Published online 30 April 2015 *Nucleic Acids Research, 2015, Vol. 43, Web Server issue* ***W605–W611***
doi: 10.1093/nar/gkv396

PUG-SOAP and PUG-REST: web services for programmatic access to chemical information in PubChem

Sunghwan Kim†, Paul A. Thiessen†, Evan E. Bolton* and Stephen H. Bryant

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Department of Health and Human Services, Bethesda, MD 20894, USA

Received February 6, 2015; Revised April 10, 2015; Accepted April 15, 2015

ABSTRACT

PubChem (http://pubchem.ncbi.nlm.nih.gov) is a public repository for information on chemical substances and their biological activities, developed and maintained by the US National Institutes of Health (NIH). PubChem contains more than 180 million depositor-provided chemical substance descriptions, 60 million unique chemical structures and 225 million bioactivity assay results, covering more than 9000 unique protein target sequences. As an information resource for the chemical biology research community, it routinely receives more than 1 million requests per day from an estimated more than 1 million unique users per month. Programmatic access to this vast amount of data is provided by several different systems, including the US National Center for Biotechnology Information (NCBI)'s Entrez Utilities (E-Utilities or E-Utils) and the PubChem Power User Gateway (PUG)—a common gateway interface (CGI) that exchanges data through eXtended Markup Language (XML). Further simplifying programmatic access, PubChem provides two additional general purpose web services: PUG-SOAP, which uses the simple object access protocol (SOAP) and PUG-REST, which is a Representational State Transfer (REST)-style interface. These interfaces can be harnessed in combination to access the data contained in PubChem, which is integrated with the more than thirty databases available within the NCBI Entrez system.

INTRODUCTION

PubChem (http://pubchem.ncbi.nlm.nih.gov) (1–4) is a public repository for information on chemical substances and their biological activities, developed and maintained by the US National Center for Biotechnology Information (NCBI), as a part of the US National Library of Medicine (NLM), an institute within the US National Institutes of Health (NIH). Initially launched in 2004, PubChem is a resource for researchers in areas such as cheminformatics, chemical biology and medicinal chemistry. It routinely receives millions of requests from many tens of thousands users per day.

As depicted in Figure 1, PubChem organizes its data into three primary databases: Substance, Compound and BioAssay (1). The Substance database (http://www.ncbi.nlm.nih.gov/pcsubstance—accession SID) contains chemical substance descriptions deposited by individual contributors. Different data sources may provide information on the same molecule, hence the same chemical structure may appear multiple times as different records in the Substance database. To provide a non-redundant view, chemical structures in the Substance database are normalized through a process called 'standardization' (1) and the unique chemical structures are identified and stored in the Compound database (http://www.ncbi.nlm.nih.gov/pccompound—accession CID). Descriptions of biological experiments on chemical substances are stored in the BioAssay database (http://www.ncbi.nlm.nih.gov/pcassay—accession AID).

Currently PubChem contains more than 180 million depositor-provided substance descriptions, 60 million unique chemical structures and 225 million biological test results from 1 million assays, covering more than 9000 unique protein target sequences (as of January 2015). Programmatic access to this vast amount of data by the chemical biology and biomedical communities presents new opportunities for data-driven research in a 'big data' era.

NCBI's Entrez Utilities (also called E-Utilities or E-Utils) are used for programmatic access to information contained in the Entrez system (5,6). While suited for accessing text- or numeric-fielded data, E-Utilities do not have the ability to deal with the more complex types of data specific to PubChem, such as chemical structures and tabular bioactivity data. Thus, PubChem provides additional program-

*To whom correspondence should be addressed. Tel: +1 301 451 1811; Fax: +1 301 480 4559; Email: bolton@ncbi.nlm.nih.gov
†These authors contributed equally to the paper as first authors.

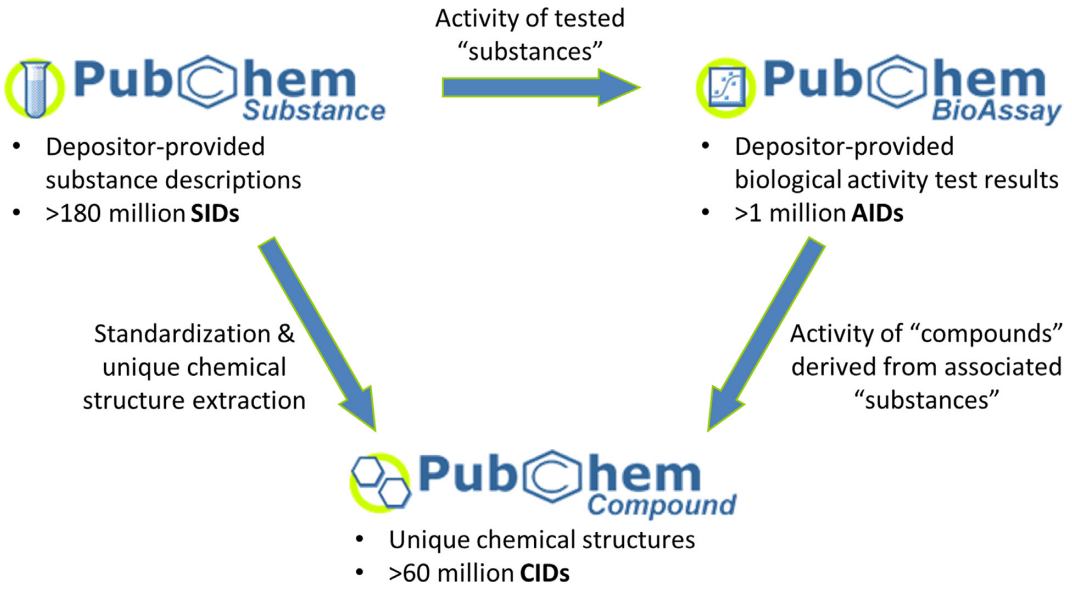

Figure 1. Three primary databases in PubChem and the data flow between them. SID, CID and AID are the database identifiers for the Substance, Compound and BioAssay databases, respectively.

matic access routes specialized for PubChem data and analysis services, one of which is called the Power User Gateway (PUG). While suitable for low-level programmatic access to PubChem, PUG exchanges data through a complex eXtended Markup Language (XML—http://www.w3.org/XML) schema describing aspects of available query and analysis data services. For the sake of user-friendliness and for integration with a variety of third party tools, PubChem provides two easier-to-use web service access methods: PUG-SOAP, which uses the simple object access protocol (SOAP) (http://www.w3.org/TR/soap) and PUG-REST, which is a Representational State Transfer (REST)-style interface (7,8).

This article provides an overview of PUG, PUG-SOAP and PUG-REST programmatic interfaces. More detailed information on these services can be found in documents available on the PubChem website, as summarized in Table 1. These web pages provide a variety of examples to help users learn to how to access these services.

POWER USER GATEWAY (PUG)

PUG provides programmatic access to PubChem services via a single common gateway interface (CGI), called 'pug.cgi', available at http://pubchem.ncbi.nlm.nih.gov/pug/pug.cgi. This CGI (hereafter referred to simply as PUG) is the central gateway to several PubChem services. Instead of taking any Uniform Resource Locator (URL) arguments, PUG exchanges data through XML. That is, to perform any request to PUG, one needs to formulate an input in XML and then send it to PUG via a Hypertext Transfer Protocol (HTTP) POST. The CGI will interpret the incoming request, initiate the appropriate action and then return results also in XML format.

Examples of PubChem services enabled for use by PUG are:

- PubChem Substance/Compound download (http://pubchem.ncbi.nlm.nih.gov/pc_fetch/pc_fetch.cgi)
- PubChem BioAssay data download (http://pubchem.ncbi.nlm.nih.gov/assay/assay.cgi)
- PubChem structure standardization (http://pubchem.ncbi.nlm.nih.gov/standardize/standardize.cgi)
- PubChem chemical structure search (http://pubchem.ncbi.nlm.nih.gov/search/search.cgi)
- PubChem score matrix (http://pubchem.ncbi.nlm.nih.gov/score_matrix/score_matrix.cgi)
- PubChem Identifier Exchange Service (http://pubchem.ncbi.nlm.nih.gov/idexchange/idexchange.cgi)

Each PUG supported service has its own input and output. All XML used by PUG is described in the data type definition (DTD) file [http://pubchem.ncbi.nlm.nih.gov/pug/pug.dtd] and, alternatively, in the XML Schema definition (XSD) file [http://pubchem.ncbi.nlm.nih.gov/pug/pug.xsd]. PUG XML can also be used to import and export queries within supported PubChem query and analysis service web pages, which can help programmatic users learn how to compose valid PUG XML requests and to verify that a formulated PUG XML request does what is intended. Note that most PUG requests are queued, allowing users to submit a number of long-running tasks. The initial PUG response contains a 64-bit identifier for a requested task, and must be used in any further communication with PUG concerning that task. Upon request, PUG will check the status of a task given its identifier, returning the results of the task if completed or the status of the task if not completed.

The PubChem PUG Help page (http://pubchem.ncbi.nlm.nih.gov/pug/pughelp.html) provides detailed information on PUG, including how to formulate XML inputs for each PubChem-enabled service. It contains examples of simple workflows that illustrate the use of PUG and explains how to enhance data analysis by using PUG in conjunction with E-Utilities (via Entrez history).

Table 1. Documents that provide detailed information on the Power User Gateway (PUG), PUG-SOAP and PUG-REST services

- **PUG Help (http://pubchem.ncbi.nlm.nih.gov/pug/pughelp.html)**
  provides a detailed description on PUG, with examples for each PUG-enabled service.
- **PUG-SOAP Help (http://pubchem.ncbi.nlm.nih.gov/pug_soap/pug_soap_help.html)**
  provides an introduction to PUG-SOAP, with a general description of functions available through PUG-SOAP.
- **PUG-SOAP Client Help (http://pubchem.ncbi.nlm.nih.gov/pug_soap/client_help.html)**
  provides ready-to-run PUG-SOAP examples for SOAP-aware applications (Taverna and Pipeline Pilot) and programming languages (Java, Python, Perl, C#/.NET, and Visual Basic .Net).
- **PUG-SOAP Web Service Reference (http://pubchem.ncbi.nlm.nih.gov/pug_soap/PUG_SOAP.html)**
  provides more detailed description on PUG-SOAP functions, including the SOAP messages, XML schema types, enumerations, etc.
- **PUG-REST Help (http://pubchem.ncbi.nlm.nih.gov/pug_rest/PUG_REST.html)**
  A PUG-REST specification document that details the syntax of the HTTP requests and the available functions.
- **PUG-REST Tutorial (http://pubchem.ncbi.nlm.nih.gov/pug_rest/PUG_REST_Tutorial.html)**
  A tutorial-style document that explains how the PUG-REST service works and how to construct the URLs for PUG-REST requests, with a variety of usage cases as illustrations.

PUBCHEM WEB SERVICES

PUG-SOAP: a SOAP-based interface

While PUG exchanges data through a relatively complex XML schema, PUG-SOAP contains much of the same functionality broken down into simpler functions, as defined via the web service definition language (WSDL; http://www.w3.org/TR/wsdl), using SOAP formatted message envelopes for information exchange. The WSDL for PUG-SOAP can be found at http://pubchem.ncbi.nlm.nih.gov/pug_soap/pug_soap.cgi?wsdl. This WSDL/SOAP layer is most suitable for SOAP-aware GUI workflow applications (e.g. Taverna and Pipeline Pilot) and programming/scripting languages (e.g. C, C++, C#, .NET, Perl, Python and Java).

In PUG-SOAP, complex data objects like chemical structures and lists of PubChem database identifiers are abstracted into so-called 'keys.' These keys are simple strings that can easily be exchanged between the PUG-SOAP server and client applications. Their use reduces the need to send and receive intermediate results and allows one to readily chain queries between different PubChem services (by using an output key as an input key), thus reducing bandwidth requirements.

There are four types of keys: structure keys, list keys, assay keys and download keys.

- **Structure key**: contains a single chemical structure. Currently, multiple structures are not supported. When a chemical structure in a supported format [such as SMILES (9–11), InChI (12–14), ASN.1 (text or binary), XML and SDF (15)] is provided as an initial input, PUG-SOAP returns a structure key in which the input chemical structure is stored. This key can be used in subsequent functions that take a chemical structure as an input, such as identity search, similarity search, substructure/superstructure search and chemical structure standardization.
- **List key**: holds a set of identifiers (such as PubChem SIDs, CIDs or AIDs). While it is possible for the users to directly input a set of IDs into a list key, list keys are often returned as an output and can be reused as an input. For example, a similarity search (which takes a structure key as input) returns as output a CID list key that contains compounds similar to the query abstracted in the structure key input. This list key returned in this example may then be used to retrieve the list of CIDs, to download structures for the CIDs, to put the CID result set into Entrez, to limit the search space of subsequent CID-based queries and so on.
- **Assay key**: specifies a set of rows and columns from an assay table. These can be used in a subsequent request, such as preparing a download file containing the bioassay data.
- **Download key**: provides an URL (usually FTP) from which the desired records may be downloaded.

The functions available through PUG-SOAP are listed in Supplementary Table S1, with their inputs, outputs and brief descriptions. More details on these functions can be found in the PUG-SOAP web service reference in HTML and PDF formats (http://pubchem.ncbi.nlm.

nih.gov/pug_soap/PUG_SOAP.html and http://pubchem.ncbi.nlm.nih.gov/pug_soap/PUG_SOAP.pdf, respectively). These functions can be classified into three categories: input, processing and output functions. Input functions are used to specify structures and ID lists for further operations. Processing functions operate on the input list, and may be an analysis, search or some other supported operation. Output functions are used to retrieve the results. By design, the input functions all begin with 'Input' and the output functions are begins with 'Get'. Processing functions may have any name.

Processing functions may take a while to complete, so most requests are queued asynchronously. An appropriate key is returned when making a processing function request (e.g. structure keys, assay keys, list keys or download keys). A status check function allows you to use this key to poll the server to determine when the request finishes. Upon completion, the same key can then be used to retrieve results, or the key may be used as an input to another appropriate processing function.

PUG-REST: a REST-style interface

PUG-REST, a REST-style web service access layer to PubChem, provides a simplified access route to PubChem without the overhead of XML or SOAP envelopes that are required with PUG and PUG-SOAP. PUG-REST also provides convenient access to information on PubChem records not possible with the other PUG services. It is intended to handle short, synchronous requests—that is, the result is given in a single call that may last at most 30 s (the default timeout on PubChem servers), without any intermediate step to poll whether that request has completed. A detailed description of PUG-REST is given in the specification document, available at http://pubchem.ncbi.nlm.nih.gov/pug_rest/PUG_REST.html. A tutorial-style document on PUG-REST is also available at http://pubchem.ncbi.nlm.nih.gov/pug_rest/PUG_REST_Tutorial.html.

Concepts and Syntax of PUG-REST requests

Figure 2 shows the conceptual framework of the PUG-REST services, the construction of an URL path for PUG-REST requests and several examples. Conceptually, a PUG-REST request has three parts: (i) the PUG-REST server takes identifiers (directly or indirectly by means of an implicit query) from PubChem's three primary databases as an input (i.e., SID for Substance, CID for Compound and AID for BioAssay), (ii) it performs a requested operation using the identifiers and (iii) it returns the results to the requestor in a desired output type and format. The details of a PUG-REST request are directly encoded in an URL consisting of a prologue (which is common to all requests) and the three-part request: input, defining identifiers of interest; operation, specifying what to do with the identifiers; and output, defining the desired output format. Some PUG-REST URL requests take one or more parameters that require appending a question mark '?' at the end of the request URL and then providing a list of '&' separated option name and option value pairs like those shown in examples 1, 2, 8 and 9 provided in Figure 2. It is noteworthy that the three parts of the PUG-REST request are, for the most part, independent of each other, allowing for many possible actions that can be encoded in a single PUG-REST URL.

There can be conflicts with URL syntax that prevent some PUG-REST requests from being directly encoded in an URL. For example, any chemical name, InChI or SMILES string that has a '/' (forward slash) or other special characters reserved in the URL syntax cannot be provided directly in an URL. There are also length restrictions that preclude long lists of identifiers from being part of a URL. To work around these limitations, many parameters to PUG-REST may be provided via HTTP POST; more details on the use of HTTP POST can be found in the PUG-REST tutorial.

The intent of PUG-REST is to provide a flexible architecture that handles many types of operations useful to PubChem users. The allowed PUG-REST keywords are summarized in Supplementary Figure S1 (see Supplementary Data). Inputs to PUG-REST can be specified in many different ways, for example, using identifiers (CID, SID and AID), chemical names and chemical structures representations (SMILES, SDF, InChI and InChI key) and associated cross-references (summarized in Supplementary Table S2). The input assay identifiers can also be specified by the assay target (protein or gene) as well as assay type (screening, confirmatory, summary, dose-response, cell-based, RNAi and so on). The output formats accepted in PUG-REST include XML, ASNT (NCBI's text version of ASN.1), ASNB (standard binary ASN.1), JSON(P), SDF, CSV, PNG and TXT, as summarized in Supplementary Table S3. Note that not all formats are applicable to the results of all operations. In the case of XML formatted PUG-REST output, there is an XML schema found at http://pubchem.ncbi.nlm.nih.gov/pug_rest/pug_rest.xsd.

URL examples for PUG-REST requests

Figure 2 provides a number of examples demonstrating valid PUG-REST URLs. The two URLs in Example 1 retrieve entire records for CIDs 2244 and 1983, including the computed three-dimensional (3-D) structure (as specified by the 'record_type = 3d' option after the '?' mark). The full record retrieval is the default action if no operation is specified. Therefore, the two requests in Example 1 give identical results. If PNG is specified as the output format for the full record retrieval (as in Example 2), the image of the structure of the compound will be returned. In PUG-REST, images are considered a flavor of full-record output, because they depict the structure as a whole. It is also noteworthy that the 'record_type' option can be omitted in Example 2, because the 2-dimensional image of the molecule is retrieved by default.

It is also possible to retrieve specific fields from the input records (as shown in Examples 3–5), such as compound properties (see Supplementary Table S4 for a complete list) or cross-references associated with PubChem records (see Supplementary Table S2 for a complete list). One can get assays that target a particular gene or protein (Example 6) or specify the assays of interest by their AIDs (Examples 7 and 8). Through PUG-REST, one can access a variety of information from the BioAssay database, such as assay de-

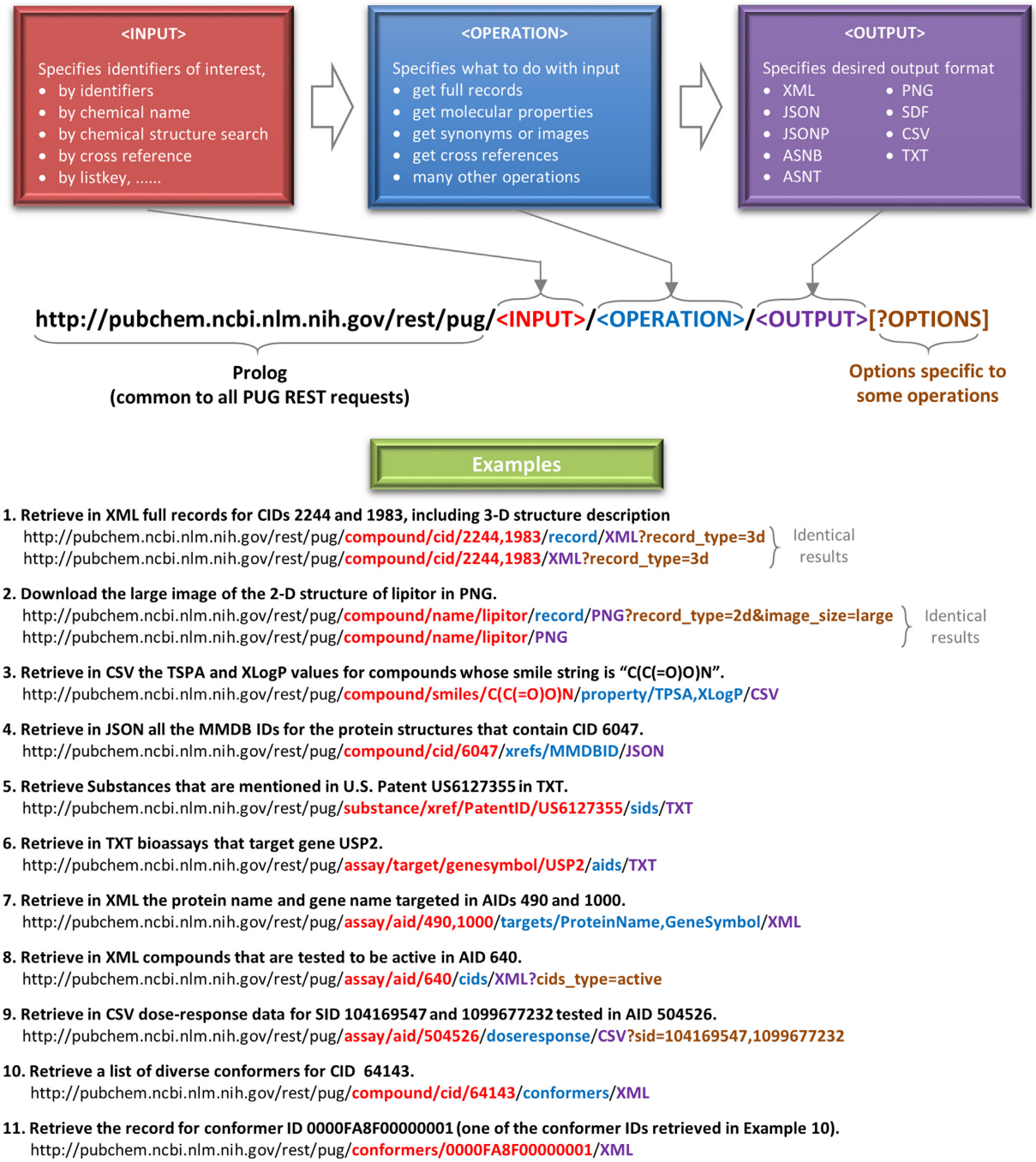

Figure 2. Construction of URLs for PUG-REST requests.

scriptions, assay targets (Example 7), bioactivities of tested compounds, dose-response data (Example 9) and so on.

The 3-D structures of compounds (computed) or substances (most experimentally determined) are available via PUG-REST. Example 10 illustrates how to get conformer IDs for a compound. These conformer IDs can be used to download the 3-D structure of the conformer (Example 11).

Asynchronous operations in PUG-REST

PubChem has a standard time limit of 30 s per web service request and all examples in Figure 2 will finish rapidly. However, some requests possible in PUG-REST can take longer than the 30 s limit, for various reasons, and will return an error message (error status codes are summarized in Supplementary Table S5). To work around certain slower operations, PUG-REST can use an 'asynchronous' approach, where a so-called 'list key' is returned as a response to the request—much the same as the list keys in PUG-SOAP

Figure 3. Asynchronous substructure search using the PUG-REST interface. The list key returned from the initial request will be used both to check the status of the search and to retrieve the final results.

(note that PUG-REST and PUG-SOAP list keys are interchangeable). After making a request that returns a list key, the caller will need to check periodically whether the operation has finished, and, when complete, retrieve the results. Examples of asynchronous operations are: chemical structure search including identity search, substructure search, superstructure search, similarity search and molecular formula search. Figure 3 illustrates how to submit a PUG-REST request for substructure search and retrieve the results using a list key.

However, it should be noted that recent re-engineering of many of PubChem's search services has made them fast enough to work synchronously. These operations have a 'fast' prefix on them in PUG-REST, in order to preserve backwards compatibility of the asynchronous variants as described above. For example, the 'fastsubstructure' input can be used to retrieve a list of CIDs from a chemical substructure search; this input can then be used with any other PUG-REST operation on CIDs, in a single request.

Beyond structure search, any operation that results in a list of SIDs, CIDs or AIDs can be stored in a list key on the server side, and can be retrieved in part or whole by subsequent requests. List keys are very helpful to reduce bandwidth requirements, especially when a long list of identifiers results from an operation request. These list keys can also be very helpful when chaining requests, using the output of one request as the input of another, without having to send the whole list back and forth. Note that a list key is not permanent; it expires after a period of time (8 h at the time of writing).

Request volume limitations

All PubChem web pages (or requests to NCBI in general) have a policy that users should throttle their web page requests, which includes web-based programmatic services. Users should limit their web-requests to no more than three per second. Violators of usage policies may result in the user being temporarily blocked from accessing PubChem (or NCBI) resources.

DISCUSSION

PubChem PUG is a pure XML-based interface that requires some knowledge about a relatively complex XML schema specific to each PUG-enabled service. PUG-SOAP provides an easier programmatic access route to much of the same functionality as PUG. SOAP web service interfaces are readily available for most scripting and programming languages.

As technology changes, PubChem continues to transform its services to help meet the needs of the community. PUG-REST represents the latest generation of PubChem programmatic services. It is the simplest to use and learn as it does not require the overhead of XML and SOAP envelopes. Information necessary to make a PUG-REST request can be encoded into a single URL that can be written by hand without programming expertise. These simplified URLs can be readily incorporated into web pages or into complex work flows. New features continue to be added to PUG-REST as needs arise, and as the PubChem team receives feedback from its users; see the 'change log' at the

top of the PUG-REST specification document for the latest developments.

As mentioned previously, PUG-REST was intended to handle short, synchronous requests. However, because certain types of tasks (such as chemical structure search including identity search, similarity search, substructure/superstructure search and molecular formula search) took inherently much longer than other tasks, it was not reasonable to handle them as synchronous requests. Considering that these tasks were very commonly requested through PUG and PUG-SOAP, it was necessary to make them available in PUG-REST, even if they were not suitable to treat synchronously. Therefore, it was an inevitable choice to provide them as asynchronous operations. Since then, we have made significant progress in improving the speed of chemical similarity searches, making them fast enough to be implemented as synchronous operations. These operations have a 'fast' prefix on them in PUG-REST, in order to preserve backwards compatibility of the asynchronous variants.

PUG-REST supports certain methods for accessing information from PubChem that are not available through PUG and PUG-SOAP, and *vice versa*. Importantly, PUG-REST provides the flexibility to customize requests by combining into a single request the input, operation and output parts that are (mostly) independent. This makes PUG-REST an ideal option in many use cases. However, PUG-REST does not support access to all of the capabilities of PUG. For example, many PubChem Bioactivity analysis services are PUG-aware but do not have a PUG-REST counterpart. While PUG XML may be complex, PUG-aware PubChem web pages allow you to save a request in PUG XML format. These can then be modified or used to repeat a query, interactively or programmatically.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENT

Special thanks to the PubChem team and to the hundreds of data contributors for making their data openly accessible within PubChem.

FUNDING

Intramural Research Program of the National Library of Medicine, National Institutes of Health (in part). Funding for open access charge: Intramural Research Program of the National Library of Medicine, National Institutes of Health.

Conflict of interest statement. None declared.

REFERENCES

1. Bolton,E.E., Wang,Y., Thiessen,P.A. and Bryant,S.H. (2008) PubChem: integrated platform of small molecules and biological activities. *Annu. Rep. Comput. Chem.*, **4**, 217–241.
2. Wang,Y., Xiao,J., Suzek,T.O., Zhang,J., Wang,J. and Bryant,S.H. (2009) PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res.*, **37**, W623–W633.
3. NCBI Resource Coordinators. (2015) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **43**, D6–D17.
4. Wang,Y., Suzek,T., Zhang,J., Wang,J., He,S., Cheng,T., Shoemaker,B.A., Gindulyte,A. and Bryant,S.H. (2014) PubChem BioAssay: 2014 update. *Nucleic Acids Res.*, **42**, D1075–D1082.
5. McEntyre,J. (1998) Linking up with Entrez. *Trends Genet.*, **14**, 39–40.
6. Schuler,G.D., Epstein,J.A., Ohkawa,H. and Kans,J.A. (1996) Entrez: molecular biology database and retrieval system. *Methods Enzymol.*, **266**, 141–162.
7. Fielding,R.T. (2000) Representational state transfer (REST). In: *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.
8. Fielding,R.T. and Taylor,R.N. (2000) Principled design of the modern Web architecture. In: *Proceedings of the 22nd International Conference on Software Engineering*, Limerick, pp. 407–416.
9. Weininger,D. (1988) SMILES, a chemical language and information-system .1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, **28**, 31–36.
10. Weininger,D., Weininger,A. and Weininger,J.L. (1989) SMILES .2. Algorithm for generation of unique smiles notation. *J. Chem. Inf. Comput. Sci.*, **29**, 97–101.
11. Weininger,D. (1990) SMILES .3. Depict—graphical depiction of chemical structures. *J. Chem. Inf. Comput. Sci.*, **30**, 237–243.
12. McNaught,A. (2006) The IUPAC International Chemical Identifier: InChl — a new standard for molecular informatics. *Chem. Int.*, **28**, 12–14.
13. Frey,J.G. (2006) Using InChI. *Chem. Int.*, **28**, 14–15.
14. Heller,S., McNaught,A., Stein,S., Tchekhovskoi,D. and Pletnev,I. (2013) InChI—the worldwide chemical structure identifier standard. *J. Cheminform.*, **5**, 7.
15. Dalby,A., Nourse,J.G., Hounshell,W.D., Gushurst,A.K.I., Grier,D.L., Leland,B.A. and Laufer,J. (1992) Description of several chemical-structure file formats used by computer-programs developed at molecular design limited. *J. Chem. Inf. Comput. Sci.*, **32**, 244–255.