Journal of Cheminformatics

DATABASE

Open Access

The Open Spectral Database: an open platform for sharing and searching spectral data

Stuart J. Chalk*

Abstract

Background: A number of websites make available spectral data for download (typically as JCAMP-DX text files) and one (ChemSpider) that also allows users to contribute spectral files. As a result, searching and retrieving such spectral data can be time consuming, and difficult to reuse if the data is compressed in the JCAMP-DX file. What is needed is a single resource that allows submission of JCAMP-DX files, export of the raw data in multiple formats, searching based on multiple chemical identifiers, and is open in terms of license and access. To address these issues a new online resource called the Open Spectral Database (OSDB) http://osdb.info/ has been developed and is now available. Built using open source tools, using open code (hosted on GitHub), providing open data, and open to community input about design and functionality, the OSDB is available for anyone to submit spectral data, making it searchable and available to the scientific community. This paper details the concept and coding, internal architecture, export formats, Representational State Transfer (REST) Application Programming Interface and options for submission of data.

Results: The OSDB website went live in November 2015. Concurrently, the GitHub repository was made available at https://github.com/stuchalk/OSDB/, and is open for collaborators to join the project, submit issues, and contribute code.

Conclusion: The combination of a scripting environment (PHPStorm), a PHP Framework (CakePHP), a relational database (MySQL) and a code repository (GitHub) provides all the capabilities to easily develop REST based websites for ingestion, curation and exposure of open chemical data to the community at all levels. It is hoped this software stack (or equivalent ones in other scripting languages) will be leveraged to make more chemical data available for both humans and computers.

Keywords: Spectral data, REST API, Open science, Open data, JCAMP-DX, XML, Scientific data model

Background

Tools to make research data freely available are vitally important to the open science movement. Such tools must play well with both humans and computers because of the importance of data import/export into other systems for analysis, verification, and data mining. One important data type in this area is instrumental spectra, used for identification and analysis in a variety of different application areas. Many websites (e.g. NIST Webbook [1], ChemSpider [2], University of the West Indies—Chemistry [3]) contain spectral files available in the current de-facto data standard, Joint Committee on Atomic and Molecular Physical Data—Data Exchange format (JCAMP-DX) [4–7] and this format can be exported from the majority of instrument software available today. However, the usefulness of spectral data in JCAMP-DX format is somewhat limited due to the specification being over 30 years old, and if saved using compression, difficult to transfer to other software. Providing a mechanism to allow conversion of legacy data in JCAMP-DX format is an important activity in-of-itself, as the community needs spectral data for comparison/standardization in many different applications.

In order to leverage data in JCAMP-DX and make it more easily available and searchable, a website has been developed [8] that allows (1) upload of JCAMP-DX

*Correspondence: schalk@unf.edu
Department of Chemistry, University of North Florida, Jacksonville, FL 32224, USA

files, (2) extraction and conversion of the data and metadata to an extensible markup language (XML) pseudo JCAMP-DX equivalent, and (3) encoding of the data in JavaScript Object Notation for Linked Data (JSON-LD) [9] using a generic scientific data model (SDM) [10]. Each of these three formats is available for download.

The website has been developed using open-source software (as far as possible), using open standards, and is openly being made available using the GitHub code repository. The website is built in the Representational State Transfer (REST) style [11] and has a documented Application Programming Interface (API) [12] for computer based discovery and export.

Implementation

The foundation of the OSDB website is the common Apache [13], MySQL [14], and PHP [15] software stack that can be installed on any computer system as: LAMP (for Linux), WAMP (for Windows) and MAMP [for OSX (Mac)]. Coding was done using the PHPStorm [16] Integrated Development Environment (IDE) (free for faculty and students) and scripts are written in PHP implementing the CakePHP object oriented framework [17]. Because of the use of this standard open-source software developers can either; deploy on their own physical server, publish using one of a number of online hosting sites, or use a virtual machine, for creation of data websites.

Website functionality

As the goal of the project was to develop a site that offered a standardized REST style API, the PHP framework CakePHP was used to develop the code. CakePHP standardizes the development of PHP scripts by use of the model–view–controller (MVC) [18] model, implemented using Object Oriented Programming (OOP) [19]. The MVC paradigm separates code into logical sections; the model—to access a database table, the view—presentation of data as a web page, and the controller—the logic that coordinates the processing of a webpage request into a Hypertext Markup Language (HTML) document. As an example, if the user goes to the systems index page [20], then the following code is executed (Fig. 1).

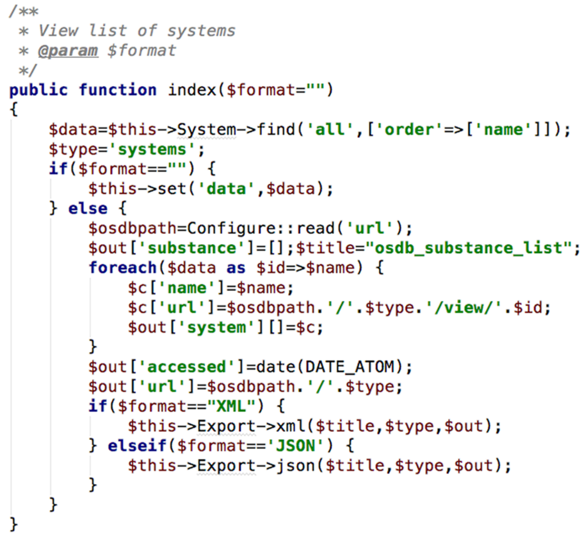
The function 'index' in the SystemsController.php file executes a set of PHP commands to present a list of current chemical systems in the database, and is one of a number of methods of the SystemsController class. This is executed by default where is no action after "/systems" in the URL. In the first line the call to `$this->System->find` accesses the System 'model' (that accesses the 'systems' database table) and executes the find 'method' to retrieve all the systems, and place the data in the `$data` variable. The `$this->set` command assigns the data returned to a variable called 'data' that will be available to the PHP code in the view file. For "/systems", once the code in the controller has finished, CakePHP knows to then return the 'index.ctp' file (CakePHP template file) to the browser—shown in Fig. 2.

Note that the function in Fig. 1 has a variable (`$format`) in the function arguments and when no value is passed the default of an empty string is set. When `$format` is tested as being equal to an empty string the `$this->set` command completes and the HTML file 'index.ctp' is rendered. However, if the URL "/systems/index/XML" is accessed, the same call is made except the data is not sent to the view and is instead is converted to an array and reformatted as XML (`$this->Export->xml()`) and passed to the browser. Note that the action 'index' must be included in the URL so that CakePHP does not try and run the action 'XML' (i.e. "/systems/XML" will cause an error).

The view file takes the list of all the systems in `$data` (view variable), iterates through each one (`$data` is a PHP array—see Fig. 3a) and prints out a HTML link on the webpage as an unordered list "<ul>" and shown in Fig. 3b.

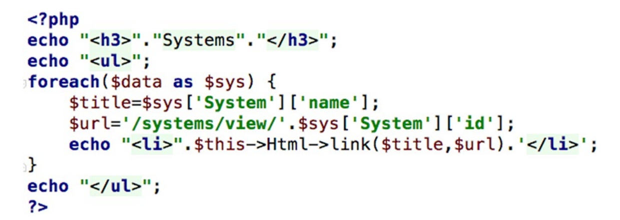
Spectral file format

The Joint Committee on Atomic and Molecular Physical Data (JCAMP) published the specification for the Data eXchange format (DX) for spectral data for UV/Vis and



```
/**
 * View list of systems
 * @param $format
 */
public function index($format="")
{
    $data=$this->System->find('all',['order'=>['name']]);
    $type='systems';
    if($format=="") {
        $this->set('data',$data);
    } else {
        $osdbpath=Configure::read('url');
        $out['substance']=[];$title="osdb_substance_list";
        foreach($data as $id=>$name) {
            $c['name']=$name;
            $c['url']=$osdbpath.'/'.$type.'/view/'.$id;
            $out['system'][]=$c;
        }
        $out['accessed']=date(DATE_ATOM);
        $out['url']=$osdbpath.'/'.$type;
        if($format=="XML") {
            $this->Export->xml($title,$type,$out);
        } elseif($format=='JSON') {
            $this->Export->json($title,$type,$out);
        }
    }
}
```

Fig. 1 CakePHP controller code to retrieve chemical system data and pass it to the view file



```
<?php
echo "<h3>"."Systems"."</h3>";
echo "<ul>";
foreach($data as $sys) {
    $title=$sys['System']['name'];
    $url='/systems/view/'.$sys['System']['id'];
    echo "<li>".$this->Html->link($title,$url).'</li>';
}
echo "</ul>";
?>
```

Fig. 2 PHP snippet used to display and index of chemical systems

IR [6], MS [7], IMS [21], NMR [5], ESR [22], and CD [23] data. JCAMP-DX files are ASCII text files populated with LABELLED-DATA-RECORDs or LDRs. These are defined to allow reporting of spectral metadata and raw/processed instrument data. The instrument data is reported as XY pairs, nominally in tabular format where one line contains a starting X value and a number of equally spaced Y values, the X value of which can be calculated using the LDR DELTAX. In addition, because the format was developed when disk space was at a premium, the data can be reported in a number of compressed formats, referred to as ASCII Squeezed Difference Form (ASDF), that use letters and symbols to encode data in more compact formats (Table 1).

Table 2 shows data in normal fixed format and equivalent storage in four compression formats.

When a JCAMP file is uploaded to the OSDB website, a record is added to the database that contains the file metadata. A unique id is generated and the file is saved with the id as its filename (available at id.jdx). The file is then read into a variable in PHP and processed line by line using a JCAMP plugin written in our laboratory. The plugin processes the file using the following seven steps:

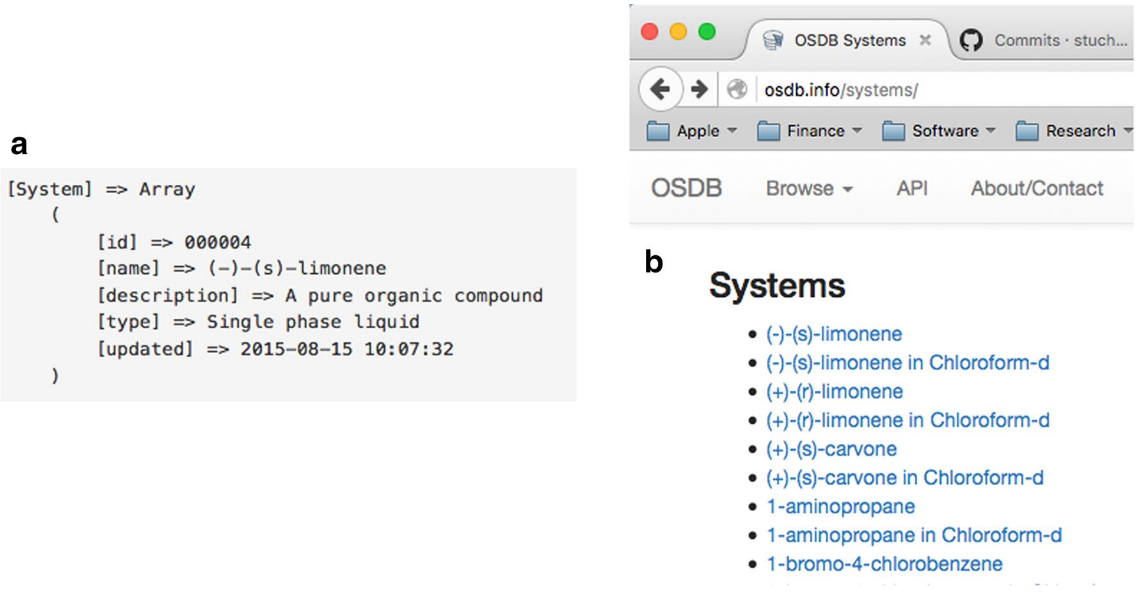
- Clean—remove non-ACSII characters extra spaces at the start/end of lines
- Uncomment—remove (and save) comments (indicated by $$)
- Get LDRs—detect LDRs in the file
- Validate—check the LDRs to identify if the file is valid JCAMP-DX
- Standardize—standardize data in certain LDR fields
- Decompress—expand data in any of the ASDF formats and calculates respective X values

The metadata, data, comments and any processing errors are stored in an array in PHP and then converted to XML and saved. Figures 4 and 5 show a comparison of the original JCAMP file (e.g. "../spectra/view/000000115/JCAMP") and JCAMP saved in XML (e.g. "../spectra/view/000000115/XML").

Spectral data in the JCAMP file is stored both in its <raw> state along with the <pro> (cessed) expanded format. Any discrepancies between the data in the original JCAMP file and the process data are annotated in the errors element, with details of the issues.

Data management

The spectral data is also saved in a MySQL database designed around the SDM outlined in the recent paper [24]. The SDM is a generic framework of organizing the data and metadata obtained in scientific experiments. In a nutshell,



```
[System] => Array
    (
        [id] => 000004
        [name] => (-)-(s)-limonene
        [description] => A pure organic compound
        [type] => Single phase liquid
        [updated] => 2015-08-15 10:07:32
    )
```

Systems

- (-)-(s)-limonene
- (-)-(s)-limonene in Chloroform-d
- (+)-(r)-limonene
- (+)-(r)-limonene in Chloroform-d
- (+)-(s)-carvone
- (+)-(s)-carvone in Chloroform-d
- 1-aminopropane
- 1-aminopropane in Chloroform-d
- 1-bromo-4-chlorobenzene

Fig. 3 System data in **a** PHP array, **b** a web page

Table 1 Pseudo-digits for ASDF

1. ASCII digits	0	1	2	3	4	5	6	7	8	9
2. Positive SQZ digits	@	A	B	C	D	E	F	G	H	I
3. Negative SQZ digits		a	b	c	d	e	f	g	h	i
4. Positive DIF digits	%	J	j	L	M	N	O	P	Q	R
5. Negative DIF digits		j	l	l	m	n	o	p	q	r
6. Positive DUP digits		S	T	U	V	V	W	X	Y	Z

Table 2 Example of ASDF Formats (only Y data points shown)

FIX form: (22 chars)	1	2	3	3	2	1	0	−1	−2	−3
PAC form: (19 chars)	1 + 2 + 3 + 3 + 2 + 1 + 0 − 1 − 2 − 3									
Or:	1 2 3 3 2 1 0 − 1 − 2 − 3									
SQZ form: (10 chars)	1BCCBA@abc									
DIF form: (10 chars)	1JJ%jjjjjj									
DIFDUP form: (7 chars)	1JT%jX									

the SDM describes data and its metadata containers that can be used to aggregate scientific data and metadata and relate them together. It can be implemented in any database, XML Markup Language, or text format, for example in JavaScript Object Notation for Linked Data, JSON-LD [9]. If implemented in JSON-LD the information can be semantically represented using a set of SDM JSON-LD context files (see Fig. 6). More information and examples of the SDM format can be found on the projects website [10].

Graphical user interface

A common problem currently with website design is building a graphical user interface (GUI) that works equally well on computers, tablets, and phones. Thankfully, the developers of Twitter have built a free CSS/JavaScript solution to this problem called Bootstrap [25]. Bootstrap uses predefined Cascading Style Sheet (CSS) [26] classes that are designed to work with HTML 5 [27] and plugins for the popular jQuery [28] JavaScript library. As a result, the basic Bootstrap implementation produces a clean, browser natural, device neutral GUI that minimizes the development of the site interface.

Results

The website allows users to access the data in the system via endpoints (MVC controller) for:

- Systems [20]
- Spectra [29]
- Compounds [30]
- (Analytical) Techniques [31]
- Collections [32]

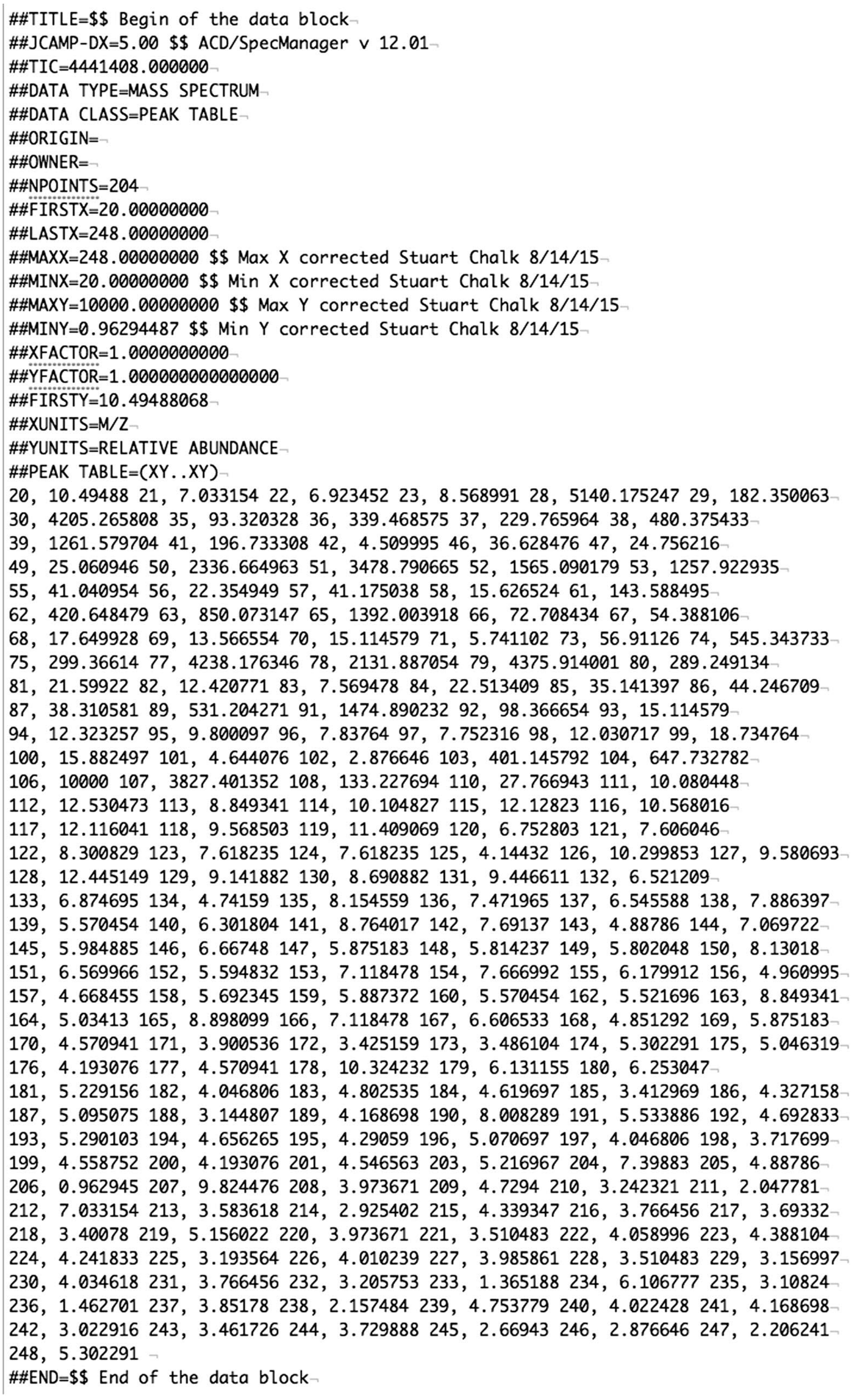
The default pages accessed via "/<controllername>" URLs all provide an index of available resources of that type. For instance, Fig. 7 shows the spectra index, with spectra organized by compound, with a JSmol [33] view of the structure, a link to the compound page, and external links to more data about the compound.

In addition to the index view for spectra, users can search of a specific compound using the search box at the top of the page. The search is performed over the 'identifier' database table which is populated from the PubChem PUGREST interface [34] and contains names, SMILES, PubChem CIDs, InChI strings and InChIKeys.

Clicking on the name of a compound on the compound index page [30] brings up a summary page of a compound with the JSmol molecular view, metadata, and links to the spectra for that compound. Figure 8 shows the page for aspirin. Note the external links to view the chemical on PubChem [35] and Wikidata [36] a free linked database that underpins a number of websites including Wikipedia.

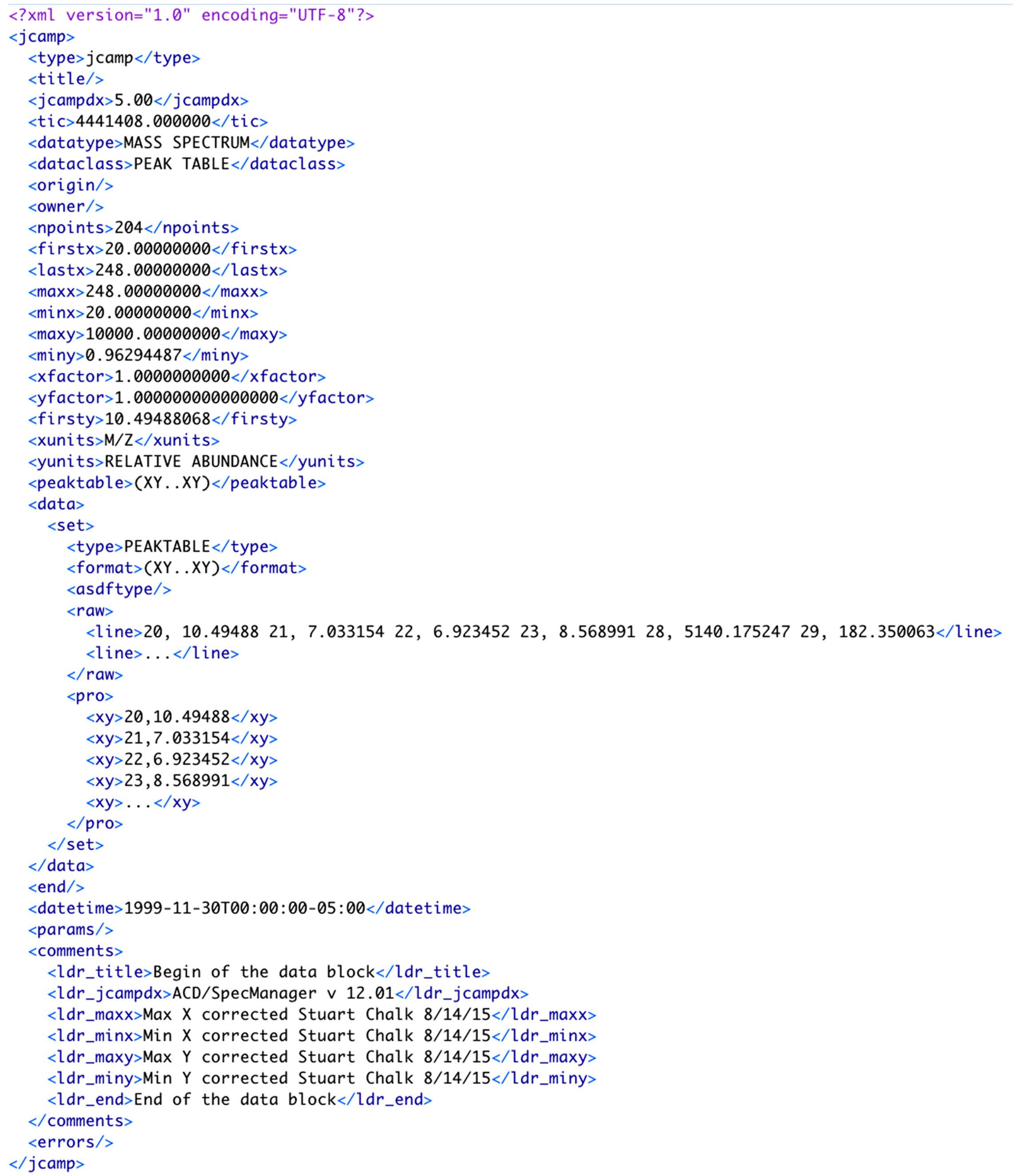
Clicking on a spectral link (under Systems and Spectra) brings up the view of the spectrum (see Fig. 9 for the MS spectrum of Aspirin). Metadata about the spectrum is available on the left side by clicking each of the four buttons. The spectrum can also be downloaded in JCAMP, XML, and SciData (JSON-LD) formats by clicking the respective icon.

To contribute a spectrum to the OSDB, users first signup for an account under "My OSDB" (top right) and then click on "Add Spectra" on the top menu bar. The form for upload of JCAMP files (Fig. 10a) can be used to upload a file from a local drive or a web address. When entering the



```
##TITLE=$$ Begin of the data block¬
##JCAMP-DX=5.00 $$ ACD/SpecManager v 12.01¬
##TIC=4441408.000000¬
##DATA TYPE=MASS SPECTRUM¬
##DATA CLASS=PEAK TABLE¬
##ORIGIN=¬
##OWNER=¬
##NPOINTS=204¬
##FIRSTX=20.00000000¬
##LASTX=248.00000000¬
##MAXX=248.00000000 $$ Max X corrected Stuart Chalk 8/14/15¬
##MINX=20.00000000 $$ Min X corrected Stuart Chalk 8/14/15¬
##MAXY=10000.00000000 $$ Max Y corrected Stuart Chalk 8/14/15¬
##MINY=0.96294487 $$ Min Y corrected Stuart Chalk 8/14/15¬
##XFACTOR=1.0000000000¬
##YFACTOR=1.000000000000000¬
##FIRSTY=10.49488068¬
##XUNITS=M/Z¬
##YUNITS=RELATIVE ABUNDANCE¬
##PEAK TABLE=(XY..XY)¬
20, 10.49488 21, 7.033154 22, 6.923452 23, 8.568991 28, 5140.175247 29, 182.350063¬
30, 4205.265808 35, 93.320328 36, 339.468575 37, 229.765964 38, 480.375433¬
39, 1261.579704 41, 196.733308 42, 4.509995 46, 36.628476 47, 24.756216¬
49, 25.060946 50, 2336.664963 51, 3478.790665 52, 1565.090179 53, 1257.922935¬
55, 41.040954 56, 22.354949 57, 41.175038 58, 15.626524 61, 143.588495¬
62, 420.648479 63, 850.073147 65, 1392.003918 66, 72.708434 67, 54.388106¬
68, 17.649928 69, 13.566554 70, 15.114579 71, 5.741102 73, 56.91126 74, 545.343733¬
75, 299.36614 77, 4238.176346 78, 2131.887054 79, 4375.914001 80, 289.249134¬
81, 21.59922 82, 12.420771 83, 7.569478 84, 22.513409 85, 35.141397 86, 44.246709¬
87, 38.310581 89, 531.204271 91, 1474.890232 92, 98.366654 93, 15.114579¬
94, 12.323257 95, 9.800097 96, 7.83764 97, 7.752316 98, 12.030717 99, 18.734764¬
100, 15.882497 101, 4.644076 102, 2.876646 103, 401.145792 104, 647.732782¬
106, 10000 107, 3827.401352 108, 133.227694 110, 27.766943 111, 10.080448¬
112, 12.530473 113, 8.849341 114, 10.104827 115, 12.12823 116, 10.568016¬
117, 12.116041 118, 9.568503 119, 11.409069 120, 6.752803 121, 7.606046¬
122, 8.300829 123, 7.618235 124, 7.618235 125, 4.14432 126, 10.299853 127, 9.580693¬
128, 12.445149 129, 9.141882 130, 8.690882 131, 9.446611 132, 6.521209¬
133, 6.874695 134, 4.74159 135, 8.154559 136, 7.471965 137, 6.545588 138, 7.886397¬
139, 5.570454 140, 6.301804 141, 8.764017 142, 7.69137 143, 4.88786 144, 7.069722¬
145, 5.984885 146, 6.66748 147, 5.875183 148, 5.814237 149, 5.802048 150, 8.13018¬
151, 6.569966 152, 5.594832 153, 7.118478 154, 7.666992 155, 6.179912 156, 4.960995¬
157, 4.668455 158, 5.692345 159, 5.887372 160, 5.570454 162, 5.521696 163, 8.849341¬
164, 5.03413 165, 8.898099 166, 7.118478 167, 6.606533 168, 4.851292 169, 5.875183¬
170, 4.570941 171, 3.900536 172, 3.425159 173, 3.486104 174, 5.302291 175, 5.046319¬
176, 4.193076 177, 4.570941 178, 10.324232 179, 6.131155 180, 6.253047¬
181, 5.229156 182, 4.046806 183, 4.802535 184, 4.619697 185, 3.412969 186, 4.327158¬
187, 5.095075 188, 3.144807 189, 4.168698 190, 8.008289 191, 5.533886 192, 4.692833¬
193, 5.290103 194, 4.656265 195, 4.29059 196, 5.070697 197, 4.046806 198, 3.717699¬
199, 4.558752 200, 4.193076 201, 4.546563 203, 5.216967 204, 7.39883 205, 4.88786¬
206, 0.962945 207, 9.824476 208, 3.973671 209, 4.7294 210, 3.242321 211, 2.047781¬
212, 7.033154 213, 3.583618 214, 2.925402 215, 4.339347 216, 3.766456 217, 3.69332¬
218, 3.40078 219, 5.156022 220, 3.973671 221, 3.510483 222, 4.058996 223, 4.388104¬
224, 4.241833 225, 3.193564 226, 4.010239 227, 3.985861 228, 3.510483 229, 3.156997¬
230, 4.034618 231, 3.766456 232, 3.205753 233, 1.365188 234, 6.106777 235, 3.10824¬
236, 1.462701 237, 3.85178 238, 2.157484 239, 4.753779 240, 4.022428 241, 4.168698¬
242, 3.022916 243, 3.461726 244, 3.729888 245, 2.66943 246, 2.876646 247, 2.206241¬
248, 5.302291 ¬
##END=$$ End of the data block¬
```

Fig. 4 JCAMP-DX file

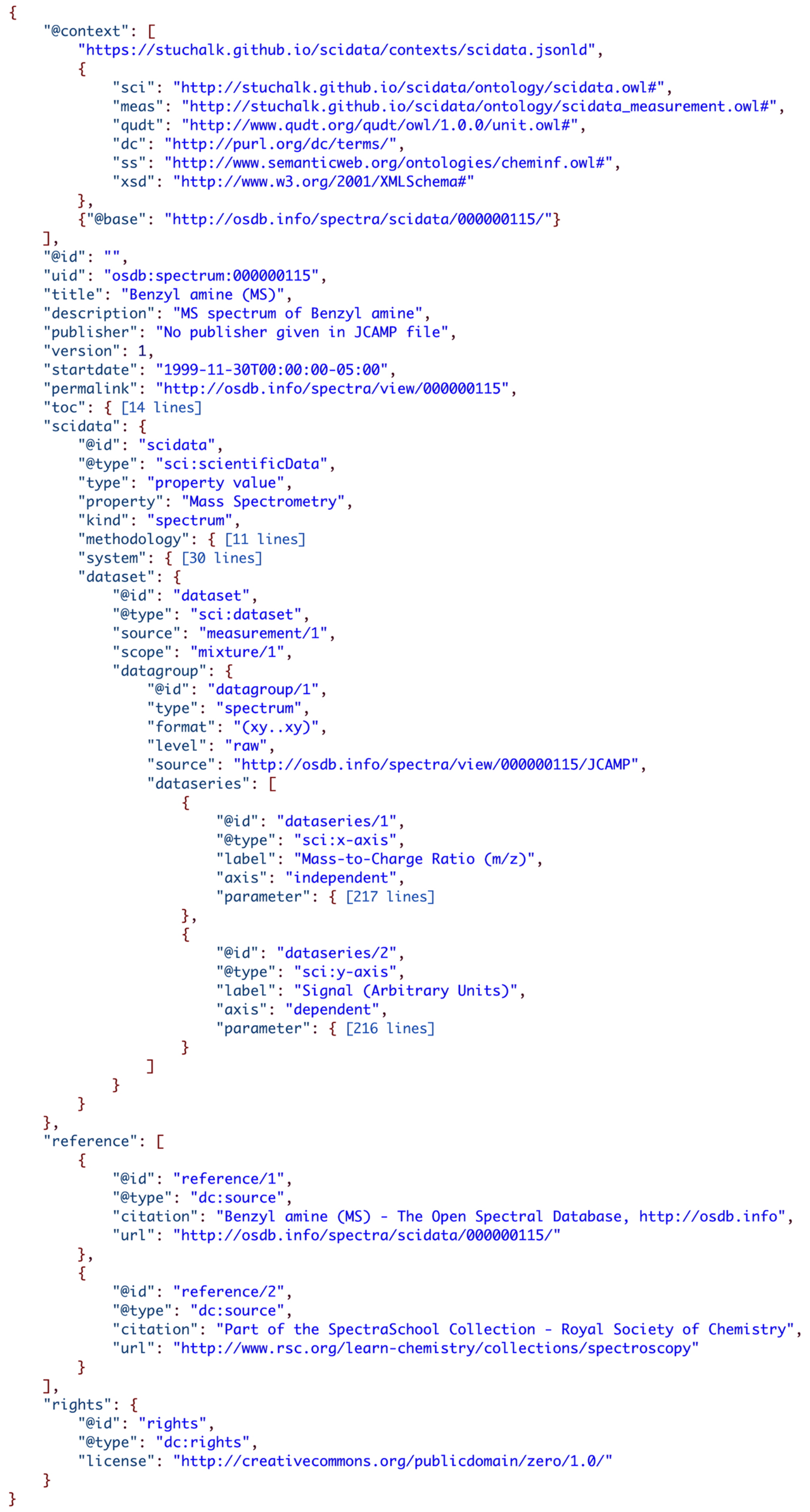


```
<?xml version="1.0" encoding="UTF-8"?>
<jcamp>
  <type>jcamp</type>
  <title/>
  <jcampdx>5.00</jcampdx>
  <tic>4441408.000000</tic>
  <datatype>MASS SPECTRUM</datatype>
  <dataclass>PEAK TABLE</dataclass>
  <origin/>
  <owner/>
  <npoints>204</npoints>
  <firstx>20.00000000</firstx>
  <lastx>248.00000000</lastx>
  <maxx>248.00000000</maxx>
  <minx>20.00000000</minx>
  <maxy>10000.00000000</maxy>
  <miny>0.96294487</miny>
  <xfactor>1.0000000000</xfactor>
  <yfactor>1.000000000000000</yfactor>
  <firsty>10.49488068</firsty>
  <xunits>M/Z</xunits>
  <yunits>RELATIVE ABUNDANCE</yunits>
  <peaktable>(XY..XY)</peaktable>
  <data>
    <set>
      <type>PEAKTABLE</type>
      <format>(XY..XY)</format>
      <asdftype/>
      <raw>
        <line>20, 10.49488 21, 7.033154 22, 6.923452 23, 8.568991 28, 5140.175247 29, 182.350063</line>
        <line>...</line>
      </raw>
      <pro>
        <xy>20,10.49488</xy>
        <xy>21,7.033154</xy>
        <xy>22,6.923452</xy>
        <xy>23,8.568991</xy>
        <xy>...</xy>
      </pro>
    </set>
  </data>
  <end/>
  <datetime>1999-11-30T00:00:00-05:00</datetime>
  <params/>
  <comments>
    <ldr_title>Begin of the data block</ldr_title>
    <ldr_jcampdx>ACD/SpecManager v 12.01</ldr_jcampdx>
    <ldr_maxx>Max X corrected Stuart Chalk 8/14/15</ldr_maxx>
    <ldr_minx>Min X corrected Stuart Chalk 8/14/15</ldr_minx>
    <ldr_maxy>Max Y corrected Stuart Chalk 8/14/15</ldr_maxy>
    <ldr_miny>Min Y corrected Stuart Chalk 8/14/15</ldr_miny>
    <ldr_end>End of the data block</ldr_end>
  </comments>
  <errors/>
</jcamp>
```

Fig. 5 JCAMP-DX file in XML format

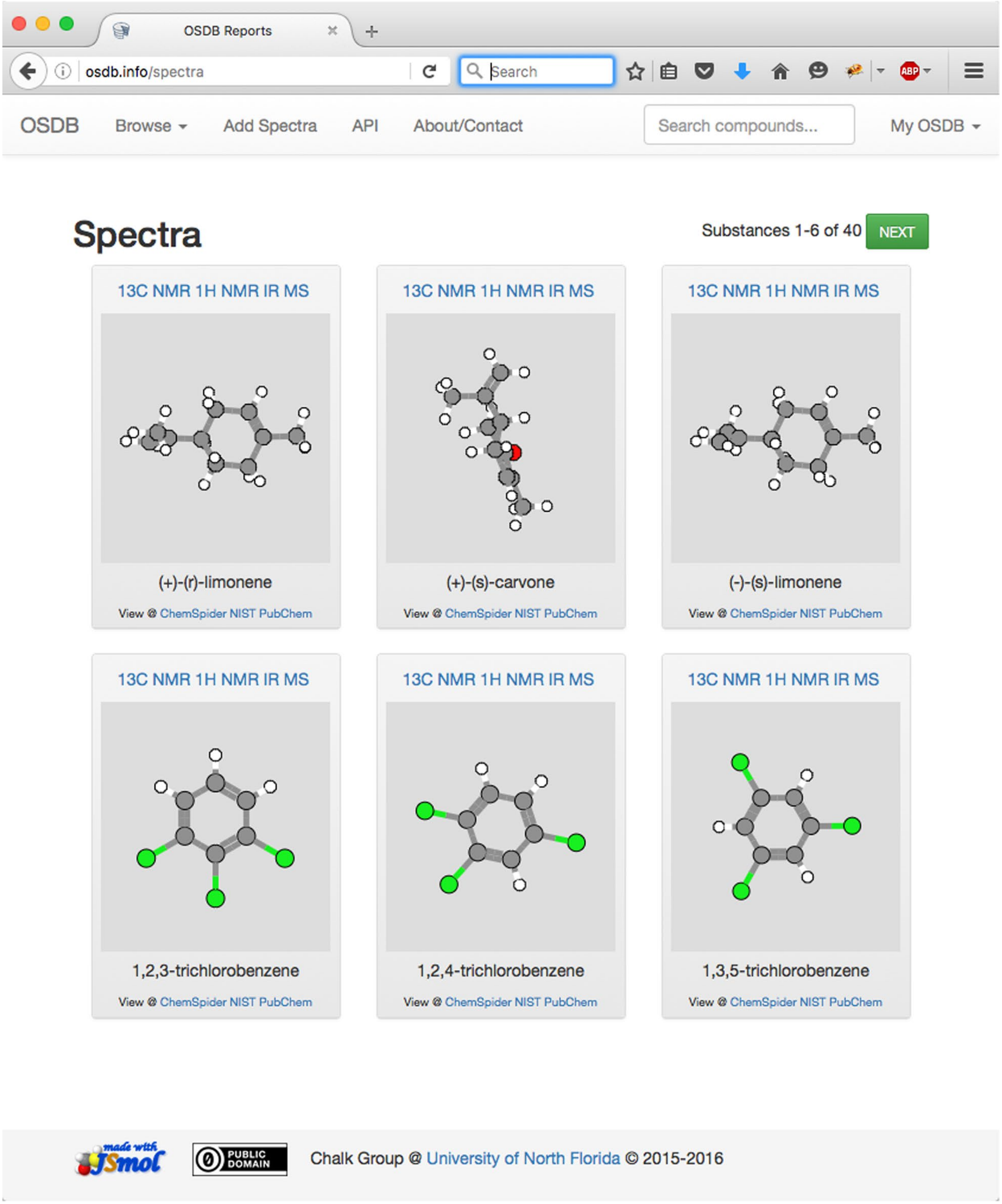
compound name the form searches and displays (Fig. 10b) the existing compounds in the system and clicking on one of the names found selects that compound. If the user is uploading local files they can add as many as they like by clicking the “Add another file” button (Fig. 10c).

For computer access to all of the above functionality (except file upload) a REST API is available and described here [37] (Fig. 11) built using the widely popular Swagger API framework [38]. As an example for spectral data the API allows access to the files via a number of formats—OSDB ID, Splash [39], and compound name and technique code (comp|tech). It is also possible to access just the plot of the spectrum which is useful for embedding the data into another website.



```
{
    "@context": [
        "https://stuchalk.github.io/scidata/contexts/scidata.jsonld",
        {
            "sci": "http://stuchalk.github.io/scidata/ontology/scidata.owl#",
            "meas": "http://stuchalk.github.io/scidata/ontology/scidata_measurement.owl#",
            "qudt": "http://www.qudt.org/qudt/owl/1.0.0/unit.owl#",
            "dc": "http://purl.org/dc/terms/",
            "ss": "http://www.semanticweb.org/ontologies/cheminf.owl#",
            "xsd": "http://www.w3.org/2001/XMLSchema#"
        },
        {"@base": "http://osdb.info/spectra/scidata/000000115/"}
    ],
    "@id": "",
    "uid": "osdb:spectrum:000000115",
    "title": "Benzyl amine (MS)",
    "description": "MS spectrum of Benzyl amine",
    "publisher": "No publisher given in JCAMP file",
    "version": 1,
    "startdate": "1999-11-30T00:00:00-05:00",
    "permalink": "http://osdb.info/spectra/view/000000115",
    "toc": { [14 lines]
    "scidata": {
        "@id": "scidata",
        "@type": "sci:scientificData",
        "type": "property value",
        "property": "Mass Spectrometry",
        "kind": "spectrum",
        "methodology": { [11 lines]
        "system": { [30 lines]
        "dataset": {
            "@id": "dataset",
            "@type": "sci:dataset",
            "source": "measurement/1",
            "scope": "mixture/1",
            "datagroup": {
                "@id": "datagroup/1",
                "type": "spectrum",
                "format": "(xy..xy)",
                "level": "raw",
                "source": "http://osdb.info/spectra/view/000000115/JCAMP",
                "dataseries": [
                    {
                        "@id": "dataseries/1",
                        "@type": "sci:x-axis",
                        "label": "Mass-to-Charge Ratio (m/z)",
                        "axis": "independent",
                        "parameter": { [217 lines]
                    },
                    {
                        "@id": "dataseries/2",
                        "@type": "sci:y-axis",
                        "label": "Signal (Arbitrary Units)",
                        "axis": "dependent",
                        "parameter": { [216 lines]
                    }
                ]
            }
        }
    },
    "reference": [
        {
            "@id": "reference/1",
            "@type": "dc:source",
            "citation": "Benzyl amine (MS) - The Open Spectral Database, http://osdb.info",
            "url": "http://osdb.info/spectra/scidata/000000115/"
        },
        {
            "@id": "reference/2",
            "@type": "dc:source",
            "citation": "Part of the SpectraSchool Collection - Royal Society of Chemistry",
            "url": "http://www.rsc.org/learn-chemistry/collections/spectroscopy"
        }
    ],
    "rights": {
        "@id": "rights",
        "@type": "dc:rights",
        "license": "http://creativecommons.org/publicdomain/zero/1.0/"
    }
}
```

Fig. 6 Spectra data in the SciData scientific data model format



Spectra

Substances 1-6 of 40

13C NMR 1H NMR IR MS

(+)-(r)-limonene

View @ ChemSpider NIST PubChem

13C NMR 1H NMR IR MS

(+)-(s)-carvone

View @ ChemSpider NIST PubChem

13C NMR 1H NMR IR MS

(-)-(s)-limonene

View @ ChemSpider NIST PubChem

13C NMR 1H NMR IR MS

1,2,3-trichlorobenzene

View @ ChemSpider NIST PubChem

13C NMR 1H NMR IR MS

1,2,4-trichlorobenzene

View @ ChemSpider NIST PubChem

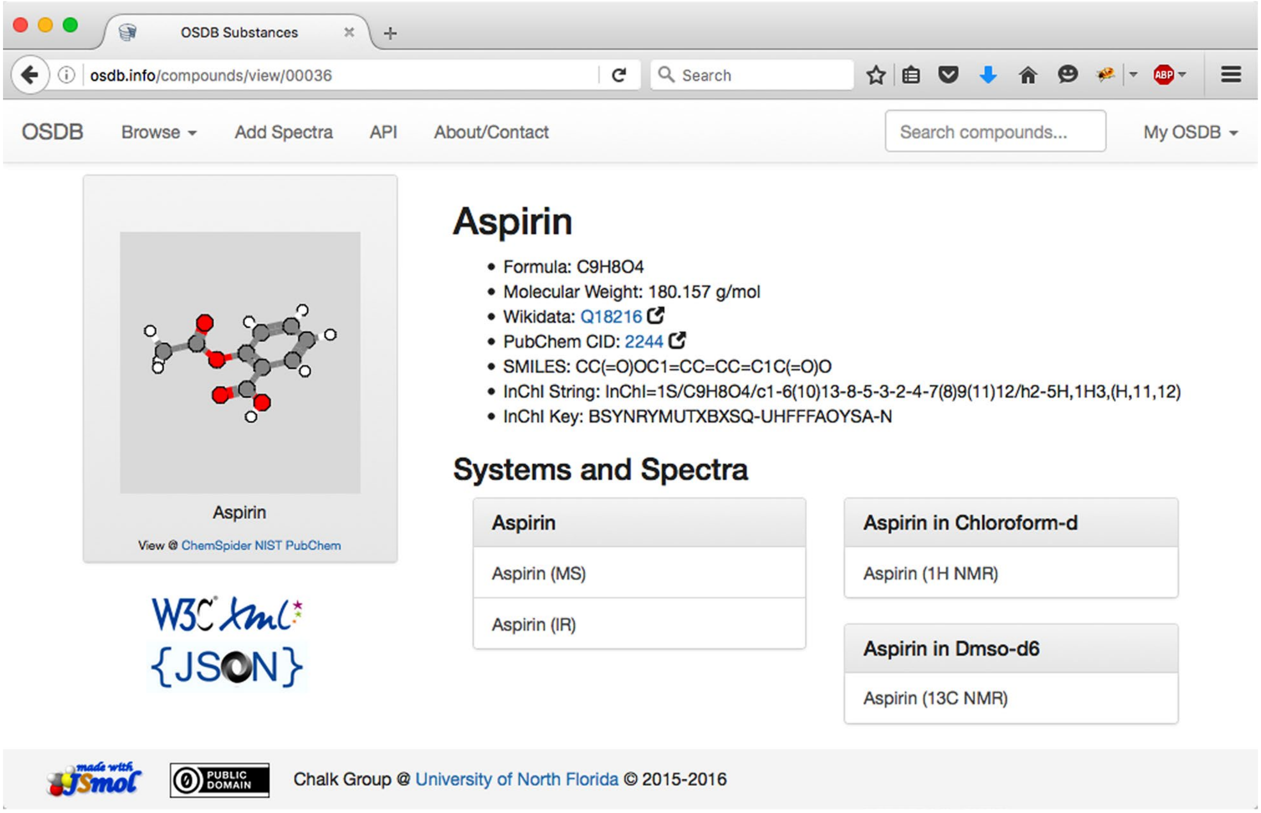
13C NMR 1H NMR IR MS

1,3,5-trichlorobenzene

View @ ChemSpider NIST PubChem

Chalk Group @ University of North Florida © 2015-2016

Fig. 7 OSDB spectral index page



Aspirin

- Formula: C9H8O4
- Molecular Weight: 180.157 g/mol
- Wikidata: Q18216
- PubChem CID: 2244
- SMILES: CC(=O)OC1=CC=CC=C1C(=O)O
- InChI String: InChI=1S/C9H8O4/c1-6(10)13-8-5-3-2-4-7(8)9(11)12/h2-5H,1H3,(H,11,12)
- InChI Key: BSYNRYMUTXBXSQ-UHFFFAOYSA-N

Systems and Spectra

Aspirin
Aspirin (MS)
Aspirin (IR)

Aspirin in Chloroform-d
Aspirin (1H NMR)

Aspirin in Dmso-d6
Aspirin (13C NMR)

Chalk Group @ University of North Florida © 2015-2016

Fig. 8 OSDB view of data for Aspirin

Features

The basic REST website provides a mechanism to add data to the repository, access it in a standardize way and download the data in multiple formats. However, the key to making the data truly accessible is by integration with other platforms and expanded search capabilities. These features have been added to the OSDB website through the following additions.

PubChem lookup for chemical metadata

When a new compound is entered on the spectra upload page and new spectrum uploaded the system does a check for the compound in the current system and if not found searches PubChem using the Power User Gateway REST API [34]. PubChem allows extensive searching of the data and metadata the system holds via the API, which has a myriad of options and has the generalized URL.

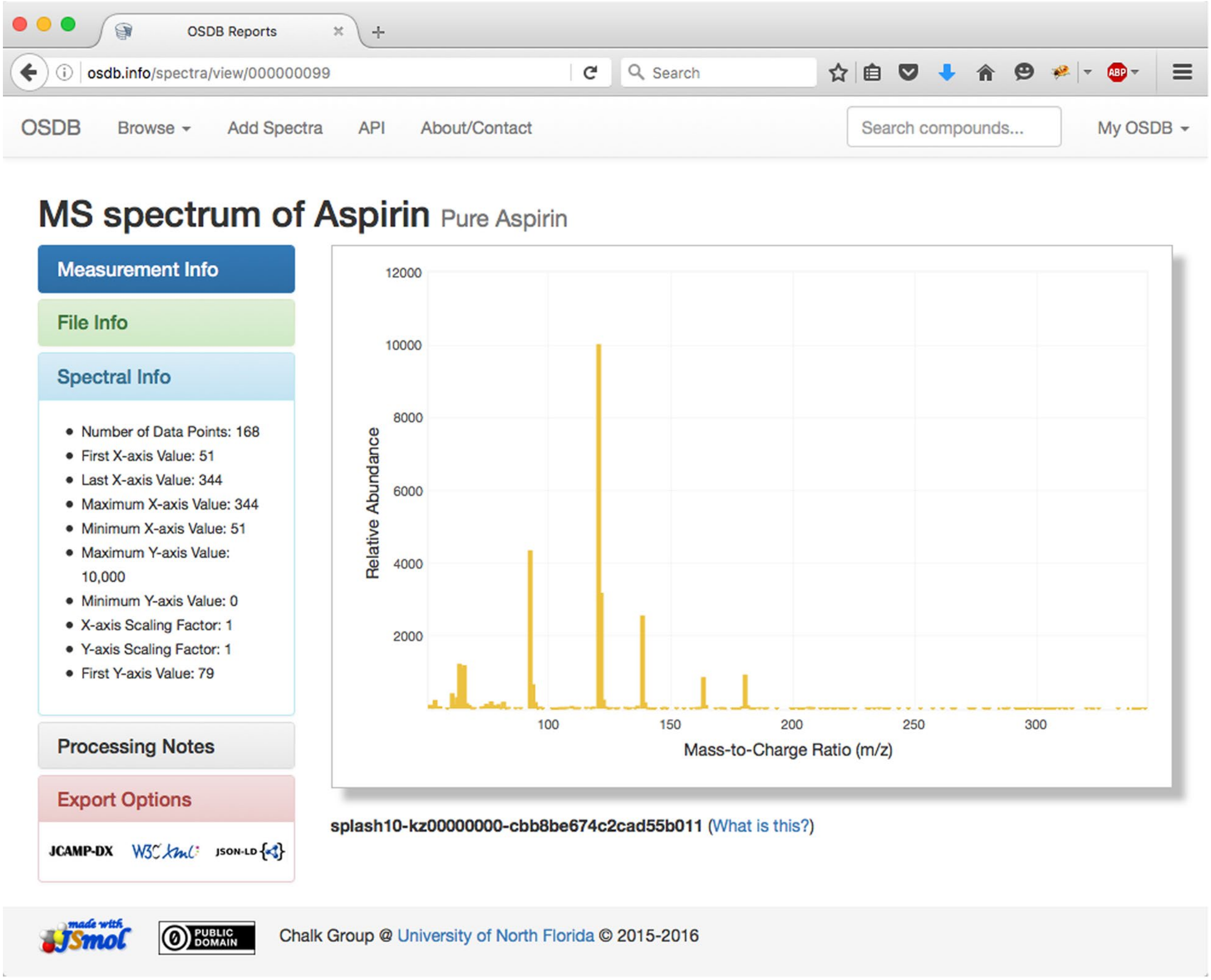
> *"http://pubchem.ncbi.nlm.nih.gov/rest/pug/ <inputspecification>/<operation specification>/ [<output specification>][?<operation_options>]"*

As an example of using this API to gather data about compounds, users submit the compound name along with the spectral data to the OSDB. Figure 12 shows a PHP function written to allow the system to retrieve the PubChem CID for the compound entered, which is subsequently used to retrieve the identifier data mentioned earlier.

The function `cid` has two arguments `$name`, and `$debug` (used to check that the code is working correctly). First, access to the CakePHP HttpSocket is established [4], and URL constructed from the base PubChem API address [2], the compound name (`$name`), and '/synonym/JSON' [8]. The URL is requested (equivalent to a web browser) [40] and the resulting JSON data converted to a PHP array `$syns` [3]. The code then checks for errors in the response [3] and retrieves the CID for the compound [5]. The value of the CID is then returned to the calling function. Other functions in the Chemical class return all the synonyms for a compound, and property data for a compound.

Retrieve Wikidata ID

Similar to the PubChem example, a function was written to search the Wikidata website [36], this time using a SPARQL query [41] encoded in a URL (Fig. 13). Three separate searches are coded to retrieve the Wikidata ID via InChIKey, SMILES, or PubChem CID. If the script



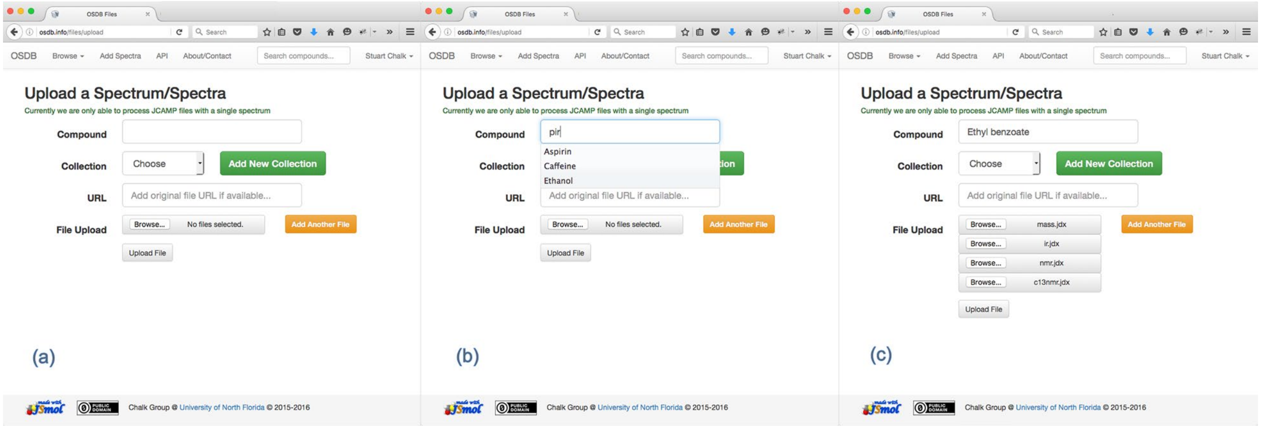
MS spectrum of Aspirin Pure Aspirin

- Number of Data Points: 168
- First X-axis Value: 51
- Last X-axis Value: 344
- Maximum X-axis Value: 344
- Minimum X-axis Value: 51
- Maximum Y-axis Value: 10,000
- Minimum Y-axis Value: 0
- X-axis Scaling Factor: 1
- Y-axis Scaling Factor: 1
- First Y-axis Value: 79

splash10-kz00000000-cbb8be674c2cad55b011 (What is this?)

Chalk Group @ University of North Florida © 2015-2016

Fig. 9 MS spectrum of aspirin



Upload a Spectrum/Spectra

Currently we are only able to process JCAMP files with a single spectrum

Compound

Collection

URL Add original file URL if available...

File Upload No files selected.

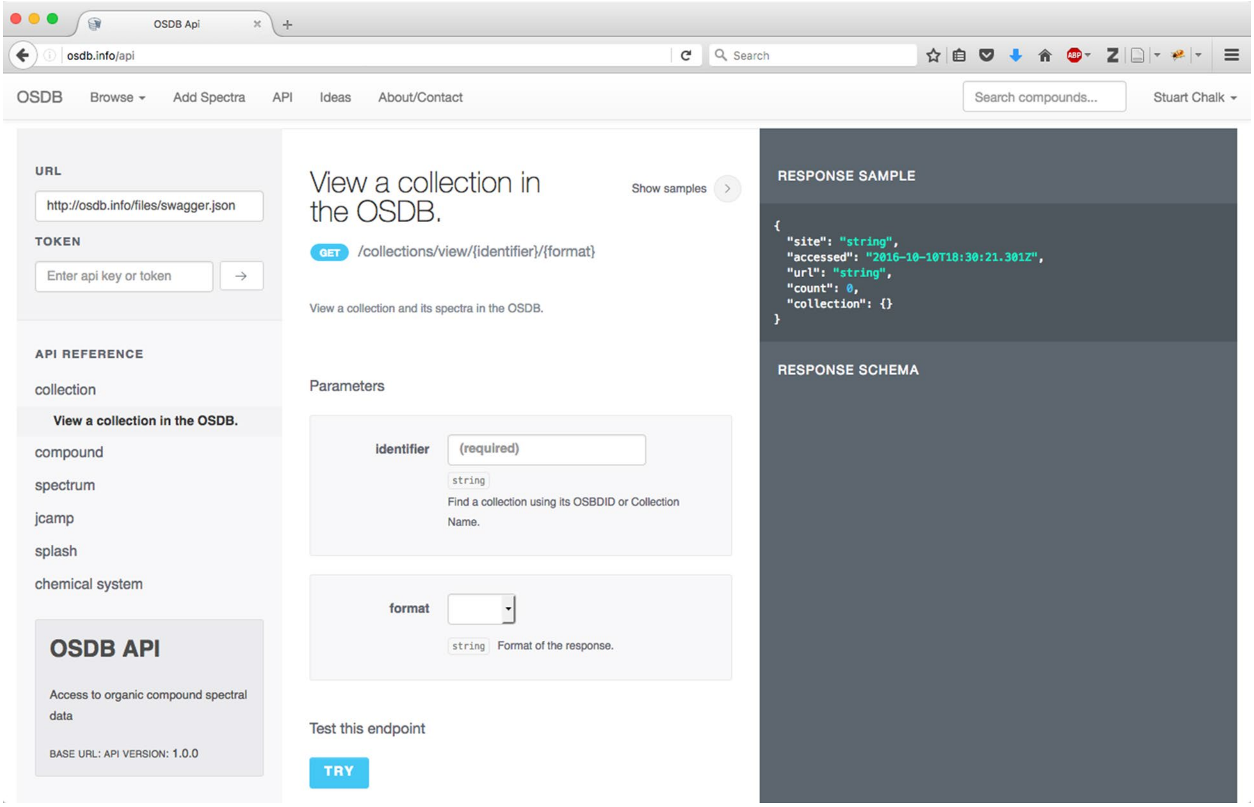
(b) Compound pir

(c) Compound Ethyl benzoate

File Upload mass.jdx ir.jdx nmr.jdx c13nmr.jdx

Chalk Group @ University of North Florida © 2015-2016

Fig. 10 Uploading a file to the OSDB. **a** Upload page, **b** compound search suggestions, and **c** adding multiple files for upload



URL

http://osdb.info/files/swagger.json

TOKEN

Enter api key or token

API REFERENCE

collection

View a collection in the OSDB.

compound

spectrum

jcamp

splash

chemical system

OSDB API

Access to organic compound spectral data

BASE URL: API VERSION: 1.0.0

View a collection in the OSDB.

GET /collections/view/{identifier}/{format}

View a collection and its spectra in the OSDB.

Parameters

identifier (required)
string
Find a collection using its OSBDID or Collection Name.

format
string Format of the response.

Test this endpoint

RESPONSE SAMPLE

```
{
  "site": "string",
  "accessed": "2016-10-10T18:30:21.301Z",
  "url": "string",
  "count": 0,
  "collection": {}
}
```

RESPONSE SCHEMA

Fig. 11 The OSDB REST API specification

calling this function tries all three approaches and does not get an ID, it assumes that the compound is not in the Wikidata database.

Generate the Splash for a spectrum

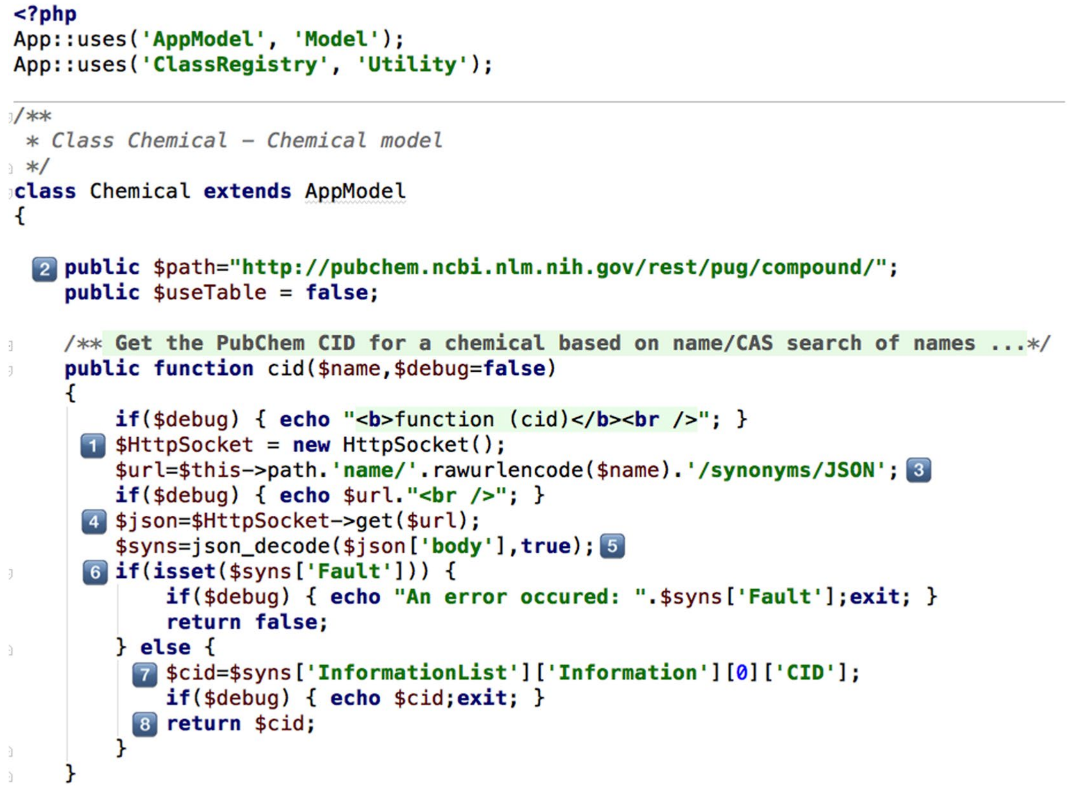
A recent addition to the identifier scene is the 'Spectral Hash' or Splash [39]. This identifier evolved out of work started at the 2015 Metabolomics Hackathon [42] where participants became enthusiastic about unique spectral identifiers similar to the InChIKey. In order to generate a Splash for a spectrum the spectral data is encoded in a JSON object and then sent to the Splash website. The code in Fig. 14 does just that.

Discussion

The OSDB website, as outlined above, provides access to spectral data and its metadata in a standardize way. However, it is important to point out that what can be done with the data is up to the user. This applies to the OSDB website as well as after spectra have been downloaded. For instance, the website does not currently allow for searching the raw data/metadata across all spectra (all of it is in the database but can only be found searching for a complete spectrum).

In order to make this site truly useful the code and data of the project should be made openly available. In this way the user is not limited to the functionality that the original developers envisioned but can develop their own functions/features, enhance the integration of the site, and output the data in new formats for new web or mobile applications. In addition, the openness of the project means it can be used in education as a tool to develop the next generation of cheminformaticians—potentially building their own website from the source code as a course project.

For all these reasons (and many more) the project is available as a free download on GitHub [40]. GitHub is a hosting service for the well-respected Git source code repository system [43]. Git allows multiple developers to write code for one project and centrally coordinate version control, patching, extension and attribution. GitHub does this though a website and adds features like issue tracking, collaborative (discussion based) code review, and team management. Anyone can download the code, work on an enhancement or issue, submit updates, fix issues, and discuss project goals, timelines, and features. The basic site has been built and users can let the developers (that's all of us) know what needs to be added,



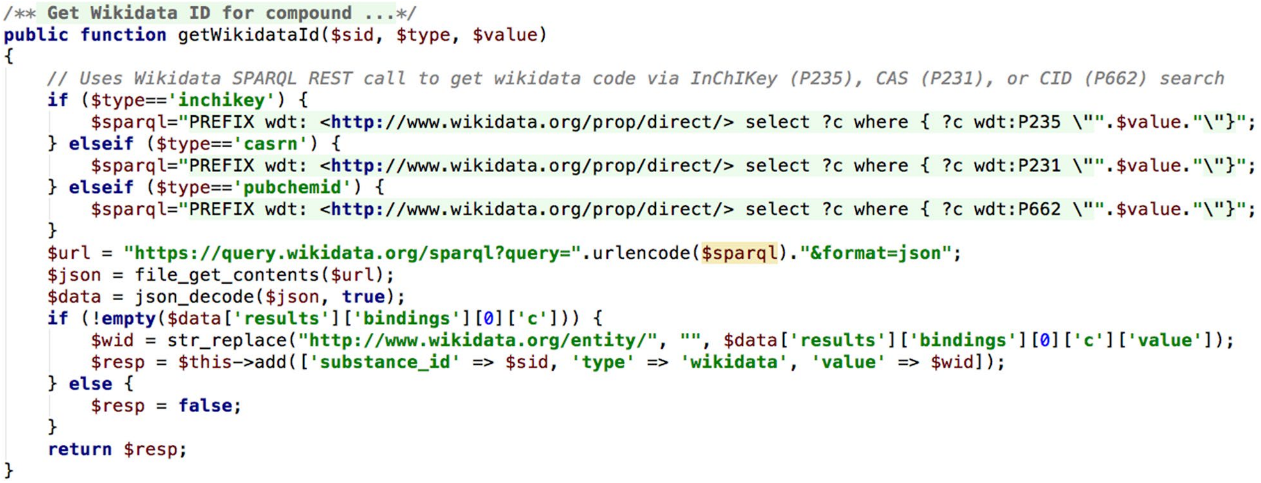
```php
<?php
App::uses('AppModel', 'Model');
App::uses('ClassRegistry', 'Utility');

/**
 * Class Chemical - Chemical model
 */
class Chemical extends AppModel
{

    public $path="http://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/";
    public $useTable = false;

    /** Get the PubChem CID for a chemical based on name/CAS search of names ...*/
    public function cid($name,$debug=false)
    {
        if($debug) { echo "<b>function (cid)</b><br />"; }
        $HttpSocket = new HttpSocket();
        $url=$this->path.'name/'.rawurlencode($name).'/synonyms/JSON';
        if($debug) { echo $url."<br />"; }
        $json=$HttpSocket->get($url);
        $syns=json_decode($json['body'],true);
        if(isset($syns['Fault'])) {
            if($debug) { echo "An error occured: ".$syns['Fault'];exit; }
            return false;
        } else {
            $cid=$syns['InformationList']['Information'][0]['CID'];
            if($debug) { echo $cid;exit; }
            return $cid;
        }
    }
```

Fig. 12 Function to search the PubChem REST API for chemical names and CAS #'s



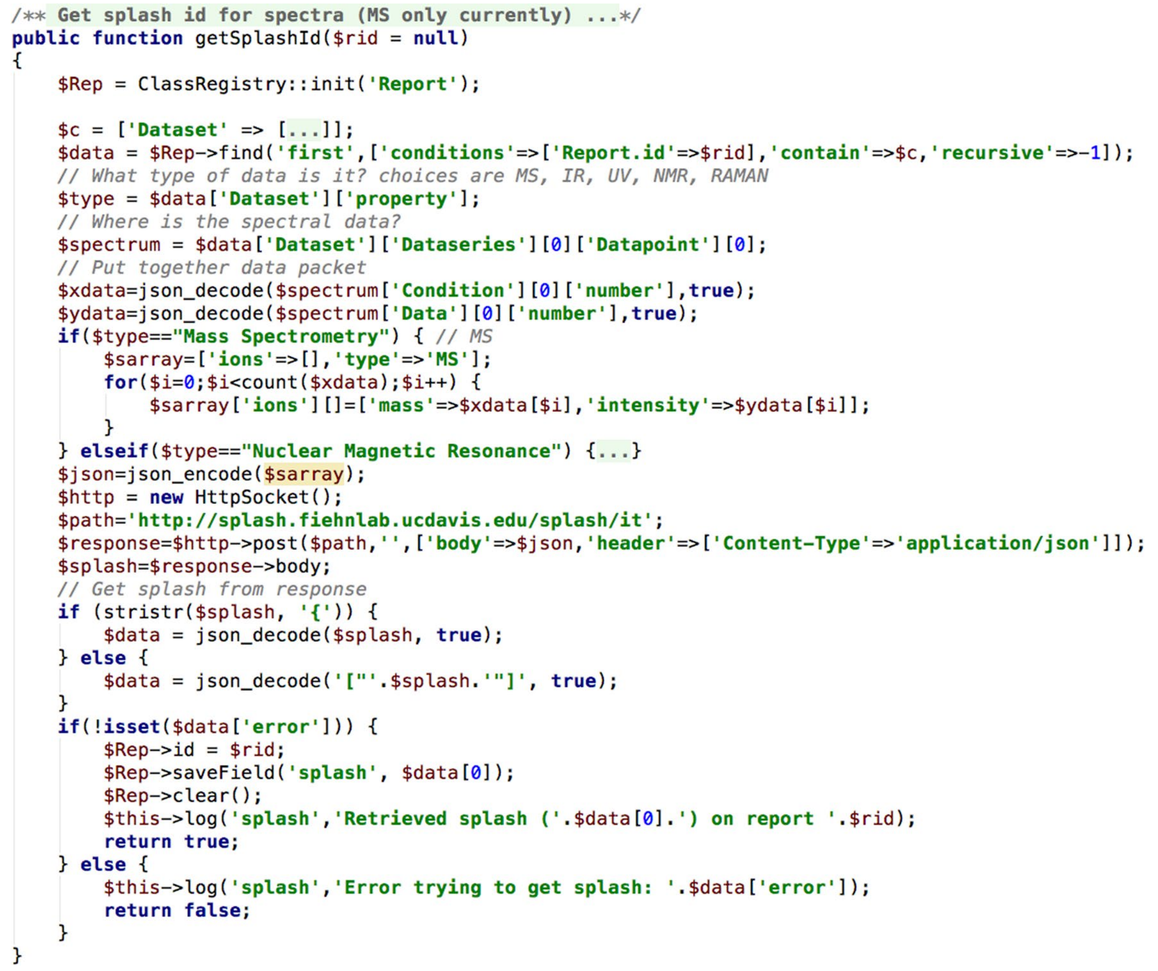
```php
/** Get Wikidata ID for compound ...*/
public function getWikidataId($sid, $type, $value)
{
    // Uses Wikidata SPARQL REST call to get wikidata code via InChIKey (P235), CAS (P231), or CID (P662) search
    if ($type=='inchikey') {
        $sparql="PREFIX wdt: <http://www.wikidata.org/prop/direct/> select ?c where { ?c wdt:P235 \"".$value."\"}";
    } elseif ($type=='casrn') {
        $sparql="PREFIX wdt: <http://www.wikidata.org/prop/direct/> select ?c where { ?c wdt:P231 \"".$value."\"}";
    } elseif ($type=='pubchemid') {
        $sparql="PREFIX wdt: <http://www.wikidata.org/prop/direct/> select ?c where { ?c wdt:P662 \"".$value."\"}";
    }
    $url = "https://query.wikidata.org/sparql?query=".urlencode($sparql)."&format=json";
    $json = file_get_contents($url);
    $data = json_decode($json, true);
    if (!empty($data['results']['bindings'][0]['c'])) {
        $wid = str_replace("http://www.wikidata.org/entity/", "", $data['results']['bindings'][0]['c']['value']);
        $resp = $this->add(['substance_id' => $sid, 'type' => 'wikidata', 'value' => $wid]);
    } else {
        $resp = false;
    }
    return $resp;
}
```

Fig. 13 CakePHP function to search and retrieve the Wikidata ID for a compound

changed or removed, and implement it themselves. Readers are encouraged to check out the 'Projects' page [44] for ideas on additional features/enhancements that you could work on.

Conclusion

This paper describes a new project to support open spectral research data on the web. Anyone can contribute to the content, to the code, to the concept, or to the



```
/** Get splash id for spectra (MS only currently) ...*/
public function getSplashId($rid = null)
{
    $Rep = ClassRegistry::init('Report');

    $c = ['Dataset' => [...]];
    $data = $Rep->find('first',['conditions'=>['Report.id'=>$rid],'contain'=>$c,'recursive'=>-1]);
    // What type of data is it? choices are MS, IR, UV, NMR, RAMAN
    $type = $data['Dataset']['property'];
    // Where is the spectral data?
    $spectrum = $data['Dataset']['Dataseries'][0]['Datapoint'][0];
    // Put together data packet
    $xdata=json_decode($spectrum['Condition'][0]['number'],true);
    $ydata=json_decode($spectrum['Data'][0]['number'],true);
    if($type=="Mass Spectrometry") { // MS
        $sarray=['ions'=>[],'type'=>'MS'];
        for($i=0;$i<count($xdata);$i++) {
            $sarray['ions'][]=['mass'=>$xdata[$i],'intensity'=>$ydata[$i]];
        }
    } elseif($type=="Nuclear Magnetic Resonance") {...}
    $json=json_encode($sarray);
    $http = new HttpSocket();
    $path='http://splash.fiehnlab.ucdavis.edu/splash/it';
    $response=$http->post($path,'',['body'=>$json,'header'=>['Content-Type'=>'application/json']]);
    $splash=$response->body;
    // Get splash from response
    if (stristr($splash, '{')) {
        $data = json_decode($splash, true);
    } else {
        $data = json_decode('["'.$splash.'"]', true);
    }
    if(!isset($data['error'])) {
        $Rep->id = $rid;
        $Rep->saveField('splash', $data[0]);
        $Rep->clear();
        $this->log('splash','Retrieved splash ('.$data[0].') on report '.$rid);
        return true;
    } else {
        $this->log('splash','Error trying to get splash: '.$data['error']);
        return false;
    }
}
```

Fig. 14 CakePHP code to generate a Splash identifier

management/vision. This paper also outlines the components needed to put together such a project and it can be used as a template to build other websites with different functionality and/or different types of chemical data.

The current version of the OSDB is just a starting point. There are many additional features one can envision for the site and it is a hope that the reader has ideas of their own and adds them. Open source code has become a mainstay in the computing world. With the tools, concepts and frameworks outlined in this paper, open source research data will hopefully become a mainstay of the scientific community.

Abbreviations

API: Application Programming Interface; ASDF: ASCII Squeezed Difference Form; CSS: Cascading Style Sheet; GUI: graphical user interface; HTML: Hypertext Markup Language; LAMP: Linux, Apache, MySQL, and PHP; JCAMP-DX: Joint Committee on Atomic and Molecular Physical Data—Data Exchange; JSON: JavaScript Object Notation; JSON-LD: JavaScript Object Notation for Linked Data; LDR: Linked Data Record; MAMP: Mac, Apache, MySQL, and PHP; MVC: model–view–controller; NMR: nuclear magnetic resonance; OOP: object oriented programming; OSDB: Open Spectral Database; OWL: web ontology language; PHP: pre-hypertext processor; REST: representational state transfer; SDM: scientific data model; SMILES: simplified molecular-input line-entry system; SPARQL: SPARQL protocol and RDF query language; SQL: structured query language; URI: uniform resource identifier; WAMP: Windows, Apache, MySQL, and PHP; XML: extensible markup language.

Acknowledgements

Thanks to Tony Williams for encouraging me to start this project. Thanks to J. C. Bradley for pioneering open science and paving the way for projects like this to be conceptualized.

Competing interests

The author declares that he has no competing interests.

Availability of data and materials

All data associated with this project is available at https://github.com/stuchalk/OSDB.

Received: 14 March 2016 Accepted: 4 October 2016
Published online: 14 October 2016

References

1. NIST Materials Measurement Laboratory (2016) NIST chemistry WebBook. National Institute for Standards and Technology, Gaithersburg. http://webbook.nist.gov/. Accessed 19 July 2016
2. Williams T, Tkachenko V (2016) ChemSpider. Royal Society of Chemistry, Cambridge. http://www.chemspider.com/. Accessed 19 July 2016
3. Lancashire R (2016) JCAMP-DX sample files. The University of the West Indies, Mona. http://wwwchem.uwimona.edu.jm/spectra/index.html. Accessed 19 July 2016
4. IUPAC (2016) IUPAC subcommittee on electronic data standards. http://jcamp-dx.org/. Accessed 1 Mar 2016
5. Davies AN, Lampen P (1993) JCAMP-DX for NMR. Appl Spectrosc. doi:10.1366/0003702934067874
6. Grasselli JG (1991) JCAMP-DX, a standard format for exchange of infrared-spectra in computer readable form. Pure Appl Chem. doi:10.1351/pac199163121781
7. Lampen P, Hillig H, Davies AN, Linscheid M (1994) JCAMP-DX for mass-spectrometry. Appl Spectrosc. doi:10.1366/0003702944027840
8. Chalk S (2016) The Open Spectral Database. University of North Florida, Jacksonville. http://osdb.info/. Accessed 1 Mar 2016
9. Sporny M, Longley D, Kellogg G, Lanthaler M, Lindström N (2016) JSON-LD 1.0—a JSON-based Serialization for Linked Data. The World Wide Web Consortium. http://www.w3.org/TR/json-ld/. Accessed 1 Mar 2016
10. Chalk S (2016) SciData—a scientific data model. University of North Florida, Jacksonville. http://stuchalk.github.io/scidata/. Accessed 1 Mar 2016
11. Fielding RT, Taylor RN (2002) Principled design of the modern Web architecture. ACM Trans Internet Technol 2(2):115–150. doi:10.1145/514183.514185
12. Mann A (2014) What's an API? A beginner's guide to the application programming interface. http://www.slideshare.net/CAinc/whats-an-api-a-beginners-guide-to-the-application-programming-interface. Accessed 23 June 2016
13. ASF (2016) The Apache HTTP server project. The Apache Software Foundation (ASF), Forest Hill. http://httpd.apache.org/. Accessed 1 Mar 2016
14. Oracle (2016) MySQL open-source database oracle corporation. http://www.mysql.com/. Accessed 1 Mar 2016
15. The PHP Group (2016) PHP: hypertext preprocessor. The PHP Group. http://php.net/. Accessed 1 Mar 2016
16. JetBrains (2016) PHPStorm. PHP IDE. https://www.jetbrains.com/phpstorm/. Accessed 1 Mar 2016
17. CSF (2016) CakePHP: the rapid PHP development framework. Cake Software Foundation (CSF). http://cakephp.org/. Accessed 1 Mar 2016
18. Fowler M (2006) GUI architectures: model–view–controller. ModelViewController. http://martinfowler.com/eaaDev/uiArchs.html. Accessed 23 June 2016
19. Oracle (2015) Lesson: object-oriented programming concepts. https://docs.oracle.com/javase/tutorial/java/concepts/. Accessed 23 June 2016
20. Chalk S (2016) The Open Spectral Database—system index. University of North Florida, Jacksonville. http://osdb.info/systems. Accessed 1 Mar 2016
21. Baumbach JI, Davies AN, Lampen P, Schmidt H (2001) JCAMP-DX. A standard format for the exchange of ion mobility spectrometry data (IUPAC recommendations 2001). Pure Appl Chem. doi:10.1351/pac200173111765
22. Cammack R, Fann Y, Lancashire RJ, Maher JP, McIntyre PS, Morse R (2006) JCAMP-DX for electron magnetic resonance (EMR). Pure Appl Chem. doi:10.1351/pac200678030613
23. Woollett B, Klose D, Cammack R, Janes RW, Wallace BA (2012) JCAMP-DX for circular dichroism spectra and metadata (IUPAC Recommendations 2012). Pure Appl Chem. doi:10.1351/PAC-REC-12-02-03
24. Chalk S (2016) SciData: a data model and ontology for semantic representation of scientific data. J Cheminform
25. BCT (2016) Bootstrap Bootstrap Core Team. http://getbootstrap.com/. Accessed 1 Mar 2016
26. Çelik T, Lilley C, Baron LD, Pemberton S, Pettit B (2016) Cascading Style Sheets working group. The World Wide Web Consortium. https://www.w3.org/TR/css3-color/. Accessed 1 Mar 2016
27. Hickson I, Berjon R, Faulkner S, Leithead T, Doyle Navara E, O'Connor E, Pfeiffer S (2016) HTML5: a vocabulary and associated API's for HTML and XHTML The World Wide Web Consortium. http://www.w3.org/TR/html5/. Accessed 1 Mar 2016
28. jQuery Foundation (2016) jQuery JavaScript Library. http://jquery.com/. Accessed 1 Mar 2016
29. Chalk S (2016) The Open Spectral Database—spectra index. University of North Florida, Jacksonville. http://osdb.info/spectra. Accessed 1 Mar 2016
30. Chalk S (2016) The Open Spectral Database—compound index. University of North Florida, Jacksonville. http://osdb.info/compounds. Accessed 1 Mar 2016
31. Chalk S (2016) The Open Spectral Database—analytical technique index. University of North Florida, Jacksonville. http://osdb.info/techniques. Accessed 1 Mar 2016
32. Chalk S (2016) The Open Spectral Database—collection index. University of North Florida, Jacksonville. http://osdb.info/collections. Accessed 1 Mar 2016
33. Hanson R (2016) JSmol: JavaScript-based molecular viewer from Jmol sourceforge. https://sourceforge.net/projects/jsmol/. Accessed 19 July 2016
34. NLM (2016) PubChem Power User Gateway (PUG) REST interface documentation. https://pubchem.ncbi.nlm.nih.gov/pug_rest/PUG_REST.html. Accessed 1 Mar 2016
35. NLM (2016) PubChem. National Institutes of Health, Bethesda. https://pubchem.ncbi.nlm.nih.gov/. Accessed 19 July 2016
36. WikiMedia (2016) Wikidata. https://www.wikidata.org/. Accessed 1 Mar 2016
37. Chalk S (2016) The Open Spectral Database—API. University of North Florida, Jacksonville. http://osdb.info/api. Accessed 1 Mar 2016
38. SmartBear (2016) Swagger API framework open API initiative (OAI). http://swagger.io/. Accessed 19 July 2016
39. UC Davis (2016) Splash—the spectral hash identifier. http://splash.fiehnlab.ucdavis.edu/. Accessed 1 Mar 2016
40. Chalk S (2016) Open Spectral Database Github repository. GitHub Inc, San Francisco. https://github.com/stuchalk/OSDB/. Accessed 1 Mar 2016
41. Harris S, Seaborne A (2016) SPARQL query language for RDF. The World Wide Web Consortium. https://www.w3.org/TR/sparql11-query/. Accessed 19 July 2016
42. Metabolomics Society (2016) Metabolomics Hackathon Metabolomics Society. http://metabolomics2015.org/index.php/program/hackathon. Accessed 1 Mar 2016
43. SFC (2016) Git distributed version control system. Software Freedom Conservancy. https://git-scm.com/. Accessed 1 Mar 2016
44. Chalk S (2016) The Open Spectral Database—projects. University of North Florida, Jacksonville. http://osdb.info/pages/projects. Accessed 1 Mar 2016