Why data citation is a computational problem

Peter Buneman
University of Edinburgh

Susan Davidson
University of Pennsylvania

James Frew
University of California, Santa Barbara

This is a preprint, as of 2016-06-22.

For published version, see: http://doi.org/10.1145/2893181

1 Why Data Citation?

Citation is essential to traditional scholarship. Citations identify the cited material, help retrieve it, give credit to the creator of the material, date it, and so on. In the context of printed materials, such as books and journals, citation is well understood. However, the world is now digital. Most of our scholarly and scientific resources are held online and many are in some kind of database, i.e. a structured, evolving collection of data. For example, most biological reference works have been replaced by curated databases, and vast amounts of basic scientific data – geospatial, astronomical, molecular, etc. – are now available on-line. There is strong demand [13, 23] that we should accord the same scholarly status to these databases and cite them appropriately, but how can we do this effectively?

Database citation is a challenge because of the structure and evolution of databases. Attributes such as ownership or authorship may change for different parts of the database. Even for a simple collection of files, we may want to find good methods of citing subsets of these files; that is, we want to do better than cite the whole collection or generate a huge number of citations to individual files. While principles and standards have been developed for data citation, they are unlikely to be used unless we can couple the process of extracting information from the database with that of providing a citation for it.

A citation is a collection of "snippets" of information, such as authorship, title, ownership, date, etc that are specified by the database administrators and which may be prescribed by some standard. However, if we expect people to cite digital data, simply providing principles and standards for citation is not enough – we must also *generate* the citations. Even when making conventional citations to the literature, we typically avoid typing in citations. Instead we look for the citation in some database of citations (e.g. the ACM Digital Library[1] or DBLP[2]) and insert it into our document using a reference manager (BibTeX, Mendeley, Zotero, etc.) or by copy-paste. In the context of citing databases, if the citation is not available or if the standard appears complicated, we are almost certain to omit the citation or provide an inaccurate one. In short, *unless we couple the process of generating a citation with the act of extracting the data, the advocacy of data citation will have limited effect.*

How then are we to generate citations for data extracted from a database? Using the term "database" in a broad sense and the term "query" to mean any mechanism used to extract the data – for example, a set of file names, an SQL query, a URL, a special purpose GUI, etc. The problem we then need to solve is simply formulated as follows:

$$\boxed{\text{Given a database } D \text{ and a query } Q\text{, generate an appropriate citation.}}$$

It is often the case that the curators, authors or publishers of a database have good ideas about how their data should be cited. However, it is unlikely that they will know how to associate a citation with some complex SQL query, and even less likely that the user of the data, whose query was generated by some user interface, will understand what is wanted. We need to extract the citation *automatically* from the query Q and the database D, which raises two questions:

- Does the citation depend on both Q and D, or just on the data $Q(D)$ extracted by Q from D?
- If we have appropriate citations for some queries, can we use these to construct citations for other queries?

If the retrieved data is simply a number or an image, we cannot expect to find the citation in the retrieved data. Moreover, even if the query returns nothing, it may be worthy of citation – but what citation is associated with the empty set? We need at least context information; we need both Q and D.

The answer to the second question is important because authors and publishers frequently have ideas on how to cite certain parts of the database, i.e., they can provide citations for certain queries, but they do not know what to do about other queries.

Numerous organizations [16, 12, 6, 2] have advocated data citation and developed principles [4, 8, 15, 13, 12, 7, 3, 2] that refine and standardize the notion [1, 4, 9, 8, 18, 3]. The purpose of these standards is mostly to prescribe the information in a citation – the snippets – and also to define its structure.

[1] `http://dl.acm.org/`

[2] `http://dblp.uni-trier.de/`

A major, but not the only, purpose of a citation is to identify the cited material, and citation is often linked to persistent identifiers such as DOIs[3], ARKs[4], or URIs[5]. These identifiers, while they may have certain fixed properties, do not guarantee *fixity* – that the cited material remains unchanged. Beyond observing that citations should reference the appropriate version, we do not address fixity in this paper; nor do we address the closely related topic of provenance which, in addition to archiving, involves a record of the whole process of data extraction. For a discussion of these issues and a prototype system that combines citation and provenance, see work by Pröll and Rauber [21, 22].

In this paper, we propose a general approach to citation generation (Section 3), and illustrate it in the context of two very different scientific databases (Section 2).

2 Sample Scientific Datasets

To illustrate the computational issues of data citation, we describe two scientific databases that differ widely both in their structure and in how they should be cited.

2.1 GtoPdb

The IUPHAR/BPS Guide to Pharmacology (GtoPdb) [20][6] is a relational database that contains expertly curated information about drugs in clinical use and some experimental drugs, together with information on the cellular targets of the drugs and their mechanisms of action in the body. The resource is particularly useful to researchers who hypothesize that a particular cellular mechanism is involved in a physiological process of interest, and want to find tools (drugs) to impose a specific activation level on the pathway to test their hypotheses.

Users view information through a hierarchy of web pages. The top level divides information by "families" of drug targets that reflect typical pharmacological thinking; lower levels divide the families hierarchically into sub-families and so on down to individual drug targets and drugs. At the lowest level are expert-created overviews and, for some entries, pages containing details of chemical and genetic structures and properties. Despite its underlying relational implementation, GtoPdb can therefore be thought of as a structured hierarchy.

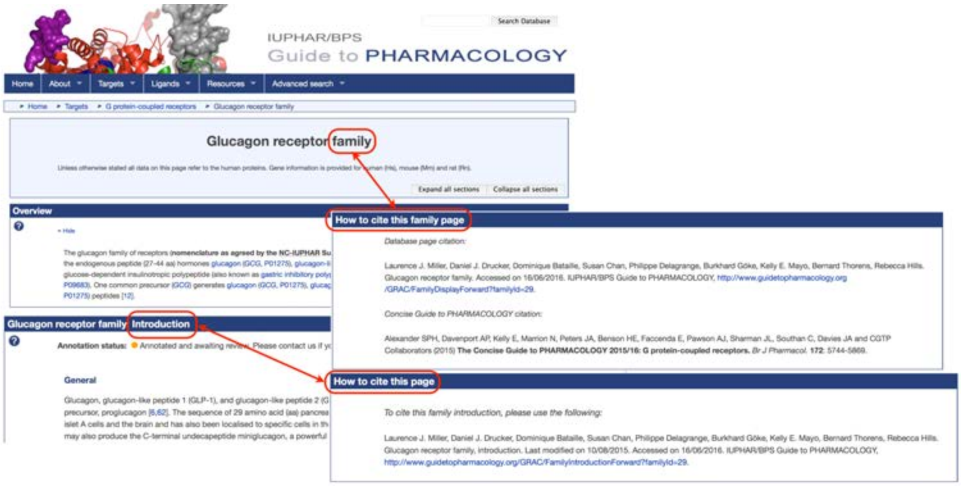
Information in GtoPdb is generated by hundreds of expert contributors, and different database entries are associated with different lists of contributors. While the suggested citation for GtoPdb as a whole (the root) is a traditional paper

[3] `http://dx.doi.org/10.1000/182`

[4] `http://confluence.ucop.edu/display/Curation/ARK`

[5] `http://www.ietf.org/rfc/rfc3986`

[6] `http://www.guidetopharmacology.org/`



Glucagon receptor family

Unless otherwise stated all data on this page refer to the human proteins. Gene information is provided for human (Hs), mouse (Mm) and rat (Rn).

The glucagon family of receptors (**nomenclature as agreed by the NC-IUPHAR Su**
the endogenous peptide (27-44 aa) hormones glucagon (GCG, P01275), glucagon-li
glucose-dependent insulinotropic polypeptide (also known as gastric inhibitory poly
P09683). One common precursor (GCG) generates glucagon (GCG, P01275), glucag
P01275) peptides [12].

Glucagon receptor family Introduction

Annotation status: ● Annotated and awaiting review. Please contact us if yo

General

Glucagon, glucagon-like peptide 1 (GLP-1), and glucagon-like peptide 2 (G
precursor, proglucagon [6,62]. The sequence of 29 amino acid (aa) pancrea
islet A cells and the brain and has also been localised to specific cells in th
may also produce the C-terminal undecapeptide miniglucagon, a powerful

How to cite this family page

Database page citation:

Laurence J. Miller, Daniel J. Drucker, Dominique Bataille, Susan Chan, Philippe Delagrange, Burkhard Göke, Kelly E. Mayo, Bernard Thorens, Rebecca Hills. Glucagon receptor family. Accessed on 16/06/2016. IUPHAR/BPS Guide to PHARMACOLOGY, http://www.guidetopharmacology.org/GRAC/FamilyDisplayForward?familyId=29.

Concise Guide to PHARMACOLOGY citation:

Alexander SPH, Davenport AP, Kelly E, Marrion N, Peters JA, Benson HE, Faccenda E, Pawson AJ, Sharman JL, Southan C, Davies JA and CGTP Collaborators (2015) **The Concise Guide to PHARMACOLOGY 2015/16: G protein-coupled receptors.** *Br J Pharmacol.* **172**: 5744-5869.

How to cite this page

To cite this family introduction, please use the following:

Laurence J. Miller, Daniel J. Drucker, Dominique Bataille, Susan Chan, Philippe Delagrange, Burkhard Göke, Kelly E. Mayo, Bernard Thorens, Rebecca Hills. Glucagon receptor family, introduction. Last modified on 10/08/2015. Accessed on 16/06/2016. IUPHAR/BPS Guide to PHARMACOLOGY, http://www.guidetopharmacology.org/GRAC/FamilyIntroductionForward?familyId=29.

Figure 1: GtoPdb Family and Introductory pages with independent citations

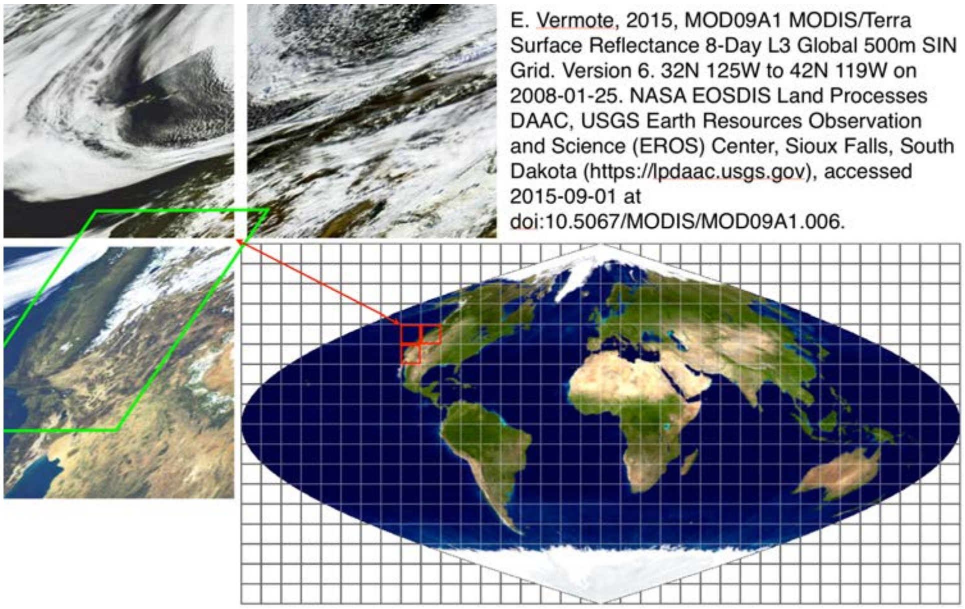
written by its curators, a citation to a subtree of GtoPdb includes the contributors who generated the content, see Figure 1. The citation may also depend on the path to the subtree (the query), as a few targets are members of more than one family, and the classification of the target is part of the citation. Queries against GtoPdb may return a boolean value or the empty set, and to cite this fact – for example to determine the relevant contributors – one clearly needs the query. A useful property of GtoPdb is that nearly all the information needed to construct a citation, such as names of contributors, is in the database itself.

2.2 MODIS

MODIS [24] (MODerate-resolution Imaging Spectrometer) is an electromechanical optical imaging system currently flying aboard NASA's Terra and Aqua satellites. Each MODIS sensor images the entire surface of the Earth every one to two days as a strip approximately 2000 km wide beneath the satellite's orbit. The MODIS sensor records the top-of-atmosphere radiance in several spectral bands, but MODIS data products typically process these values into Earth surface properties such as reflectance, snow cover, ocean color, etc.

MODIS data products are distributed as *granules*: fixed-sized subsets representing either an interval (typically 5 min) of the satellite's orbit or a tile within a standard map projection of all or part of the Earth (see Figure 2). Each MODIS granule is created, stored, and distributed as a Hierarchical Data Format file. MODIS data product search and access systems typically identify and return entire granules, not subsets thereof.

Each MODIS data product defines a granule naming convention, typically incor-



E. Vermote, 2015, MOD09A1 MODIS/Terra Surface Reflectance 8-Day L3 Global 500m SIN Grid. Version 6. 32N 125W to 42N 119W on 2008-01-25. NASA EOSDIS Land Processes DAAC, USGS Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota (https://lpdaac.usgs.gov), accessed 2015-09-01 at doi:10.5067/MODIS/MOD09A1.006.

Figure 2: The MODIS grid: highlighted tiles (red) of spatial extent for California (green), with citation

porating the product identifier, a version number, date-times of acquisition and generation, and (if applicable) a tile identifier. A granule name is thus a unique identifier for the granule, but is not in itself a complete citation for two reasons. First, applications of MODIS data products frequently use multiple granules, and there is no standard way to refer to a set of granules other than by complete enumeration. Second, applications of MODIS data products frequently focus on spatiotemporal regions of interest that are not precisely aligned with granule boundaries; thus, an application's query against a MODIS data product may not be precisely reflected in the corresponding set of product granules. For example, compare the latitude-longitude bounding box for California in Figure 2 with the non-rectangular set of MODIS tiles that intersect the box. While enumerating this set is important for provenance, a spatio-temporal bounding box is a compact description of the coverage, which – if expressed in a common co-ordinate system – allows easy searching for studies relevant to a particular region. Such bounding boxes are a common feature of geospatial citations; indeed a spatial bounding box is one of the optional fields in the DataCite schema [9].

3 Towards a Solution

We now address the problem of generating a citation for a query Q on database D. As with both GtoPdb and MODIS, the citation will depend on both Q and

D. This would appear to be a major problem, since anything that involves the analysis of a query or program is likely to be computationally expensive, if not undecidable. However, as we will see, the problem may be alleviated if we have a base of citations for *views* of D – *citable units* – which may then be used to generate citations for other queries. From a practical perspective, it is unlikely that a data publisher will be able to associate a citation with an arbitrarily complex query; however, it should be possible for them to say "For this part of the database, the citation should look like this". If several "parts of the database" can be formalized as a view, then we have a basis for generating citations.

3.1 Views and Citable Units

The standard notion of a database view is: given a database schema S, a *view* is some function V which, when applied to any instance of S (i.e., any database that conforms to S), produces a database in some other schema S'. Note that the input and output database schemas do not have to be in the same data model: we could, for example, have an XML view of a relational database. Views have been used in traditional database architectures to describe "areas of responsibility" for parts of a database. What we propose here is to use them to create "citable units"[7].

Figure 3 shows a simplified[8] representation of GtoPdb as a hierarchy, which is how it is published as web pages and understood by many contributors and users. There are four different classes of nodes in the hierarchy: the root, families, introductions (to families), and targets. Each of these nodes defines a view which is the subtree beneath it, and the GtoPdb curators have specified a different citation for each class. The higher levels of the hierarchy have citations with collaborators (editors or curators) and the lower levels with contributors. The curators of GtoPdb would like to carry citation down to the level of tables and tuples, but currently a citation for any other node in the hierarchy is the citation for the nearest ancestor of that node.

This is a promising start for defining citations for the hierarchical (web) presentation of the database, but recall that the underlying database is relational. How do we use these ideas to provide a citation for some SQL query against the database? We can turn this into a question about views. Suppose we have a database schema S, a view V over S and a query Q. If Q can be expressed as a query over V, then the citation associated with V is a *candidate citation* for Q. More formally, if, there is a query Q' such that, for all instances D of S, $Q(D) = Q'(V(D))$, then the citation for V is a candidate citation for Q.

[7] In the on-line appendix, *********** Section 2 has a discussion of citable units and Section 3 has some recommended reading on database views.

[8] To simplify presentation, we assume that families are all directly under the root in GtoPdb. In reality, some families may be grouped together as subfamilies of another family.

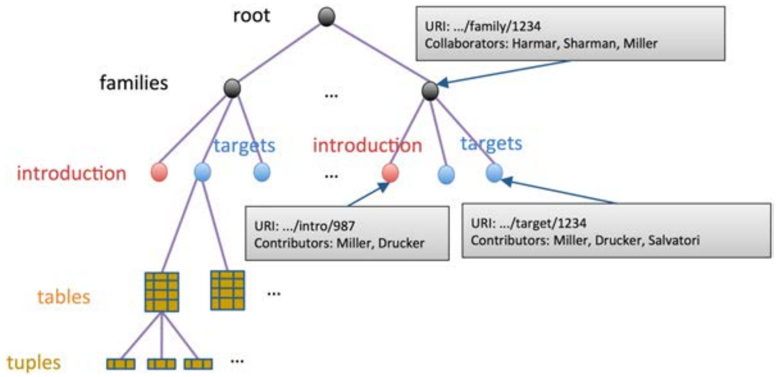


Figure 3: The GtoPdb hierarchy showing the citable views and some partial citations.

The view (the subtree) for each node in the hierarchy is given by a simple query on the underlying database. For example, there is a TARGET table whose primary key is a target identifier TID. For any value x of TID, and for any table that has TID as a foreign key, we select the rows that contain x. We now get a set of tables, each of which is a subset of the rows of the table in the original database. This is a view defined by x and each such value of TID defines a distinct *target view*. A similar construction works for families: there is a FAMILY table whose primary key is a target family identifier FID. For any value x of TID, and for any table that has FID as a foreign key, we select the rows that contain x. However, we also include in this view the union of tables of subfamilies of FID or (in the case of lowest level families), the union of target tables contained in FID. Each value of FID defines a distinct *family view*.

So the question of which citation to use for a relational query boils down to whether it can be answered using one of these relational views. Unfortunately, while simple to state, the problem of rewriting a query using views is non-trivial; it has been extensively studied in the context of query optimization, maintenance of physical data independence, and data integration [14, 17, 10]. The general problem is no simpler than program equivalence, which is undecidable; however, for answering *conjunctive queries* over *conjunctive views* the problem is NP-complete with practically efficient solutions. However, even if we are in a restricted situation where the problem is solvable, there may be 1) no views that support a given query; 2) more than one candidate view; or 3) the query may be expressible as a function on two or more candidate views, e.g., $Q(D) = Q'(V1(D), V2(D))$.

In spite of these issues, the formulation is useful in many practical cases, in particular when the views form a hierarchy that allows us to choose the "best" view from a candidate set.

3.2 Hierarchies of views

A hierarchy of views is formed by a view refinement (subview) relationship: Given two views W and V of the same database, we call W a *subview* of view V if there is a view W' such that $W(D) = W'(V(D))$ for all instances D of the database. Trivially, each view of the database is a subview of the view returning the database itself. What we want for a citation is a *smallest* view V for which Q is a subview.

In GtoPdb, there is a natural view hierarchy: the view for target TID is a subview of any family view which contains the target TID. In the hierarchical view of the data (Figure 3), the tree for TID is a subtree of the tree for FID; in the relational representation, each table in TID is a subset of the corresponding table in FID. Each view corresponds to a simple SQL conjunctive query over the relational representation, and for such views, one can determine whether a query can be answered using a view.

To specify simple views in a hierarchical structure, we can use a path language such as XPath.[9] For example, in GtoPdb there are three classes of view: one for the family page, one for the family introduction page, and one for the target page. We can specify them as follows:

Family view:	/Root/Family[FamilyName=$$f]
Introduction view:	/Root/Family[FamilyName=$$f]/Introduction
Target view:	/Root/Family[FamilyName=$$f]/Target[TargetName=$$t]

Each of these specifies a *class* of views, parameterized by variables indicated by $$. For the family and introduction view, each value of $$f gives us a view (a node in the tree) and for the target view we need both $$f and $$t. We shall refer to these views as *parameterized* views.

In the web interface to GtoPdb, each page is specified by a path from the root such as:

/Root/Family[FamilyName=”Melatonin”]/Target[TargetName=”MT1”]/LigandTable

This can be answered using the Target view defined above. It can also be answered by following the link in the Family view to “MT1”; however, the former is more specific and would therefore be the preferred citable unit. Recall that the citations for the two views could be different, as illustrated by the grey boxes in Figure 3.

Equally, suppose someone had queried the underlying database with a simple selection on the Family table with Name = 'Calcitonin'. Given that each citatable view in GtoPdb is a set of conjunctive queries, it is possible – and in this case easy – to determine that this could be answered using the Family view for Calcitonin.

As we have seen, it is possible that a query could be answered in two ways,

[9] http://www.w3.org/TR/xpath/

```
{ Title: "IUPHAR/BPS Guide to Pharmacology", Version: $v,
  Family: $$f, Contributors: $a, URI: "www.iuphar.org" }
⟵
/Root[VersionNumber: $v]/Family[FamilyName: $$f]/Introduction[Contributor-list: $a]

{ Title: "IUPHAR/BPS Guide to Pharmacology", Version: 26, Family: "Calcitonin",
  Contributors: ["Debbie Hay", "David R. Poyner"], URI: "www.iuphar.org" }
```

Figure 4: A citation specification and a sample result for GtoPdb

perhaps through the union of several Target views or through one Family view. This could be resolved through a policy by the data publisher or by presenting the alternatives to whoever wants to construct the citation.

3.3 Generating citations

Having set up a basis for identifying an appropriate citation, how do we generate one? Here we propose a simple rule-based language in which we use XPath syntax to define *patterns* that are matched against a hierarchy (the body of the rule) to produce the required citation (the head of the rule). Figure 4 shows a simple rule for generating a citation together with a citation that is generated by that rule. The right-hand side of the rule is an XPath-like expression that contains two kinds of variables: $\$\$x$ variables are the view parameters; and $\$x$ variables are bound once the $\$\$x$ variables have been matched. Here, the contributors, which depend on the family and the version number, which is unique to the database, are extracted.

The left-hand side of the rule contains the citation in whatever syntax is preferred. Here, we have assumed a simple JSON-style syntax, but the syntax could be in one of the numerous citation "styles", or some more generic syntax such as BibTeX [10] or DataCite [9]. In this example we have assumed that the database name and the URI are constants in the citation.

The sample result in Figure 4 is the citation for the simple path
/Root/Family[FamilyName="Calcitonin"]
It is also the citation for a simple SQL selection on the Family table with Name = 'Calcitonin' the SQL query above. In these cases, it is again easy to determine that it can be answered using the appropriate relational version of the Family view.

[10] http://www.bibtex.org/

3.4 Citations and MODIS

From a database perspective, MODIS is much simpler than GtoPdb. It is a hierarchically organized collection of products (e.g. surface reflectance products) consisting of a set of granules, which we assume for now are tiles (see Figure 2). A typical retrieval will ask for a set of tiles that cover a certain region of the Earth's surface and whose time stamp is within a given interval – a spatio-temporal bounding box of granules. For example, supposing one were interested in the surface reflectance for California on 25 January 2008, the granules could be specified by a bounding box whose latitude and longitude are the ranges [32,42] and [-125, -119][11] and time 2008-01-25.

The query to retrieve these granules can be expressed as a range query. If we group MODIS products into a hierarchy, our spatio-temporal query may be expressed in a path language as follows:

```
/root/product[ProdName="surface reflectance"]/file[Lat ≥ 32 and Lat ≤ 42 and
                                                   Lon ≥ -125 and Lon ≤ -119 and
                                                   Time = 2008-01-25]
```

This closely reflects the retrieval capabilities of many MODIS product distribution systems. To describe this common bounding box retrieval pattern, an appropriate parameterized view would be:

```
/root/product[ProdName=$$p]/file[ Lat ≥ $$minlat and Lat ≤ $$maxlat and
                                  Lon ≥ $$minlon and Lon ≤ $$maxlon and
                                  Time ≥ $$mint and Time ≤ $$maxt]
```

GtoPdb and MODIS differ in where they store information needed to construct the citation. In GtoPdb it is in the database, while in MODIS it is mostly kept elsewhere. This is easily solved by having functions in the citation rule that query an appropriate metadata repository with parameters extracted from the matching rule. For example, in Figure 5, m_auth() is a function that, given a product and version, queries the metadata for authorship. To our knowledge, there is currently no such organized metadata repository in MODIS, but having one would clearly be beneficial.

The version and access time (DATE function) are also not part of the view definition but can be calculated when the query is executed. Note that in MODIS, when newer analysis software becomes available, the entire database of products is re-analyzed yielding a complete new version; old versions are not kept. While this is undesirable from the standpoint of provenance and reproducibility, the citation carries useful information even though its referent may not exist.

[11] This is approximately the green box in Figure 2.

```
{ author: m_auth($p,$$v), m_year:($p,$$v), title: m_title($p), version: $v,
  bounding-box : [$$minlong, $$minlat, $$maxlong, $$maxlat], interval: [$$mint, $$maxt],
  organization: m_org($p), url: m_url($p), accessed: DATE(), doi = m_doi($p,$$v}
⟵
/root/product[ProdName=$p]/version[vnum=$$v]
    /file[Lat ≥ $$minlat and Lat ≤ $$maxlat and
        Lon ≥ $$minlon and Lon ≤ $$maxlon and
        Time ≥ $$mint and Time ≤ $$maxt]

{ author: "E. Vermote", title: "MOD09A1 ... SIN Grid", version: 6,
  bounding-box: [-125, 32, -119, 42], interval: [2008-01-25, 2008-01-25],
  organization: "NASA EOSDIS ... South Dakota", URL: "https://lpdaac.usgs.gov",
  accessed: "2015-09-01", doi: "10.5067/MODIS/MOD09A1.006" }
```

Figure 5: A citation rule for MODIS

4 Conclusions

We have addressed a critical issue in the adoption of data citation: automatically generating a citation from the query and database that was used to obtain the data. A preliminary implementation of the rule-based citation language for hierarchical data is reported in [5]. What we describe here is quite general and applies to any database with a well-defined query language. Rewriting queries through views was originally developed for query optimization and subsequently exploited in data integration. The idea of using them for data citation bears some relationship to that of using them to define security levels in a database [11].

We believe that using database views to specify citable units is key to both specifying and generating citations. It is important for any data publisher who wants their data to be properly cited to define these views, and to ensure that the data necessary to generate the citation from them is available. We have shown how this can be done for two quite different scientific databases, and we believe that the idea can work on forms of data such as RDF [25] and databases that are deployed in other fields such as the humanities. We have looked briefly at some examples, and the main issue is that the data needed to generate the citation may not be available, either in the database or some metadata repository[12].

We focussed on one specific computational problem in this paper, but it is almost impossible to do this in isolation from other topics such as citation standards. For example, the citation snippets required by the curators of our two examples do not quite conform to the DataCite metadata schema [9]: although DataCite has an entry for a spatial bounding box, it does not have one for a temporal

[12] See the use cases and linked data sections of in the annotated bibliography in Section 1 of the on-line appendix *********

interval as required by MODIS. A good problem for database research is to determine whether citations generated by a rule are consistent with a given citation schema.

We also mentioned archiving (fixity) and provenance as related computational challenges, but there are many others. We have tacitly assumed a rather conventional view of citations and how they will be used, but there are many ways in which this may radically change, e.g., the 10,000 author paper or the paper with 10,000 references. Maybe, by analogy with PageRank [19], there should be some notion of transitivity of credit in citation. These are all likely to require new ideas from computer science.

Acknowledgements: Tony Harmar, who led the development of GtoPdb, introduced us to the problem of data citation. We are also indebted to Sarah Cohen Boulakia, Jamie Davies, Wenfei Fan, Andreas Rauber, Joanna Sharman, Gianmaria Silvello and the reviewers much useful input. This work is supported by NSF IIS 1302212 and EPSRC SOCIAM EP/J017728/1.

References

[1] Altman, M., and King, G. A proposed standard for the scholarly citation of quantitative data. *D-Lib Magazine 13*, 3/4 (March/April 2007).

[2] American Geophysical Union. AGU publications data policy. `http://publications.agu.org/author-resource-center/publication-policies/data-policy/` (accessed Nov 2015), December 2013.

[3] American Meteorological Society. Data archiving and citation. `http://www2.ametsoc.org/ams/index.cfm/publications/authors/journal-and-bams-authors/journal-and-bams-authors-guide/data-archiving-and-citation/` (accessed Nov 2015).

[4] Ball, A., and Duke, M. How to cite datasets and link to publications. `http://www.dcc.ac.uk/resources/how-guides/cite-datasets` (accessed Nov 2015), June 2012.

[5] Buneman, P., and Silvello, G. A rule-based citation system for structured and evolving datasets. *IEEE Data Eng. Bull. 33*, 3 (2010), 33–41.

[6] Coalition on Publishing Data in the Earth and Space Sciences (COPDESS). Statement of commitment from earth and space science publishers and data facilities. `http://www.copdess.org/wp-content/uploads/2015/01/statementofcommitment.pdf` (accessed Nov 2015), January 2015.

[7] CODATA-ICSTI Task Group on Data Citation Standards and Practices. Out of cite, out of mind: The current state of practice, policy, and technology for the citation of data. *Data Science Journal 12* (2013), CIDCR1–CIDCR75.

[8] Data Observation Network for Earth (DataONE). Data citation and attribution. https://www.dataone.org/citing-dataone (accessed Nov 2015).

[9] DataCite. DataCite metadata schema for the publication and citation of research data. http://schema.datacite.org/meta/kernel-3/doc/DataCite-MetadataKernel_v3.1.pdf (accessed Nov 2015), October 2014.

[10] Deutsch, A., Popa, L., and Tannen, V. Query reformulation with constraints. *SIGMOD Record 35*, 1 (2006), 65–73.

[11] Fan, W., Chan, C.-Y., and Garofalakis, M. Secure XML querying with security views. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data* (2004), ACM, pp. 587–598.

[12] Federation of Earth Science Information Partners (ESIP). Data citation guidelines for data providers and archives. http://doi.org/10.7269/P34F1NNJ (accessed Nov 2015).

[13] FORCE11. Data citation synthesis group: Joint declaration of data citation principles. https://www.force11.org/datacitation (accessed Nov 2015), 2014.

[14] Halevy, A. Y. Answering queries using views: A survey. *VLDB J. 10*, 4 (2001), 270–294.

[15] International Council for Science Committee on Data for Science and Technology. Data citation standards and practices. http://www.codata.org/task-groups/data-citation-standards-and-practices (accessed Nov 2015), 2010.

[16] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S. Citation and peer review of data: Moving towards formal data publication. *International Journal of Digital Curation 6*, 2 (2011), 4–37.

[17] Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (New York, NY, USA, 2002), PODS '02, ACM, pp. 233–246.

[18] McCallum, I., Plag, H.-P., and Fritz, S. GEOSS data citation guidelines: Version 2.0. http://www.gstss.org/library/GEOSS_Data_Citation_Guidelines_V2.0.pdf (accessed Nov 2015), October 2012.

[19] Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[20] Pawson, A. J., Sharman, J. L., et al. The IUPHAR/BPS Guide to PHARMACOLOGY: an expert-driven knowledgebase of drug targets and their ligands. *Nucleic acids research 42*, D1 (2014), D1098–D1106.

[21] Pröll, S., and Rauber, A. Scalable data citation in dynamic, large databases: Model and reference implementation. In *Proceedings of the 2013 IEEE International Conference on Big Data* (2013), pp. 307–312.

[22] Pröll, S., and Rauber, A. A scalable framework for dynamic data citation of arbitrary structured data. In *DATA 2014 - Proceedings of 3rd International Conference on Data Management Technologies and Applications, Vienna, Austria, 29-31 August, 2014* (2014), pp. 223–230.

[23] Research Data Alliance (RDA) Working Group on Data Citation. Making data citable: Case statement. https://rd-alliance.org/group/data-citation-wg/case-statement/wg-data-citation-making-data-citable-case-statement.html (accessed Nov 2015).

[24] Salomonson, V. V., Barnes, W., and Masuoka, E. J. Introduction to MODIS and an overview of associated activities. In *Earth Science Satellite Remote Sensing: Vol. 1: Science and Instruments* (Berlin, Heidelberg, 2006), Springer Berlin Heidelberg, pp. 12–32.

[25] Silvello, G. A methodology for citing linked open data subsets. *D-Lib Magazine 21*, 1/2 (2015).