source is split into two entries in the BMUSE configuration file.

[9] `https://github.com/AlasdairGray/IDP-KG/blob/main/notebooks/ETLProcess.ipynb` accessed Sept 2021

Bioschemas Profiles. Within the three data sources, we expected to find markup conforming to the following Bioschemas profiles:

- `DataCatalog` (v0.3-RELEASE)
- `Dataset` (v0.3-RELEASE)
- `Protein` (v0.11-RELEASE)
- `SequenceAnnotation` (v0.1-DRAFT)
- `SequenceRange` (v0.1-DRAFT)

Additionally, within these profiles there are uses of the Schema.org types `PropertyValue` and `DefinedTerm`, which must be processed separately, and references to pages of type `ScholarlyArticle`.

While all the data conforms to the same data vocabulary, there are differences in the underlying usage. DisProt and MobiDB provide protein centric representations of the data. PED provides a cluster of proteins on a single page. These differences need to be consolidated into a coherent knowledge graph model centred around proteins.

Instance Merging. Each of the data sources uses their own identifier scheme to identify concepts in their data. Within the IDP-KG, we need to aggregate the data from the multiple sources into a single consolidated entry, which will need its own identifier. In considering the different entity types, it was decided that only the proteins would be merged, as there is no clear way to decide when two annotations are equivalent and it is not expected that multiple instances of the `Dataset` and `DataCatalog` data would appear in the different datasets.

The IDPcentral team have decided they will use UniProt accessions [12] as a central spine for identifying proteins. This means that for each source web page about a protein, where the protein is identified by the data source's IRI, e.g. `https://disprot.org/DP00003`, the conversion process needs to align and merge this to a UniProt accession number. Fortunately each source includes a `schema:sameAs` declaration to the UniProt accession, although different UniProt namespaces were used by the different sources. Each protein was given an IRI in the IDPcentral namespace of the form

```
https://idpcentral.org/id/<accession>
```

where `<accession>` is replaced by the UniProt accession for the protein.

Knowledge Graph Construction. While constructing the IDPcentral knowledge graph, it was assumed that the data sources would contain declarations of the same property of information, e.g. the name of the protein. However, we do not assume that they are consistent in their content. There are two cases to consider. The first is that each source contains different values but these compliment each other, e.g. a list of synonyms where no source will necessarily have a complete set but by merging the data from the sources the IDPcentral knowledge graph would have a more complete set. The second case is where two sources

have differing values for a property which should have a single specific value, e.g. protein name. Rather than decide that a specific source's value should be used, we have decided to include all values available in the sources together with the provenance. Users of the data can then decide on the correct value, and feedback issues to the source with the erroneous value.

To support providing statement level provenance, we adopted the named graph approach that was used in the Open PHACTS platform [4]. This involves placing data statements in named graphs based on the page where they have been harvested from. The provenance data declared about the named graph is stored in the default graph.

4 Data Analysis

To verify the generated knowledge graph, we performed various data analyses. These build on the queries from [5] but go further in their analysis. The queries are available in a Notebook[10] and also through the IDP-KG SPARQL endpoint[11].

4.1 Knowledge Graph Statistics

We first give an overview of the IDP-KG using the statistics recommended in the HCLS Community Profile for Dataset Descriptions [3]. A summary of some of the key statistics can be found in Table 1, with the full statistics available in the notebook. The basic statistics show that the key difference between our three knowledge graphs is the number of proteins. This is shown by the number of properties and classes being constant between the three samples. This verified that we were getting consistent performance from our ETL process over the different harvested data samples, and our domain experts have verified the content of the test-8 knowledge graph. We note that there are only two instances of the `Dataset` type. This is due to an unresolved bug in BMUSE, but does not affect the retrieval of proteins.

Table 2 presents a comparison between the number of proteins found in the original data sources and the number in the IDP-KG. The comparison gives the number of proteins in the different intersections of the data sources. The table shows that the data harvesting completely recreates the information available in the data sources.

4.2 IDP Analysis Queries

Now that we have verified that the IDP-KG is complete with respect to the content of the sources, we can use it to analyse the data available about IDPs.

[10] `https://github.com/AlasdairGray/IDP-KG/blob/main/notebooks/AnalysisQueries.ipynb` accessed Sept 2021

[11] We have deployed the "Snorql - Extended Edition (`https://github.com/ammar257ammar/snorql-extended`)" query interface at `https://swel.macs.hw.ac.uk/idp` with access to the same queries that are used in the analysis notebook.

KG	Test-8	Sample-25	Full
Triples	766	7,704	278,572
Subjects	179	1,706	62,972
Properties	34	34	34
Objects	207	1,818	67,334
Classes	8	8	8
Literals	140	715	18,177
Graphs	10	81	4,287

KG	Test-8	Sample-25	Full
DataCatalog	1	2	2
Dataset	1	2	2
DefinedTerm	32	126	4,262
PropertyValue	57	652	17,607
Protein	8	69	2,701
ScholarlyArticle	7	75	2,578
SequenceAnnotation	32	350	15,767
SequenceRange	32	350	15,767

Table 1. HCLS Dataset Description statistics for IDP-KG.

Description	IDP Sources	IDP-KG
Havested Pages		4286
DisProt Pages		2039
MobiDB Pages		2075
PED Pages		172
Protein Pages		4284
DisProt entries (from sitemap)	2038	2038
MobiDB entries (from sitemap)	2074	2074
PED entries (from sitemap)	172	172
Distinct Proteins (Union)	2701	2701
DisProt Proteins	2038	2038
MobiDB Proteins	2074	2074
PED Proteins	90	90
DisProt \ (MobiDB ∪ PED)	586	586
MobiDB \ (DisProt ∪ PED)	624	624
PED \ (DisProt ∪ MobiDB)	34	34
(DisProt ∪ MobiDB)	2667	2667
(DisProt ∪ PED)	2077	2077
(MobiDB ∪ PED)	2115	2115
DisProt ∩ MobiDB	1445	1445
DisProt ∩ PED	51	51
MobiDB ∩ PED	49	49
(DisProt ∩ MobiDB) \ PED	1401	1401
(DisProt ∩ PED) \ MobiDB	7	7
(MobiDB ∩ PED) \ DisProt	5	5
DisProt ∩ MobiDB ∩ PED	44	44

Table 2. Comparison of proteins present in the IDP-KG and the original data sources.

The answers presented in this section are possible due to the aggregation of the data into a single knowledge graph. We only perform the following analysis over the full IDP-KG. The full set of responses to these queries are available through the notebook or IDP-KG SPARQL endpoint.

From Table 1 we can see that there are 15,767 annotations on the proteins. These correspond to 11,046 from DisProt, 4,488 from MobiDB, and 233 from PED (annotations per dataset query). Using the annotations in multiple datasets query, we can see that there are 912 proteins with annotations from more than one dataset, with https://idpcentral.org/id/P04637 having a total 77 annotations, contributed by all 3 datasets. Using the annotations per article query, we find that there are 2,578 distinct scholarly articles referenced in the annotations, with the article `pubmed:20657787` providing 80 annotations. Finally, using the annotations per term code query, we found that 149 codes from the Intrinsically Disordered Protein Ontology are used, with `IDPO:00076` (Disorder) being the

most common with 7,542 instances, followed by `IDPO:00063` (Protein Binding) with 1,325 instances.

5 Related Work

Schema.org markup is extensively used by search engines (Google, Microsoft, and Yandex) to optimise search results (SEO) [7]. Rather than trying to infer the topic and content of a page, the markup states explicitly what the page is about. Based on this markup, search companies have been building extensive knowledge graphs about the content of the Web, with the Google Knowledge Graph being the most widely known. As well as improving search results, these internal knowledge graphs are used to provide information boxes and rich snippets for search results. Google have developed a dedicated Dataset Search Portal based on the markup embedded within web pages about data on the web [2]. The work reported here uses the same approach of harvesting data from the Web to generate a knowledge graph, but rather than doing this at the scale of the Web, we have focused on a specific life sciences community who had a need to aggregate their disparate data sources without needing to establish an agreed set of web services. The ELIXIR TeSS training portal [1] uses Bioschemas markup embedded within web pages to populate its registry. TeSS maintains a list of sources that it gathers its data from, and as there is no overlap in the content it does not need to reconcile the concepts that it retrieves.

The work presented in this paper relies on the ability to harvest markup embedded within web pages. The common crawl [10] is a public dataset containing content retrieved from the Web. While it contains large amounts of data that can be utilised to imitate the search engines, it does not have the focus required for this work. Gleaner[12] is an open source tool that can be used for harvesting markup embedded within web sites. It has been built to exclusively extract Schema.org markup; which limits it applicability when using new types and properties that have yet to be included into the Schema.org vocabulary. It also does not track where content has been retrieved from.

6 Conclusions and Future Work

In this work, we have shown that Bioschemas markup can be harvested, transformed using a standard API (*c.f.* HTTP Get), and used to generate a community focused knowledge graph. We verified that the breadth of coverage was equivalent to the original sources, and showed that the resulting knowledge graph can be used to gain further insight into the domain. As future work, we plan to extend the number of sources from which we harvest data and to further exploit the IDP-KG to gain further insights into IDPs. We also intend to extend our transformation framework so that it can be applied in other life sciences communities with Bioschemas markup.

[12] `https://gleaner.io/` accessed September 2021

Acknowledgements. This work was funded through the ELIXIR Strategic Implementation Study Exploiting Bioschemas Markup to Support ELIXIR Communities `https://elixir-europe.org/about-us/commissioned-services/exploiting-bioschemas-markup-support-elixir-communities`. Early stages of this work were carried out during the BioHackathon Europe 2020 organized by ELIXIR in November 2020.

References

1. Beard, N., Bacall, F., Nenadic, A., et al: TeSS: a platform for discovering life-science training opportunities. Bioinformatics **36**(10), 3290–3291 (2020). https://doi.org/10.1093/bioinformatics/btaa047
2. Brickley, D., Burgess, M., Noy, N.: Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In: WWW '19. pp. 1365–1375 (2019). https://doi.org/10.1145/3308558.3313685
3. Dumontier, M., Gray, A.J., Marshall, M.S., et al: The health care and life sciences community profile for dataset descriptions. PeerJ **4** (2016)
4. Gray, A.J.G., Groth, P., Loizou, A., et al: Applying linked data approaches to pharmacology: Architectural decisions and implementation. Semantic Web **5**(2), 101–113 (2014). https://doi.org/10.3233/SW-2012-0088
5. Gray, A.J.G., Papadopoulos, P., Mičetić, I., Hatos, A.: Exploiting Bioschemas Markup to Populate IDPcentral. Tech. rep., BioHackrXiv (2021). https://doi.org/10.37044/osf.io/v3jct, type: article
6. Gray, A.J., Goble, C.A., Jimenez, R.: Bioschemas: From Potato Salad to Protein Annotation. In: ISWC (Posters, Demos & Industry Tracks) (2017)
7. Guha, R.V., Brickley, D., Macbeth, S.: Big data makes common schemas even more necessary. CACM **59**(2) (2016). https://doi.org/10.1145/2844544
8. Hatos, A., Hajdu-Soltész, B., Monzon, A.M., et al: DisProt: intrinsic protein disorder annotation in 2020. Nucleic Acids Research **48**(D1), D269–D276 (2020). https://doi.org/10.1093/nar/gkz975
9. Lazar, T., Martínez-Pérez, E., Quaglia, F., et al: PED in 2021: a major update of the protein ensemble database for intrinsically disordered proteins. Nucleic Acids Research **49**(D1), D404–D411 (2021). https://doi.org/10.1093/nar/gkaa1021
10. Patel, J.M.: Introduction to Common Crawl Datasets. In: Getting Structured Data from the Internet: Running Web Crawlers/Scrapers on a Big Data Production Scale, pp. 277–324. Apress (2020). https://doi.org/10.1007/978-1-4842-6576-5_6
11. Piovesan, D., Necci, M., Escobedo, N., et al: MobiDB: intrinsically disordered proteins in 2021. Nucleic Acids Research **49**(D1), D361–D367 (2021). https://doi.org/10.1093/nar/gkaa1058
12. The UniProt Consortium: UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Research **49**(D1), D480–D489 (2021). https://doi.org/10.1093/nar/gkaa1100