Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Databases and ontologies

PER BRENDA: Parse and Extract tool for Machine Learning applications using the BRENDA Database

Gian Marco Visani[1,&], Thomas McNulty[1,&], and Soha Hassoun[1,2,*]

[1]Department of Computer Science, Tufts University, 161 College Ave, Medford, MA, 02155, USA

[2]Department of Chemical and Biological Engineering, Tufts University, 4 Colby St, Medford, MA, 02155, USA

[&]Equal contribution.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: The BRENDA database is a rich source of enzymatic reaction information. To fully automate the extraction of this data in a format suitable for machine learning application, we propose a tool, PER (Parse and ExtRact) BRENDA. Our tool parses the BRENDA textfile into a JSON file that mimics the tree-like structure of the BRENDA database as visible through the web interface. PER BRENDA allows users to efficiently custom query the JSON file for information associated with enzymes as specified by their Enzyme Commission (EC) number. PER BRENDA significantly facilitates the access of the BRENDA data when compared to current APIs.

Availability and implementation: The Python code and instructions for use are available at https://github.com/hassounlab/PER_BRENDA

Contact: soha@cs.tufts.edu

Supplementary information: NA

1 Introduction

The BRENDA (BRaunschweig ENzyme DAtabase) database provides an extensive overview of enzymes and provides valuable data covering more than 50 categories of enzyme properties for hundreds of millions of enzymes (Jeske, et al., 2019). This data is compiled through manual and automated mining of millions of literature references. Organized based on the Enzyme Classification (EC) system (Liebecq, 1997), each enzyme entry in the BRENDA database is functionally organized into sections such as the Enzyme Nomenclature, Enzyme-Ligand Interactions, Diseases, Functional Parameters, Organism-related Information, and others. Since the 2006 BRENDA release, a SOAP-based web service API (https://brenda-enzymes.org/soap.php) provided access to different methods for extracting various data fields such as the enzyme's natural substrates, pathways, pH ranges and stability, reactions, and others. This API is implemented in multiple languages including PERL, PHP, Python and JAVA, and is heavily utilized by the bioinformatics community. This API is rich, allowing for detailed *queries*. For example, there are functions to retrieve EC numbers based on K_M values, and also to retrieve K_M values per EC. As it is possible to download the BRENDA database as a textfile under the Creative Commons Attribution License, there are available python parsers including brendapy, which extracts the protein information from the textfile (https://pypi.org/project/brendapy/). This parser was specifically developed in the context of extracting enzyme parameters, such as K_M and K_i, while building kinetic pathway models.

Current APIs, however, have several disadvantages. First, while functional, the APIs do not easily correspondence with the meticulously organized visual rendition of database. An end user who is accustomed to the BRENDA tables within each section must mentally map the visual display of the database to methods, and sometimes call several methods to retrieve the various table entries. This is particularly problematic when cross-referencing data (e.g. finding the sequence accession numbers associated with K_M values, or with reactions). Second, the APIs do not target machine learning applications where there is heavy emphasis on structured input data. Parsing the desired information from the textfile is non-trivial and may slow down research labs wishing to use the BRENDA data for their studies. Lastly, the BRENDA database does not link compound names to known chemical identities (e.g., PubChem, KEGG, or BioCyc IDs), further hindering the utilization of the data for machine learning applications that rely on molecular or sequence structures.

To address these shortcomings, we designed a novel Parse and ExtRaction tool for BRENDA. The tool, termed PER BRENDA, provides several advantages. First, PER BRENDA parses the downloaded textfile and a JSON file is returned with a structure that mimics, to the best of our abilities, the structure of the BRENDA database web interface. JSON is an open standard file format the uses human-readable text to serialize attribute-value pairs for data objects. Second, unlike the multitude of methods offered by the BRENDA API, PER BRENDA offers *one* parameterizable method for extracting the JSON file. The method is parameterized using keywords that reflect the structure of the BRENDA web interface. Importantly, the JSON file organization reflects the tables provided in the various tables presented through the BRENDA web interface. Third, an additional feature allows extracting select JSON entries into comma-separated textfile, a format easily read by machine learning and python code. Fourth, we generate a listing of all compounds involved with the query. This list can then be mapped to PubChem or KEGG IDs using other services, thus significantly enhancing the utility of the compound data available through BRENDA. PER BRENDA thus offers low overhead for retrieving the BRENDA data in a familiar format.

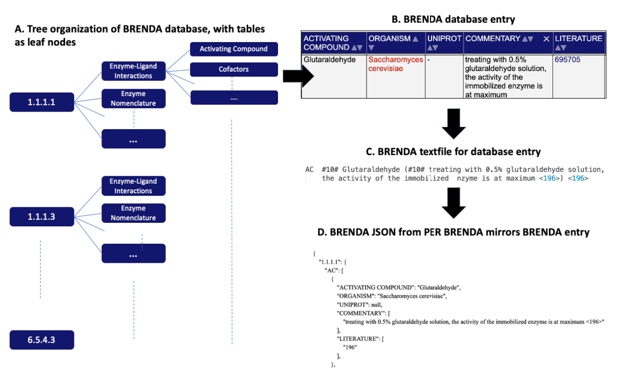
2 Methods

2.1 Parsing BRENDA Text to JSON

The JSON file is organized to reflect a multi-rooted tree-like structure (Fig. 1A), which becomes evident once carefully examining the BRENDA web interface. The root of each tree is an EC number. Then, there are categories, included the Enzyme Nomenclature, Enzyme-ligand interactions, and others. These categories are consistent across all EC entries. Each category is uniquely associated with several subcategories. For example, the Functional Parameters category comprises the following subcategories: K_M Values, Turnover Numbers, k_{cat}/K_M values, and others, while the Molecular Properties category comprises subcategories such as pH Stability, Temperature Stability, General stability, and others. Each subcategory is organized as a table. The first column in such tables contains the values of the primary data field, while other columns provide additional information. For example, the first column in the K_M Values table lists the K_M value, while the other columns list the relevant substrate, organism, the UNIPROT ID, commentary, and the literature citation.

The JSON reflects the data's hierarchical organization. At the highest Each EC number is paired with its subcategories. As the BRENDA textfile excluded categories, and as each subcategory is distinctly different from other subcategories, we decided not to include the category. Each subcategory is paired with an array reflecting the corresponding structure of the table in BRENDA. Each array element represents a row of the table. The array element is structured as an object, where attributes are names of the table column and values are the corresponding table values. The "perBRENDAParser" command line performs this translation task.

2.2 Extracting Data from JSON


A. Tree organization of BRENDA database, with tables as leaf nodes
1.1.1.1
Enzyme-Ligand Interactions
Enzyme Nomenclature
...
Activating Compound
Cofactors
....
1.1.1.3
Enzyme-Ligand Interactions
Enzyme Nomenclature
...
6.5.4.3
B. BRENDA database entry
C. BRENDA textfile for database entry
D. BRENDA JSON from PER BRENDA mirrors BRENDA entry

ACTIVATING COMPOUND	ORGANISM	UNIPROT	COMMENTARY	LITERATURE
Glutaraldehyde	Saccharomyces cerevisiae	-	treating with 0.5% glutaraldehyde solution, the activity of the immobilized enzyme is at maximum	695705

```
AC   #10# Glutaraldehyde (#10# treating with 0.5% glutaraldehyde solution,
     the activity of the immobilized  nzyme is at maximum <196>) <196>
```

```
{
  "1.1.1.1": {
    "AC": [
      {
        "ACTIVATING COMPOUND": "Glutaraldehyde",
        "ORGANISM": "Saccharomyces cerevisiae",
        "UNIPROT": null,
        "COMMENTARY": [
          "treating with 0.5% glutaraldehyde solution, the activity of the immobilized enzyme is at maximum <196>"
        ],
        "LITERATURE": [
          "196"
        ],
      },
```

Fig. 1. Translating the BRENDA textfile to BRENDA JSON.

The parsed BRENDA JSON is queried using The perBRENDAExtract function. The function parameters are a parsed JSON file name, output file name, and query parameters, which include the EC numbers, subcategories, and table column names of interest. An example use of this function is BRENDAExtract(EC = ['1.1.3.3', '5.4.1.1.'], subcategory = 'KM', fields = ['KM VALUE', 'SUBSTRATE', 'ORGANISM']). The default extraction is a JSON object; however, an optional parameter allows for extracting the file in a comma separated textfile. We also generate a textfile of the compound names involved in every query. This textfile can be submitted to compound naming services such as MetaboAnalyst (Xia, et al., 2015) or The Chemical Translation Service (Wohlgemuth, et al., 2010).

3 Implementation and Performances

The total runtime of PER BRENDA on the downloaded BREDNA textfile is 234 seconds on a personal laptop and outputs a 1.38 GB JSON. Queries to the JSON file are fast, with most taking up 1-2 minutes depending on the complexity of the query. The pandas package is utilized to facilitate the comma-separate text file queries.

Acknowledgements

Funding

This research is supported by NSF, Award CCF-1909536, and also by NIGMS of the National Institutes of Health, Award R01GM132391. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Conflict of Interest: none declared.

References

Jeske, L.*, et al.* BRENDA in 2019: a European ELIXIR core data resource. *Nucleic acids research* 2019;47(D1):D542-D549.

Liebecq, C. IUPAC-IUBMB Joint Commission on Biochemical Nomenclature (JCBN) and Nomenclature Committee of IUBMB (NC-IUBMB). Newsletter 1996. *European journal of biochemistry* 1997;247(2):733-739.

Wohlgemuth, G.*, et al.* The Chemical Translation Service—a web-based tool to improve standardization of metabolomic reports. *Bioinformatics* 2010;26(20):2647-2648.

Xia, J.*, et al.* MetaboAnalyst 3.0—making metabolomics more meaningful. *Nucleic acids research* 2015;43(W1):W251-W257.