DOI:10.1145/2893181

Using database views to define citable units is the key to specifying and generating citations to data.

BY PETER BUNEMAN, SUSAN DAVIDSON, AND JAMES FREW

Why Data Citation Is a Computational Problem

CITATION IS ESSENTIAL to traditional scholarship. Citations identify the cited material, help retrieve it, give credit to its creator, 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 scholarly and scientific resources are held online, and many are in some kind of database, or 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 For example, most biological reference works have been replaced by curated databases, and vast amounts of basic scientific data—geospatial, astronomical, molecular, and more—are now available online. There is strong demand^{13,23} that these databases should be given the same scholarly status and appropriately cited, but how can this be done effectively?

Database citation is a challenge due to the structure and evolution of databases. Attributes such as ownership

and authorship may change for different parts of the database. Even for a simple collection of files, good methods may be needed for 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.

A citation is a collection of “snippets” of information (such as authorship, title, ownership, and date) that are specified by the database administrators and that 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, authors try to avoid typing in citations. Instead, they look for the citation in some database of citations, such as the ACM Digital Library^a and DBLP,^b and insert it into their document using a reference manager, such as BibTeX, Mendeley, and Zotero, or by copy-paste. In the context of citing databases, if the citation is not available or if the standard appears complicated, an author may well omit the citation or provide an inaccurate one. In short, unless citations are generated along with answers to data, the advocacy of data citation will have limited effect.

How can citations be automatically generated for data extracted from a

a <http://dl.acm.org/>

b <http://dblp.uni-trier.de/>

» key insights

- **Most information is now published in evolving databases or datasets; even traditional reference works are published as curated databases.**
- **Data citation depends on both the query and the data; since there is an unlimited number of queries, views can be used to specify citable units.**
- **A query may be supported by several different views; the choice of what citation to use for the query can be simplified when the views form a hierarchy.**

Figure 1. GtoPdb family and introductory pages with independent citations.

The screenshot shows the IUPHAR/BPS Guide to PHARMACOLOGY website. The main heading is "Glucagon receptor family". Below it, there is a section titled "Overview" with a "Hide" button. The text in the overview section describes the glucagon family of receptors. To the right of the overview, there is a section titled "How to cite this family page" which provides database and concise guide citations. Below the overview, there is a section titled "Glucagon receptor family Introduction" with an "Annotation status" of "Annotated and awaiting review". To the right of the introduction, there is a section titled "How to cite this page" which provides a citation for the family introduction. Red boxes highlight the "Glucagon receptor family" title, the "How to cite this family page" section, the "Introduction" section, and the "How to cite this page" section. Red arrows point from the title to the "How to cite this family page" section and from the "Introduction" section to the "How to cite this page" section.

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

database? Here, we use the term “database” in a broad sense and “query” to mean any mechanism used to extract the data, such as a set of file names, an SQL query, a URL, or a special-purpose GUI. The computational problem this poses can be broadly and simply formulated as:

Given a database D and a query Q , 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. In order to extract the citation *automatically* from the query Q and the database D , two questions

need to be answered:

- 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 them to construct citations for other queries?

If the retrieved data is simply a number or an image, one 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; so we need both Q and D .

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

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

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,^c ARKs,^d and URIs^e). These identifiers, while they may have certain fixed properties, do not guarantee that the cited material remains unchanged, a property known as “fixity.” Beyond observing that citations should reference the appropriate version, we do not address fixity in this article, nor do we address the closely related topic of provenance, which 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 Pröll and Rauber.^{21,22}

In this article, we propose a general approach to citation generation, and illustrate it using two scientific databases that are radically different in both their structure and how they should be cited.

Sample scientific datasets. We now describe these two databases. One is a curated relational database that is widely used in pharmacology; the other is a collection of files in a scientific data format that supports research in Earth sciences.

GtoPdb. The IUPHAR/BPS Guide to PHARMACOLOGY (GtoPdb)^{20,f} 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. This 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 webpages. The top level divides information by “families” of drug targets that reflect typical pharmacological thinking; lower levels divide the families hierarchically into subfamilies 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 imple-

Unless citations are generated along with answers to data, the advocacy of data citation will have limited effect.

mentation, 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 journal article 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, to, say, 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.

MODIS. MODERate-resolution Imaging Spectrometer (MODIS)²⁴ is an 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 2,000km 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, and color).

MODIS data products are distributed as granules—fixed-size subsets representing either an interval (typically five minutes) 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 of granules.

Each MODIS data product defines a granule naming convention, typically incorporating 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 itself a complete citation for two reasons. First, applications of MODIS data products frequently use multiple gran-

c <http://dx.doi.org/10.1000/182>

d <http://confluence.ucop.edu/display/Curation/ARK>

e <http://www.ietf.org/rfc/rfc3986>

f <http://www.guidetopharmacology.org/>

ules, 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; an application's query against a MODIS data product may thus 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 spatiotemporal bounding box is a compact description of the coverage that—if expressed in a common coordinate system—allows easy searching for studies relevant to a particular region. Such bounding boxes are a common feature of geospatial citations; a spatial bounding box is indeed one of the optional fields in the DataCite schema.⁹

Toward 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 dependence 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 show, the problem may be alleviated if we have a base of citations for certain “views” of D that may then be used to generate citations for other queries. From a practical perspective, it is unlikely that data publishers 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 views, then there is a basis for generating citations.

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 (or any database that conforms to S), produces a database in some other schema S' . Note the input and output database schemas do not have to be in the same data model; it is possible to, for example, have an XML view of a relational database. Views have been used in traditional database

A simple rule-based language using XPath-like syntax can be used to produce an appropriate citation when the views form a hierarchy.

architectures to describe “areas of responsibility” for parts of a database. What we propose here is to use views to create “citable units.”^g

Figure 3 is a simplified^h representation of GtoPdb as a hierarchy, which is how it is published as web pages and understood by many contributors and users. The hierarchy contains four different node classes: root, families, introductions (to families), and targets. Each of these nodes defines a view that 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 citations 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 can these ideas be used to provide a citation for some SQL query against the database? We can turn this question into one about views. Suppose we are given 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 .

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 in which the 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

^g The online appendix includes a discussion on citable units and recommended reading on database views.

^h To simplify our presentation, we assume families are all directly under the root in GtoPdb; however, in reality, some families may be grouped together as subfamilies of another family.

FAMILY table in which the 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 studied extensively in the context of query optimization, maintenance of physical data independence, and data integration.^{10,14,17} The general problem is no simpler than program equivalence, which is undecidable; but for answering “conjunctive queries over conjunctive views” the problem is NP-complete with practically efficient solutions. However, even in the restricted situation where the problem is solvable, there may be no views that support a given query; more than one candidate view; or the query may be expressible as a function on two or more candidate views, as in $Q(D) = Q'(V1(D), V2(D))$.

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

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, W is 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. The natural citation is the 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 that contains the target TID. In the hierarchical view of the data, as in 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 representa-

tion, and, for such views, it is possible to determine whether a query can be answered using a view.

To specify simple views in a hierarchical structure, a path language (such as XPath¹) suffices. 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. They are specified 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 them specifies a class of views, parameterized by variables indicated by $\$ \$$. For the Family and Introduction view, each value of $\$ \$ f$ gives a view (a node in the tree) and for the target view both $\$ \$ f$ and $\$ \$ t$ are needed. We refer to these views as “parameterized” views.

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

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

i <http://www.w3.org/TR/xpath/>

This can be answered using the Target view defined earlier. 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 gray 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 citable 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 mentioned, it is possible that a query could be answered in two ways, perhaps through the union of several Target views or through one Family view. This could be resolved through a policy specified by the data publisher or by presenting the alternatives to whoever wants to construct the citation.

Generating citations. Having set up a basis for identifying an appropriate citation, how do we generate one automatically? Here, we show how a simple rule-based language using XPath-like syntax can be used to produce an appropriate citation when the views form a hierarchy. In particular, XPath syntax is used 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

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

Figure 4. A citation rule and sample result for GtoPdb.

```
{ 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 5. A citation rule and sample result for MODIS.

```
{ 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" }
```

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 names of contributors are extracted. They depend on the family and on the version number, which is unique to the database.

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^j and DataCite⁹). In this example we have assumed 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”. In these cases, it is again easy to determine that the query can be answered using the appropriate relational version of the Family view.

Citations and MODIS. From a database perspective, MODIS is much

simpler than GtoPdb. It is a hierarchically organized collection of products (such as surface reflectance products) consisting of a set of granules we assume for now are tiles, as in Figure 2. A typical retrieval will ask for a set of tiles that cover a certain region of the Earth’s surface and in which the time stamp is within a given interval—a spatiotemporal bounding box of granules. For example, if a researcher was interested in the surface reflectance for California on January 25, 2008, the granules could be specified by a bounding box in which latitude and longitude are the ranges [32,42] degrees and [-125,-119] degrees^k and whose time is 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 spatiotemporal 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 example 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 for 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

^j <http://www.bibtex.org/>

^k This is approximately the green box in Figure 2.

in MODIS, when newer analysis software becomes available, the entire database of products is reanalyzed, yielding a complete new version; old versions are not kept. While this is undesirable from the standpoint of provenance and reproducibility, the citation still carries useful information, even though its referent may not exist.

Conclusion

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 given in Buneman and Silvello.⁵ What we have described 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 views for data citation bears some relationship to that of using them to define security levels in a database.¹¹

Using database views to specify citable units is the key to both specifying and generating citations. It is important for data publishers who want their data to be properly cited to define these views and ensure 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 believe the idea can work on forms of data (such as RDF²⁵) and databases in other fields, including the humanities. We have looked at some examples, and the main barrier is that the data needed to generate the citation may not be available, either in the database or in some metadata repository.¹

In this article, we have focused on the problem of automatically generating citations, but it is almost impossible to do it 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 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 (ensuring 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 are used, but there are many ways in which the form and use of citations may change radically (such as papers with 10,000 authors or papers with 10,000 references). Maybe, by analogy with PageRank,¹⁹ there should be some notion of transitivity of credit in citation. These new approaches to the content, structure, and use of citations are all likely to require new ideas from computer science.

Acknowledgments

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 for much useful input. This work is supported by National Science Foundation Information and Intelligent Systems grant 1302212 and Engineering and Physical Sciences Research Council grant EP/J017728/1 The Theory and Practice of Social Machines.

References

- Altman, M. and King, G. A proposed standard for the scholarly citation of quantitative data. *D-Lib Magazine* 13, 3/4 (Mar./Apr. 2007).
- American Geophysical Union. *AGU Publications Data Policy*. Washington, D.C., Dec. 2013; <http://publications.agu.org/author-resource-center/publication-policies/data-policy/> (accessed Nov. 2015)
- American Meteorological Society. *Data Archiving and Citation*. Boston, MA; <https://www.ametsoc.org/ams/index.cfm/publications/authors/journal-and-bams-authors/journal-and-bams-authors-guide/data-archiving-and-citation/> (accessed Nov. 2015)
- Ball, A. and Duke, M. *How to Cite Datasets and Link to Publications*. The Digital Curation Centre, Edinburgh, U.K., 2012; <http://www.dcc.ac.uk/resources/how-guides/cite-datasets> (accessed Nov. 2015)
- Buneman, P. and Silvello, G. A rule-based citation system for structured and evolving datasets. *IEEE Data Engineering Bulletin* 33, 3 (2010), 33–41.
- Coalition on Publishing Data in the Earth and Space Sciences (COPDESS). Statement of commitment from earth and space science publishers and data facilities, 2015; <http://www.copdess.org/statement-of-commitment/> (accessed Nov. 2015)
- 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, 0 (2013), CIDCR1–CIDCR75.
- Data Observation Network for Earth (DataONE). *Data Citation and Attribution*; <https://www.dataone.org/citing-dataone> (accessed Nov. 2015)
- 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

- Deutsch, A., Popa, L., and Tannen, V. Query reformulation with constraints. *SIGMOD Record* 35, 1 (2006), 65–73.
- 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*. ACM Press, New York, 2004, 587–598.
- Federation of Earth Science Information Partners (ESIP). *Data Citation Guidelines for Data Providers and Archives*; <http://doi.org/10.7269/P34F1NNJ>
- FORCE11. Data Citation Synthesis Group: Joint Declaration of Data Citation Principles; <https://www.force11.org/datacitation> (accessed Nov. 2015)
- Halevy, A.Y. Answering queries using views: A survey. *The VLDB Journal* 10, 4 (2001), 270–294.
- International Council for Science Committee on Data for Science and Technology. *Data Citation Standards and Practices*, 2010; <http://www.codata.org/task-groups/data-citation-standards-and-practices> (accessed Nov. 2015)
- 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.
- Lenzerini, M. Data integration: A theoretical perspective. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Madison, WI, June 3–6). ACM Press, New York, 2002, 233–246.
- McCallum, I., Plag, H.-P., and Fritz, S. GEOSS data citation guidelines: Version 2.0, 2012; http://www.gstss.org/library/GEOSS_Data_Citation_Guidelines_V2.0.pdf (accessed Nov. 2015)
- Page, L., Brin, S., Motwani, R., and Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Stanford InfoLab, Stanford University, Nov. 1999.
- 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.
- Pröhl, 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* (Santa Clara, CA, Oct. 6–9). IEEE Press, 2013, 307–312.
- Pröhl, S. and Rauber, A. A scalable framework for dynamic data citation of arbitrary structured data. In *Proceedings of the Third International Conference on Data Management Technologies and Applications* (Vienna, Austria, Aug. 29–31, 2014), 223–230.
- Research Data Alliance 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)
- 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*. Springer, Berlin, Heidelberg, Germany, 2006, 12–32.
- Silvello, G. A methodology for citing linked open data subsets. *D-Lib Magazine* 21, 1/2 (2015).

Peter Buneman (opb@inf.ed.ac.uk) is a professor in the School of Informatics at the University of Edinburgh, Edinburgh, U.K.

Susan Davidson (susan@cis.upenn.edu) is the Weiss Professor of Computer and Information Science at the University of Pennsylvania, Philadelphia, PA.

James Frew (frew@ucsb.edu) is an associate professor of environmental informatics in the Bren School of Environmental Science & Management at the University of California, Santa Barbara, CA.

© 2016 ACM 0001-0782/16/09 \$15.00

Watch the author discuss her work in this exclusive *Communications* video. <http://cacm.acm.org/videos/why-data-citation-is-a-computational-problem>

¹ See use cases and linked data sections in the annotated bibliography in the online appendix.