Journal of Informetrics 14 (2020) 101080

Contents lists available at ScienceDirect

Journal of Informetrics

journal homepage: www.elsevier.com/locate/joi

Regular article

Data credit distribution: A new method to estimate databases impact

Dennis Dosso*, Gianmaria Silvello

Department of Information Engineering, University of Padua, Italy

ARTICLE INFO

Article history:
Received 10 April 2020
Received in revised form 10 August 2020
Accepted 13 August 2020

Keywords:
Data citation
Data provenance
Data credit distribution

ABSTRACT

It is widely accepted that data is fundamental for research and should therefore be cited as textual scientific publications. However, issues like data citation, handling and counting the credit generated by such citations, remain open research questions.

Data credit is a new measure of value built on top of data citation, which enables us to annotate data with a value, representing its importance. Data credit can be considered as a new tool that, together with traditional citations, helps to recognize the value of data and its creators in a world that is ever more depending on data.

In this paper we define *data credit distribution* (DCD) as a process by which credit generated by citations is given to the single elements of a database. We focus on a scenario where a paper cites data from a database obtained by issuing a query. The citation generates credit which is then divided among the database entities responsible for generating the query output. One key aspect of our work is to credit not only the explicitly cited entities, but even those that contribute to their existence, but which are not accounted in the query output.

We propose a data *credit distribution strategy* (CDS) based on data provenance and implement a system that uses the information provided by data citations to distribute the credit in a relational database accordingly.

As use case and for evaluation purposes, we adopt the IUPHAR/BPS Guide to Pharmacology (GtoPdb), a curated relational database. We show how credit can be used to highlight areas of the database that are frequently used. Moreover, we also underline how credit rewards data and authors based on their research impact, and not merely on the number of citations. This can lead to designing new bibliometrics for data citations.

1. Introduction

Citations are the "currency" of the scientific world and are essential for the dissemination of knowledge and scientific development. They can be considered as fundamental basis to achieve scholarship, and the key to give credit to authors, papers, and venues (Zou & Peterson, 2016). In this context, they serve multiple purposes: crediting an idea, paying homage to pioneers, reflecting one's knowledge of literature, or as a critique to the work of others (Cousijn, Feeney, Lowenberg, Presani, & Simons, 2019; Cronin, 1984). Citations are used as a criterion to decide on tenure, promotion, hiring, and funding grants (Cronin, 2001; Hartley, 2017; Kosten, 2016; Meho & Yang, 2007).

* Corresponding author.
E-mail addresses: dosso@dei.unipd.it (D. Dosso), silvello@dei.unipd.it (G. Silvello).

https://doi.org/10.1016/j.joi.2020.101080

Nowadays, science and research are mainly digital, and (traditional) papers are no longer the sole source of knowledge and citations. Indeed, curated scientific databases – which are "populated and updated with a great deal of human effort" (Buneman, Cheney, Tan, & Vansummeren, 2008) – are numerous and at the core of current scientific research (Buneman, Davidson, & Frew, 2016).

As globally accepted, data must be cited and citable (Callaghan et al., 2012; CODATA-ICSTI Task Group on Data Citation Standards & Practices, 2013; Lawrence, Jones, Matthews, Pepler, & Callaghan, 2011). There is also a growing belief that data citations should contribute to scientific reputation of researchers, scientists, data curators, and creators (Altman, Borgman, Crosas, & Martone, 2015; Spengler, 2012). Data citations should be also counted alongside traditional citations and contribute to bibliometrics indicators (Belter, 2014; Peters, Kraker, Lex, Gumpenberger, & Gorraiz, 2016). The rise of FAIR (*Findability, Accessibility, Interoperability, and Reusability*) principles (Wilkinson et al., 2016) further propels the need to cite data and count data citations. Such necessary practices shed light on the work of the creators and curators of datasets, work that would otherwise remain uncredited (Asmi, Rauber, Pr“oll, & van Uytvanck, 2016; Buneman, 2006; Candela, Castelli, Manghi, & Tani, 2015; Wu, Alawini, Davidson, & Silvello, 2018; Zwölf, Moreau, Ba, & Dubernet, 2019). Much of the work in the current literature considers the development of data citation as a driving force to "facilitate giving scholar credit" (Martone, 2014).

How to credit data creators and curators with the correct level of granularity is one of the central aspects of data citation research. Data is often cited at the "database level" or the "webpage level". In the first case, the whole database is cited, therefore all credit goes to database key personnel. In the second case the database has a website with webpages that can be cited instead. These pages are obtained by data aggregated by topic through the use of specific queries, and are built in a way to resemble a traditional research paper. Often the creators and curators of the webpages data are not credited or only marginally credited for their work (Alawini, Davidson, Silvello, Tannen, & Wu, 2018).

This lack of recognition greatly hinders the sharing of data and research results, as highlighted by the "research parasites" controversy (Longo & Drazen, 2016), i.e. researchers who steal the work of others for their own ends. Appropriate data citation is therefore essential for a healthy collaboration among researchers.

Hence, research on (data) credit attribution – i.e. *data credit distribution* (DCD) – has been recently attracting increasing attention (Fang, 2018; Katz, 2014; Zeng, Wu, Bratt, & Acuna, 2020). Nevertheless, DCD has the goal to improve and expand the reach of data citation techniques, but it is not an alternative to it. This means that to employ DCD techniques, data citations (at any level of granularity) must be available.

Data credit herein is considered as a data value measure in a (curated) database. The concept of "data" is left intentionally vague, since credit can be assigned to data of any kind and level of granularity – e.g., to a single attribute, record, table, on-the-fly generated data set or database as a whole. The credit on a "datum" of any kind, size, and nature, can be assigned as a *real* positive value, acting as a proxy of a value measured in terms of citations, accesses, clicks, downloads or other surrogates for data use. We call DCD the process, method, or algorithm employed to assign credit to a given datum or data set.

As example, let us consider a scientific paper p_1 presenting a new drug α to treat the disease β caused by a variation of the gene γ. Evidence on gene-disease association β-γ is based on the record r_1 in the curated database D. The "value" of the citation from p_1 to r_1 can be transformed into an amount of credit x (e.g., $x = 1$): the way in which x is assigned to r_1 is what we call *credit distribution strategy* (CDS). If we assign all the credit x to r_1, then all and only the curators of r_1 are credited. Whereas, if we assign x to D, then all and only the database administrators are credited – this is what typically happens when we cite data papers as proxies for the databases they describe (Candela et al., 2015). On the other hand, if we distribute x in part to r_1 and in part to other records, say r_2 and r_7, which somehow contributed to the generation of r_1, then the curators of r_1, r_2 and r_7 are credited. The amount of credit given to r_1, r_2 and r_7 is an additional aspect of CDS.

DCD can therefore be used as a *tool* alongside citations to reward data and/or data creators and curators. DCD differs from common citation procedures that we are accustomed to, in particular:

1 In a traditional setting, when a paper cites another paper, one citation is given to the cited paper (and to its authors). It does not matter why and how paper p_1 is citing paper p_2[1] : the result is always a +1 "credit" from p_1 to p_2 and thus a +1 to the citation count of the authors of p_2. With a different credit distribution strategy, the "value" given to the cited entity can be *proportional* to the role played in the citing entity.
2 Traditional citations are considered as *atomic*. One citation from p_1 to p_2 can never be broken into pieces and assigned in part to p_2 and in part to other papers or data, which contributed to p_2. This is due to the intrinsic difficulty in grasping the role and "weight" of the other papers and data, and in automating the credit assignment process. Instead, we consider data credit as a *non-atomic* real value, which can be divided and distributed to multiple components of a database. Hence, we can weigh the importance of the cited entities and assign credit according to their role, as proposed to some extent in a traditional setting by the zp-index (Zou & Peterson, 2016) for the role of authors in a paper.

[1] Worth noting that there is vast research on the topic and many alternative proposals, but none of them currently work on a large scale.

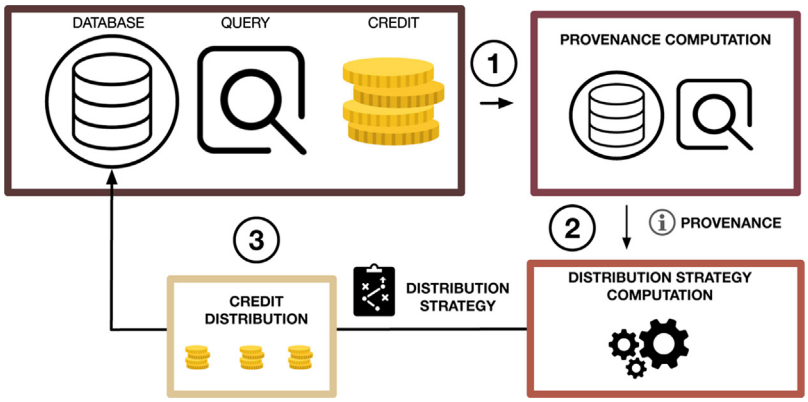

Fig. 1. Overview of the credit distribution pipeline.

This paper focuses on the distribution of the credit generated by *data citations* over *relational databases* (RDBs). We also consider the most common scenarios where papers cite data *obtained by a query* over the RDB.

The following reasons motivate the study on RDBs:

- RDBs are pervasive in the scientific world. Most scientific curated databases are relational. RDBs are the main focus of data citation methods as much of the work discusses how to cite them and get them cited (Buneman et al., 2016; Buneman & Silvello, 2010; Pröll & Rauber, 2013). Moreover, many scientific curated RDBs are accessible via Webpages dynamically generated via queries to the database.
- RDBs, being well-consolidated technologies, are widely used. The "relational database market alone has revenue upwards of $50B" (Abadi et al., 2020). Known outside the database community, they are often the test-bed for new methods that can be adapted to other databases, e.g., graphs or document databases.
- In an RDB, the data portions that can be credited may easily be defined. In particular, we consider the following: (i) the whole database, (ii) the tables, and (iii) the tuples.

1.1. Overview of the DCD pipeline

In Fig. 1, the DCD process is graphically represented. The problem input is an RDB instance I, a query Q, and the credit represented as a real value $k \in \mathbb{R}_{>0}$.

The process to split and distribute k is based on the *data provenance* methods presented in (Cheney, Chiticariu, and Tan, 2009) the provenance of $Q(I)$ describes the data generation process undertaken by Q, and the data used in I to generate the output. There are several kinds of data provenance for RDBs (Cheney et al., 2009). In this paper we focus on a specific provenance method known as *lineage* (Cui, Widom, & Wiener, 2000). The lineage of a tuple t in the output of $Q(I)$ is the set of all and only the tuples used by Q to generate that given tuple t. It is, therefore, useful to know which tuples in I deserve credit. The lineage is computed in step 1.

The provenance is the input for the computation of the *credit distribution strategy* (CDS) in step 2. The CDS is a function that distributes k on the basis of data provenance. It is therefore closely related to the kind of provenance computed in step 1. Hence, in this work we describe a CDS based on lineage, even if it can be defined by different strategies which, in turn, depend on the provenance method adopted. CDS functions are to be defined on the basis of citation policies decided at the database administration level or, even better, at the domain community level. Certainly, we are in a one-solution-does-not-fit-all scenario, but CDS can be defined with great variability and flexibility, thus allowing for ample customization.

The computed CDS is finally used to assign credit to different tuples in the database (step 3).

Our testbed is a well-known curated database, the IUPHAR/BPS[2] Guide to Pharmacology (Harding et al., 2018), also known as GtoPdb,[3] which contains expertly curated information about diseases, drugs, cellular drug targets and their mechanisms of action. We chose GtoPdb for two main reasons: (i) it is a widely used and valuable curated relational database, (ii) many papers in the literature use and cite its data (i.e., families, ligands, and receptors). Real queries used in papers can therefore be seen as data citations which, in turn, can be used to assign data credit.

This work presents different DCD scenarios which consider the queries issued on GtoPdb and how the credit associate to citations can be distributed. The first set of experiments shows how uniform distribution of credit can highlight parts of the database, according to their relevance to one "topic".

[2] International Union of Basic and Clinical Pharmacology/British Pharmacology Society.
[3] https://www.guidetopharmacology.org/

The second set of experiments assigns credit following a Pareto distribution, i.e. a few queries are issued several times assigning much more credit than the others. These experiments serve to show how credit highlights "hotspots" in the database where data are used the most.

In the third set of experiments, real queries are extracted from papers citing GtoPdb. Differences in behavior are observed between citations count and data crediting. In particular, credit can help reward authors with fewer citation whose work may greatly impact research communities. This experiment reveals not only how credit rewards data, but also how it differs from citations count in rewarding papers and authors.

Our paper *contributes* to the following:

- Defining data credit distribution with specific focus on RDB;
- Defining data credit distribution strategy and its implementation based on lineage;
- In-depth analysis of the effects of credit distribution over real-world curated data and the differences compared to traditional citation counts.

1.2. Outline

The rest of paper is organized as follows: Section 2 presents the related works; Section 3 describes the use case we adopted; Section 4 defines in detail data credit distribution and our method to solve it; in Section 5 we present the evaluation of our approach. Finally, Section 7 presents our conclusions and potential future work.

2. Related work

2.1. Data in research

We are experiencing rapid transition toward the *fourth paradigm of science*, data-intensive scientific discovery, where data are important for scientific advances as well as for traditional publications (Bechhofer et al., 2013). More and more scientific data from different scientific domains like astronomy, geology, pharmacology, medicine, and geo-spatial science are contained in, and made available through, structured, evolving, and often distributed curated databases (Buneman et al., 2016).

The scientific community is, therefore, promoting an *open research culture* (Nosek et al., 2015) founded on methods and tools to share, discover, and access experimental data. Scientific databases should be FAIR (Wilkinson et al., 2016) (Findable, Accessible, Interoperable, and Reusable). In particular, data should be accessible from the articles, journals, and papers that cite or use them (Cousijn et al., 2019).

Research aspects such as *reproducibility* of experiments, the *availability* of scientific data, and the *connection* between data and scientific results are all relevant to the domain of *data citation* (Honor, Haselgrove, Frazier, & Kennedy, 2016).

2.2. Data citation: principles

Data citation principles were first described in detail in CODATA-ICSTI Task Group on Data Citation Standards and Practices (2013). They were later summarized and endorsed by the Joint Declaration of Data Citation Principles (JDDCP) (Martone, 2014).

The principles are divided into two groups. The first contains principles concerning the role of data citation in scholarly and research activities; the second defines the main guidelines to establish a data citation system (Silvello, 2018).

The first group contains such concepts as: (i) *importance*, why data citation is important and why data should be considered as first-class citizen; (ii) *credit* and *attribution*; (iii) *evidence*; (iv) *verifiability* (these last 3 principles derive from the idea that people and institution responsible for the creation and maintenance of data should receive appropriate citations to their work); (v) *interoperability*, which requires data citation methods to be flexible enough to operate through different communities.

The second group of principles defines the requisites for data citation methods. These include: (i) *unique identification* of the data being cited, based on machine-actionable methods; (ii) (open) *access* to data; (iii) guaranteeing that *persistence* and *availability* of citations even after the lifespan of the cited entity (i.e., *fixity* of the citation); and (iv) *specificity*, i.e., a citation must lead to the data set originally cited.

2.3. Data citation: motivations

The main motivation behind the study and pursuit of data citation is that "data in research are as valuable as papers and monographs" (Ball & Duke, 2011).

Six motivations for data citation, shared by many scientific fields, can be outlined (Silvello, 2018):

- *Data attribution*, identifying the individuals that should be credited and/or accountable for data with variable granularity.

- *Data connection*, connecting papers to the data being used.
- *Data discovery*, citations to data records or subsets may act as entry points to data otherwise not findable via search engines.
- *Data sharing*, sharing data obtained by researchers within the whole community. Currently, researchers fear losing their competitive advantage and not receiving adequate credit for their effort (Mooney & Newton, 2012).
- *Data impact*, interpreted as a way to highlight the results obtained in writing papers using specific data, the frequency and modality data were used. Our paper offers new input in this domain.
- *Reproducibility*, data citation greatly impacts the reproducibility of science (Baggerly, 2010). Many authoritative journals asked to share data and provide valid methodologies to reproduce experiments.

2.4. Data citation in relational databases

RDBs have been the main target of data citation methods since the surge of the data-centric research paradigm. In particular, the RDA (Resource Data Alliance)[4] promoted a working group on data citation focusing specifically on RDBs. RDA is a community-driven initiative launched in 2013 by different commissions, including the European Commission and the United States Government's National Science Foundation. Its goal is to build the social and technical infrastructures to enable open sharing and re-use of data. RDA members come together through focused Working Groups and Interest Groups, formed by experts from all around the world (academia, the private sector and government).

The RDA "Working Group on Data Citation"[5] (Pröll & Rauber, 2013; Rauber, Ari, van Uytvanck, & Pröll, 2016) has been working in the last years on large, dynamic and changing datasets. The working group has recently finalized its guidelines for data citation, and has now moved on into an adoption phase. The datasets considered by the WG are in most cases relational.

In one of its most recent sessions (RDA Data Citation Working Group, 2020), the Working Group (WG) on Data Citation reported that there are various implementations of its guidelines for Data Citation on MySQL/Postgres relational databases. Some of these databases are: DEXHELPP[6] (Social Security Records); NERC (ARGO Global Array); EODC (Earth Observation Data Centre) (Gößwein, Miksa, Rauber, & Wagner, 2019); LNEC (River dam monitoring); MDS (Million Song Database) (Bertin-Mahieux, Ellis, Whitman, & Lamere, 2011); CBMI[7] (Center for Biomedical Informatics); VMC (Vermont Monitoring Cooperative); CCA[8] (Climate Change Center Austria); and, VAMDC (Virtual Atomic and Molecular Data Center) (Dubernet et al., 2016; Zwölf, Moreau, & Dubernet, 2016).

All these initiatives and databases support the RDA WG specifications for Data Citation, instantiated based on their needs and specific structures. This means that in all these databases it is possible to implement DCD as a tool built on top of the Data Citation techniques.

More work on data citation on relational databases has been carried out outside the scope of the RDA (Alawini, Davidson, Hu, & Wu, 2017; Buneman et al., 2016; Buneman & Silvello, 2010; Davidson, Deutch, Milo, & Silvello, 2017; Wu et al., 2018). Relational databases are pervasive in the research world and are at the basis of widely used curated databases; for instance the website https://fairsharing.org/ keeps a long updated list of curated and scientific databases (many of which are relational or graph-based) following FAIR guidelines. These databases are citable since they are complaint with the most recent guidelines, and they are in the vast majority of cases accessible via dynamically created Webpages.

Data citation techniques are primarily applied to relational database because of their diffusion and also because the portions of data that are to be cited are easily identified: the whole database, a relation, a tuple, or an attribute. Many papers (Alawini et al., 2017; Buneman, 2006; Buneman et al., 2016) consider more complex citable units, recognizing that often the *views* of a database are the ones to be cited. Generally, a *view* is a query on the database. To this end, Wu et al. (2018) suggested decomposing the database in a set of views, where each view is associated to its citation. Through a well-crafted algorithm that generates combinations of views, new citations can be built from the ones associated with the views every time a query is issued. The most common queries over relational databases, both for citation and view building purposes, are the so-called *conjunctive queries* (for a complete reference, see Abiteboul, Hull, & Vianu, 1995). These queries are "universal" across different types of database, such as relational, semistructured, and RDF. They also simplify the reasoning process used in generating citations.

At present, the most common practices to cite databases include:

1. A database cited as a whole, even though only parts of the databases are used in the papers or datasets. Alternatively, the so called "data papers" can be cited, being traditional papers that describe a database (Candela et al., 2015). An example is paper (Harding et al., 2018), which, every few years, describes the information contained in GtoPdb. This data paper works as a proxy of the database and receives all its citations.

   In this case, all the credit from the citations go to the database administrators or to the authors of the data papers.

[4] https://www.rd-alliance.org/
[5] https://www.rd-alliance.org/groups/data-citation-wg.html
[6] http://www.dexhelpp.at/
[7] https://medicine.missouri.edu/centers-institutes-labs/center-for-biomedical-informatics
[8] https://ccca.ac.at/startseite

2. Subsets of data, obtained by issuing queries to a database, are individually cited. This is the solution adopted by the *Resource Data Alliance* (RDA) working group on Data Citation (Rauber et al., 2016).

   In this case, the credit from citations can be distributed amongst those contributing to the portions of data returned by the cited queries and/or to the database administrators.
3. The database is accessible via a series of Webpages that arrange the content of the database by topic or theme. Examples in the life science domain include the Reactome Pathway database (Joshi-Tope et al., 2005), the GtoPdb (Harding et al., 2018), and the VAMDC (Zwölf et al., 2016). Every single Webpage is unequivocally identifiable and can be individually cited. These databases are particularly interesting because a single Webpage can be considered as the output of a set of several queries issued on the database. The information extracted from these queries is then displayed on the page, composing its sections. Thus, citing a Webpage is equivalent to citing the set of queries used to create it. Citing only one section is equivalent to citing the single query (or the smaller set of queries) that produced that section.

   For instance, credit can be distributed amongst those who contributed to the data used to build the Webpages and/or to those who created the Webpages and/or to the database administrators.

Despite all the research efforts dedicated to the study and promotion of data citation, none of the largest citation-based systems, such as Elsevier Scopus, Web of Science, Microsoft Academia or Google Scholar, consider scientific datasets as citable objects in academic work. Clarivate Analytics Data Citation Index (DCI) (Force, Robinson, Matthews, Auld, & Boletta, 2016) is notably an exception. DCI is an infrastructure that tracks of data usage in scientific domains and provides the technical means to connect datasets and repositories to scientific papers. However, DCI considers only citations to (previously registered and approved) databases as a whole and do not count citations to database portions such as views, tables or tuples.

Publishers, data centers, and indexing services have started to create bidirectional links between research data and scholarly literature. Such links, however, usually stem from agreements implemented by two organizations. They therefore lack a universally accepted industrial standard, and each agreement differs from the other (Burton et al., 2017). The rapid growth of bilateral agreements hinders interoperability, that is one of the principles of data citation. In fact, this kind of agreement has generated a series of undesirable side-effects. Many publishers, data centers, repositories, and infrastructure providers remain disconnected. Moreover, the heterogeneity that ensues from (considerably different) agreements and practices hinders the global interoperability among different agreements. One example of such heterogeneity may be found in identification systems such as Digital Object Identifiers[9] (DOI) and Life Science Identifiers[10] (LSID).

The Scholix framework (Burton et al., 2017) addresses this issue. As community and multi-stakeholder driven effort, it strives to facilitate information exchange between data and literature and between data and data. It can be regarded as a framework, a set of guidelines and lightweight models to facilitate interoperability among link providers.

2.5. Emergence of data credit

In recent years, the idea of crediting data and software emerged in the academic discussion. Data credit is related to data citation, since they both refer to recognizing the work of data creators and curators. In a sense, data credit can be seen as a by-product of data citation since credit attribution is not possible without the citations to data.

Katz (2014) suggests the need for a *modified citation system* that includes the idea of *transient* and *fractional credit*, to be used by developers of research products as software and data. This idea stems from two observations: (i) currently, data and software are not formally rewarded or recognized; (ii) even in traditional papers, the contribution of each author to the work is hard to understand, unless explicitly specified in the paper. This is generally true for data, where different groups of people work on the same database.

Therefore, in Katz (2014) credit is defined as a "quantity" that describes the importance of a research entity, such as papers, software or data mentioned in a citation. However, we stem from this definition of credit, arguing that the concept of credit can be built on top of the existing infrastructure handling traditional and data citations.

The underlying idea in Katz (2014) is that a *distribution* of credit from research entities (i.e., papers and data) to other research entities through citations that connect them can be performed. Indeed, thanks to traditional citations and now also to data citations, this distribution is finally possible between papers and data. Hence, Katz (2014) highlights some problems that credit can solve:

1. Credit rewards research entities, such as data that to date are not (formally) recognized (this is also one of the goals of data citation).
2. Credit can reward research entity authors similarly to what citations do.
3. Credit can reward authors in a *proportionate* way, taking into consideration their role in generating of the entity – "Some journals have tried to solve this problem by requiring that the contribution of each author be defined [...]. A technologically simple solution is to give partial credit to all authors, which can also be done for software and data. Arguably, determining

[9] https://www.doi.org
[10] http://www.lsid.info

how to weight credit of the authors may be difficult, but it should be possible". This is something that only credit can do, since traditional citations are *atomic* in nature.

4. Credit can be *transitively* transmitted. For example, if a paper A cites a paper B, some credit goes from A to B. Then, if B in turn cites data contained in the database C, a fraction of the credit received by B from A could be transitively transmitted to C. "The primary value of transitive credit is in measuring the indirect contributions to a product, which today are not quantitatively captured". This is also something that only credit can do, but is possible because of the existence of a network of citations amongst the entities.

A solution to these problems could drive more and more researchers to share and disclose their results and data. This goal is shared with data citation, and credit can help to achieve it.

Fang (2018) presents a framework to distribute the credit generated by a paper to its authors and to the papers in its reference list in a transitive way. Let us consider the *citation graph* as the graph where the nodes are papers and the links are the citations amongst them. In this graph, every paper is a source of credit, which is then transferred to the neighboring nodes. The quantity of credit received by each cited paper depends on its impact/role in the citing paper. So far, this theoretical framework is limited to papers, but it can be easily extended to a citation graph that comprises papers and data.

Zeng et al. (2020) propose the first method designed to compute credit within a network of papers citing data. Adopting a network flow algorithm, they simulate a random walker to estimate a score for each dataset, leveraging real-world usage data to compute the credit.

This is a first step toward an automatic credit computation procedure. However, it is limited to assigning credit to the whole datasets, without considering the granularity of data. Therefore, this is not a way to assign credit to a single research entity within a dataset.

Differently from Zeng et al. (2020), we do not treat the credit computation process, but we focus on the distribution process based on *data provenance*. Data provenance is a form of metadata that describes the origin and life of data. There are several forms of provenance (lineage, why-provenance, where-provenance and how-provenance) with increasing expressive power and complexity, as described in Cheney et al. (2009). Data citation and data provenance are closely linked (Alawini et al., 2018), since both are forms of annotations on data retrieved through queries.

In this work, we rely on lineage (Cui et al., 2000), which is a simple yet effective type of provenance. Lineage is somehow easier to define, understand, and calculate than the other kinds of provenance. However, it proves rather effective in showing the effects of DCD. By using lineage, we provide a first and easily interpretable solution to the DCD issue. As shown in Cheney et al. (2009), lineage can be computed for all kinds of relational operators. In this paper we focus on SPJ queries: a widely used subclass of relational queries where only the operations of selection, projection and join are allowed (Abiteboul et al., 1995).

3. Use case

Our definition of DCD and all experimental analyses are based on a widely used curated database, the IUPHAR/BPS Guide to Pharmacology (Harding et al., 2018), also known as GtoPdb. GtoPdb is a complex and well structured relational database that contains expertly curated information about diseases, drugs in clinical use, their cellular targets, and the mechanisms of action on the human body. The data are drawn from high-quality pharmacological and medicinal chemistry literature. Curated and maintained by the GtoPdb Committee and by its 96 subcommittees, it comprises a total of 512 scientists collaborating with in-house curators. The information about who does what, meaning which scientists curated which data, can be found in the database and employed for citation and credit distribution purposes.

While GtoPdb is relational in nature, its logical structure is hierarchical, as shown in Fig. 2. The root of the hierarchy is the database itself. The information contained in the database is organized into webpages focused on specific diseases, targets or ligands. A single webpage contains information extracted from the database through conjunctive queries. Thus, citing a webpage means citing the queries used to create that webpage.

This paper discusses target families. GtoPdb's subdivision proposes eight types of families: *GPCR*, *Ion channels*, *NHRs*, *Kinases*, *Catalytic receptors*, *Transporters*, *Enzymes* and *Other protein targets*. Each type contains a set of families, and each family is composed by targets, also known as receptors.

As shown in Fig. 2, each node of the hierarchy corresponds to a webpage with its URL. The pages are composed by different sections such as introduction/overview of the page, comments, the list of receptors, genes and proteins. When available, there is also contributors page list, and a section on "How to cite this page" containing a citation text snippet.

All the webpages of the target families share the same structure as shown in Fig. 3 for the "Adenosine receptors" family. For each section, we show the content and SQL code used to retrieve the data from the database that are used to build the section. The sections are: *title*, with the name of the family; *overview*, briefly describing the nature of the receptors of the family; *list of receptors* composing the family, with a short summary of each receptor and a link to its detailed page; *comments* on the family; *further readings*, containing a list of related external papers; the *references*, i.e., the papers used by experts to validate or support the data used in the page; and the list of *subcommittee members* responsible for the family and *contributors* to the family content.

Each family page can therefore be considered as a full-fledged traditional publication, comprising title, authors, abstract (overview), content, and references. Many papers in the literature cite these webpages. What happens is that many papers in

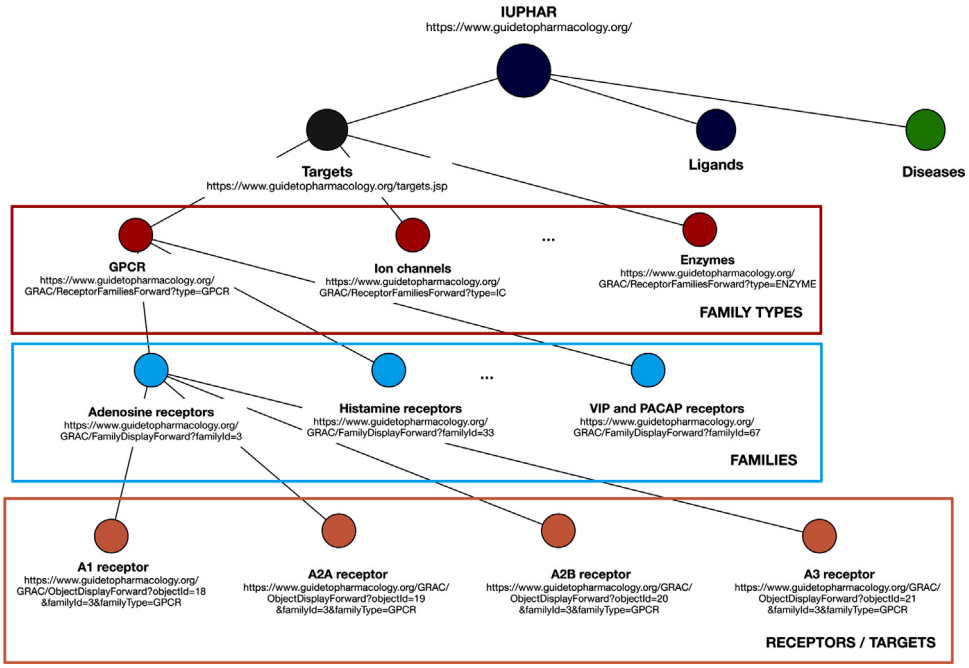

Fig. 2. Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

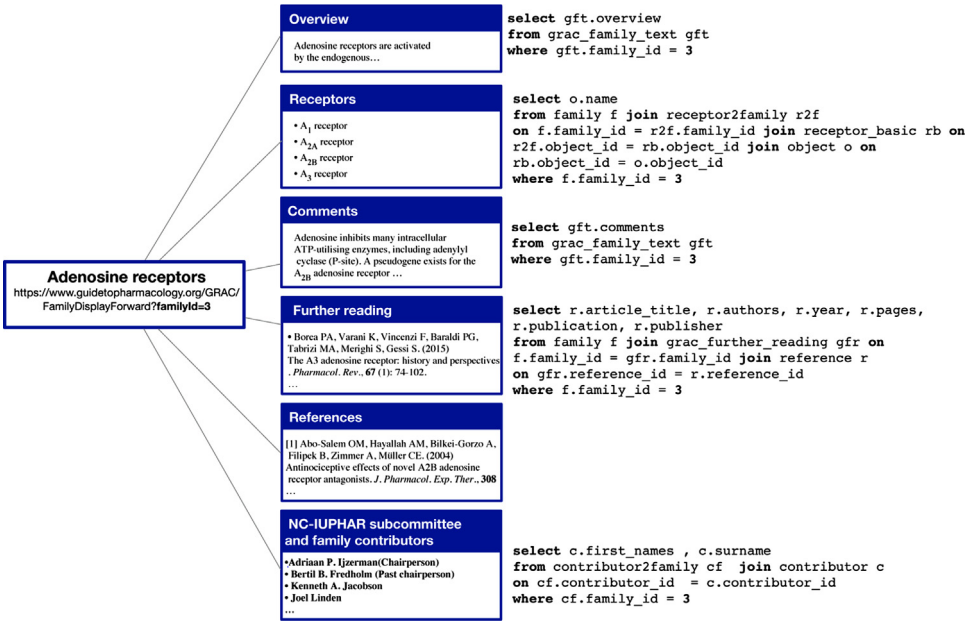

Fig. 3. Basic web-page structure of "Adenosine receptors" family (ID 3), with queries used to retrieve the information contained in every section, except references.

the literature use the GtoPdb's families without including a reference to the specific page about the cited family. They cite the data paper describing GtoPdb as a whole and thus, the citations to the specific families turn out to be "hidden" in the citing papers. In certain circumstances, as in the case of papers published as PDF in the *British Journal of Clinical Pharmacology*[11] (*BJCP*), the family, ligands and receptor names have a hyperlinks pointing to the corresponding webpages in GtoPdb. In this case, the citations to the families can be spotted and counted by using the URLs reported in the papers.

[11] https://bpspubs.onlinelibrary.wiley.com/journal/13652125

Table 1
Example of a database composed by three tables. `family` includes some receptor families in the database; `contributor`, with the name and country of contributors of the database; `contributor2family`, connecting the contributors to the families they contributed to.

	family	
id	Name	Type
f_1	Dopamine Receptors	gpcr
f_2	Bile Acid Receptor	gpcr
f_3	FAK Family	Enzyme
f_4	YANK Family	Enzyme

	contributor2family	
id	family_id	contributor_id
$c2f_1$	f_1	c_1
$c2f_2$	f_1	c_2
$c2f_3$	f_2	c_3
$c2f_4$	f_4	c_1

	contributor	
id	Name	Country
c_1	John Smith	UK
c_2	Jim Doe	UK
c_3	Hans Zimmerman	Germany
c_4	Roberta Rossi	Italy

Nevertheless, all these citations to GtoPdb webpages do not convert into credit for curators and collaborators who produced the data on the cited webpages. In most of the cases, the curators receive no recognition for their work, at least in terms of credit coming from the citations.

For our running example, consider Table 1. This simplified version of GtoPdb illustrates of three relations: `family`, `contributor` and `contributor2family`.

The tuples of the first table represent four families in GtoPdb. Just three attributes are shown: the id, name and family type. The table `contributor` contains the people who have helped to generate the data of the database In the example, four contributors are considered. Finally, the table `contributor2family` serves as a link between the families and the people who contributed to them. For instance, "John Smith" (`c1`) contributed to "Dopamine Receptors" (`f1`) as well as to the "YANK Family" (`f4`).

4. Methods

4.1. Data credit and the data credit distribution problem

Given a database instance, a *recipient of credit* corresponds to a unit of information within the same database. In the case of relational databases, recipients may be: (i) the whole database; (ii) tables; (iii) tuples; (iv) attributes.

Data credit is a value $k \in \mathbb{R}_{>0}$, used as a measure to represent the *value* of a recipient in a database. Within a database, every recipient is annotated with a given quantity of credit, which is a proxy of its importance. This work considers tuples as the recipients of credit.

In general, data credit distribution (DCD) considers a database instance I and a query Q producing a result set $Q(I)$. When Q is cited, a certain amount of credit – that without loss of generality we consider as given – is assigned to the result set $Q(I)$. DCD consists in defining a strategy to split credit into portions to be assigned to the tuples in I. Since tuples are set as our *recipients of credit*, we say that DCD operates at the at *tuple level*.

Worth noting that portions of the credit may be assigned only to tuples in $Q(I)$ – i.e., the direct recipients of the citation – or to the tuples in $Q(I)$ along with other tuples in I that somehow contributed to $Q(I)$. Data provenance is used to determine which tuples in I deserve credit and how much.

In the following, the notation used in Cheney et al. (2009) is applied. Therefore, a *tuple location* is defined as a tuple in one relation of the database tagged with its name. A tuple location is indicated with (R, t), where R is the relation in the database, and t is the tuple in R. With reference to the running example, `(family, ⟨f1, Dopamine Receptors, gpcr⟩)` is the tuple location of the first tuple in the `family` relation. The set of all the tuple locations in I is called *TupleLoc*.

DCD at tuple level is defined as follows.

Definition 4.1 *(Data credit distribution at tuple level (DCD)).* Given a database instance I, a query Q over I and the value $k \in \mathbb{R}_{>0}$, DCD is defined as the computation of the function $f_{I,Q} : \mathit{TupleLoc} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$ such that $f_{I,Q}(t, k) = h$ where $0 \leq h \leq k$ and $\sum_{t \in \mathit{TupleLoc}} f_{I,Q}(t, k) = k$.

Table 2
Result of a SQL query applied on the database of Table 1, asking the names of the families curated by a researcher based in the UK. The id attribute is added to help identify the two tuples. Every tuple is annotated with its lineage.

id	Name	Lineage
o_1	Dopamine Receptors	$\{f_1, c2f_1, c_1, c2f_2, c_2\}$
o_2	YANK Family	$\{f_4, c2f_4, c_1\}$

The process adopts a function to annotate every tuple in the database with a real value as a fraction of a given quantity k. The only constraint of the function is that the sum of the credit fractions annotating the tuples must be k. This implies that in the distribution no credit is created nor destroyed; if credit k is given, we cannot distribute more or less than k to the tuples in I. Such a function is called *data distribution strategy* (DDS).

Given I and Q, many different DDS may be defined as long as they sum up to k. For example, a valid DDS is the function which assigns all the credit to one tuple and gives 0 to all the others. Another valid function uniformly distributes the credit to *all* the tuples in I. Despite the validity of these DDS, their usefulness is questionable since they do not use effective criteria to reward the role of the tuples acting in the computation of $Q(I)$. Hence, to define a useful DDS we recur to data provenance, to lineage in particular.

4.2. Lineage

Lineage was first introduced by Cui et al. (2000) and is an example of *data provenance*. Data provenance is a form of metadata associated to relational values which serves to describe the origin, the life and the process of creation of data.

Lineage associates each tuple $o \in Q(I)$ in the output of a query to a set of tuples in the input. In general, the lineage of one tuple o is a collection of tuples that "contribute" to the creation of o, or that helped to "produce" o (Cheney et al., 2009).

The formal definition, originally presented in Cui et al. (2000), and paraphrased in Cheney et al. (2009) for one relational operator, is the following:

Definition 4.2

Lineage for a relational operator (Cheney et al., 2009)

Let Op be any relational operator over the relations $R_1, \ldots, R_n$. The lineage of a record $o \in Op(R_1, \ldots, R_n)$ is a sequence $\langle R'_1, \ldots, R'_n \rangle$ of subsets $R'_i \subseteq R_i$, such that:

1. $Op(R'_1, \ldots, R'_n) = \{o\}$.
2. $\forall i \in [1, n]$ and $\forall \ \ t \in R'_i$, $Op(R'_1, \ldots, R'_{i-1}, \{t\}, \ldots, R'_n) \neq \emptyset$.
3. $\langle R'_1, \ldots, R'_n \rangle$ is the *maximal* sequence of subsets of $R_1, \ldots, R_n$ which satisfies (1) and (2).

In other words, the lineage is a sequence $\langle R'_1, \ldots, R'_n \rangle$, that is, a set of tuples taken from the input database. This set enables the operation to actually return the tuple o (condition 1). It does not contain useless tuples (condition 2) and it is maximal among all the possible sets that satisfy condition 1 and 2: in the input dataset there are no other tuples that contribute to the production of o (condition 3). It is worthy to point out here that lineage is tuple-based, it is defined for one tuple of the output at a time.

For complex queries, composed of a sequence of more than one relational operator, the lineage is defined *inductively* (Cheney et al., 2009).

Definition 4.3

Lineage for a view (Cheney et al., 2009)

Let $Op_1 \circ \cdots \circ Op_n$ be a relational algebra query, where Op_i, with $1 \leq i \leq n$, is a relational operator. Let I be a database instance; V the resulting output of the query $Op_2 \circ \cdots \circ Op_n$ to I, and o a tuple in V. The lineage of o with respect to $Op_1 \circ \cdots \circ Op_n$ is an $I^* \subseteq I$ such as:

1. I^* is the lineage of a sub-instance V^* of V in I according to $Op_2 \circ \cdots \circ Op_n$.
2. V^* is the lineage of t in V according to Op_1.

To give an idea of what can happen with the lineage of tuples, consider the following SQL query: it asks all the family names in our example database, curated by researchers based in the United Kingdom (UK):

```
SELECT f.name
FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
JOIN contributor AS c ON c2f.contributor_id = c.id
WHERE c.country = 'UK'
```

Table 2 reports the query result. The result is composed of two tuples. The attribute id is added by us to easily identify them.

For tuple o_1, the lineage is the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$, since the tuple f_1 was joined with $c2f_1$ and then with c_1, but also with $c2f_2$ and c_2. For tuple o_2, instead, the lineage is $\{f_4, c2f_4, c_1\}$. As can be seen, the two tuples of the output present different lineages.

4.3. A lineage-based distribution strategy

With the information provided by the lineage, we define the following DDS.

Definition 4.4 *(Lineage-based distribution strategy).* Let I be a database instance, Q a query over I, $o \in Q(I)$ an output tuple and k the credit associated to o. Let L be the lineage of o and t be a generic tuple in I. t receives a credit equal to:

$$f_{I,Q}(t,k) = \begin{cases} 0 & \text{if } t \notin L \\ \dfrac{k}{|L|} & \text{if } t \in L \end{cases}$$

Note that, since lineage is tuple-based, this DDS is not defined for the whole output $Q(I)$, but only for one tuple o at a time. This highlights that if we use a tuple-based provenance like the lineage to perform DCD, one function is not sufficient. We need many CDS, one for each tuple in $Q(I)$. Every output tuple presents its lineage, thus defining its own CDS. Tuples with the same lineage will obviously have the same CDS.

The total credit of the output $Q(I)$ is first divided among the output tuples. Then, a DDS is defined for every tuple to distribute its credit portion. In our paper we assumed that every output tuple carries credit equal to 1.

In our experiments we only focus on SPJ queries. However lineage can also be computed for unions, rename and aggregation queries (Cheney et al., 2009; Wu, Alawini, Deutch, Milo, & Davidson, 2019). Therefore, DCD via lineage can also be performed with these kinds of queries, and not only with SPJ. This lineage-based CDS distributes credit only among tuples that have a role, whichever it is, in the creation of o by the query Q. These tuples receive an equal share of credit.

As an example, consider the output tuples of Table 2. As said, every tuple has credit $k = 1$. The lineage of the first tuple is the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$. Therefore, each tuple in this set receives credit 1/5. The other tuples of the database receive credit 0. The lineage of the second output tuple is $\{f_4, c2f_4, c_1\}$. Therefore each of these tuples receive credit 1/3.

At the end of the process, tuples $f_1, c2f_2, c_2$ all have credit 1/5, tuples f_4 and $c2f_4$ have credit 1/3, while tuple c_1 has credit 8/15. All the other tuples of the database have no credit.

Tuples used together with others must share the credit. More tuples in the lineage means less credit given to each one. If one tuple appears in more lineages, it will accumulate more credit from different sources, implying its relevance in building the output.

Arguably, in the case of o_1, not all of the tuples of its lineage are necessary at the same time for its generation. For example, if the database only had the set of tuples $\{f_1, c2f_1, c_1\}$ or the set $\{f_1, c2f_2, c_2\}$, the existence of o_1 would still be guaranteed. In other words, only one among the couples of tuples $\langle c2f_1, c_1\rangle$ and $\langle c2f_2, c_2\rangle$ is really necessary, while f_1 is absolutely mandatory. In this sense, one could argue that it would be fairer for f_1 to receive more credit than the other four tuples.

This highlights the limitation of the lineage. While able to find all and only the relevant tuples of an output, it is unable to distinguish the *importance* of tuples in the query computations. This information could be incorporated in the definition of a DDS in order to distribute credit based on the actual role that tuples play in the computation.

For this reason, more sophisticated and complex forms of provenances are defined for relational databases, such as why-provenance (Buneman, Khanna, & Tan, 2001) and how-provenance (Green, Karvounarakis, & Tannen, 2007). Such forms of provenance are more complex and difficult to interpret than lineage. Hence, they do not seem to be the best choice to introduce and make DCD understandable. Nonetheless, any type of provenance is suitable to be used for DCD. Provenance is required because simply distributing the credit to the tuples that appear in the final result set obtained by the query presents many limitations. The central one is that it is often the case that many tuples that are used by a query are not visualized in the final result set. It happens, for example, when one tuple A is joined with a tuple B and a further projection operation keeps only attributes of B. Hence, tuple A is not present in the final output, even though it was used by the query. If this strategy was followed, tuple A (and its creators/curators) would not receive any credit.

Moreover, provenance is fundamental also for *aggregation* queries. As an example, these are queries containing a `sum` operator, which sums all the values of one column. We do not specifically treat these types of queries in this paper since they are not SPJ, but it is still possible to compute their lineage (Cheney et al., 2009). These queries produce outputs with data that are *not* contained in the original database since they are the product of a computation on the original data. In this case, the simple strategy to only reward the data that appear in the output does not allow the assignment of any credit, since the output data was never present in the input database in the first place. Lineage, as a form of provenance, deals with all these problems and enables us to define a sensible Distribution Strategy.

5. Experimental evaluation

5.1. Experimental setup: definition of the queries

Using provenance, DCD can be performed on any type of query, as long as its output provenance can be computed. With lineage, DCD can be executed with any type of SPJ query also combined with rename, union and aggregation. Lineages of such queries output can, in fact, be computed (Cheney et al., 2009; Wu et al., 2019).

This paper considers the IUPHAR/BPS Guide to Pharmacology Database (GtoPdb) as use case. As stated above, GtoPdb is a relational curated database which contains information about targets (also called receptors), diseases and ligands. In particular, as shown in Fig. 2, the information of the database is organized on webpages, which, in turn, are hierarchically set up.

To describe GtoPdb's main features, let us focus on targets. Webpages about targets (at the bottom of the hierarchy) are grouped into target families, belonging to different *types* of families. A type can therefore be considered as a set of families. GtoPdb identifies eight family types: *GPCR*, *Ion channels*, *NHRs*, *Kinases*, *Catalytic receptors*, *Transporters*, *Enzymes* and *Other protein targets*.

When a paper uses data from GtoPdb, it can cite the full database, the family Webpage of interest or a subset of data extracted with a query. In this work we consider a full-fledged data citation context in which papers cite the specific *data* subset of interest and not the Webpage or the full database acting as data proxies.

Therefore, when a paper cites a family data, it is actually citing a set of queries needed to retrieve all the information provided by the family webpage, i.e. one query for each section composing a page, as depicted in Fig. 3. The figure maps the structure of one family, "Adenosine receptors", and the queries to obtain the information to build the corresponding page, apart from the list of references. In GtoPdb, all family pages share a similar structure (the only differences may be the presence/absence and length of the receptors lists, further readings and contributors sections). The same queries are therefore used to build all other pages by simply changing the family id (which, in our example, is 3). All these queries are SPJ.

Our evaluation considers two synthetic scenarios where two citation distributions from papers to GtoPdb families were artificially created, and one real scenario where real citations from papers to families were collected.

5.2. Synthetic scenario: uniform distribution

Our first set of experiments considers the case in which the data of every family is cited only once. This is what we call the "uniform distribution scenario". The goal of such experiments is to show how credit distributes itself in the database depending on the queries issued.

We automatically built the queries that create *every* target family contained in GtoPdb, similar to those in Fig. 3. Queries were then divided into eight classes, corresponding to the eight family types in GtoPdb. One type at a time, we executed every query of that type and collected the results. Without loss of generality, we set the credit value k to one (i.e., $k = 1$) for every tuple in a query output. To compute the query lineage, we adopted the system used and shared by (Wu et al., 2019).

Our results, illustrated in Fig. 4, report the heat-maps of three tables of GtoPdb – `family`, `receptor_basic` and `contributor` – for two family types: GPCR and enzymes. The two are among the most common families in GtoPdb. GPCR, as we will see, highlights a high collaboration rate among the authors, i.e. many of the authors who contributed to the families in GtoPdb also contributed to at least one GPCR family. Enzymes type is one of the biggest set among the types of GtoPdb, and enables us to show how this influences the distribution of credit in different areas of the database.

The Table `family` contains basic information on all GtoPdb families, such as id, name and type. Similarly, table `receptor_basic` contains general information about all GtoPdb receptors. Finally, `contributor` contains information about all the GtoPdb contributors, such as names, roles and associations.

Heat-maps can be viewed as matrices, where each cell corresponds to one tuple, and the tuples are organized in a column-major order. Every tuple has a credit value which determines the color of the corresponding cell in the heat-map: the more intense the color of the cell, the higher the credit given to the tuple.

Let us focus on the top part of Fig. 4 related to GPCR. The first thing one notices from the heat-map is that, in the table `family`, only tuples corresponding to GPCR families received credit and are therefore highlighted. Therefore, no other family types are used for the computation of GPCR families. Consequently, when citing GPCR family related data, no other family receives credit.

Nevertheless, it is interesting to note that some GPCR tuples receive more credit than others. This can be explained by considering the queries where the table `family` is used. The queries retrieve: (i) the receptors of one family and (ii) the list of further readings. The longer the two lists for one family, the higher the number of tuples in the queries result and the more the credit that is generated and distributed. Hence, the more receptors and related readings a family possesses, the more credit it should receive.

The above is a direct consequence of the strategy which assigns credit *one* to every output tuple. The queries that generate the greatest output in terms of tuples number are also the most rewarded with credit.

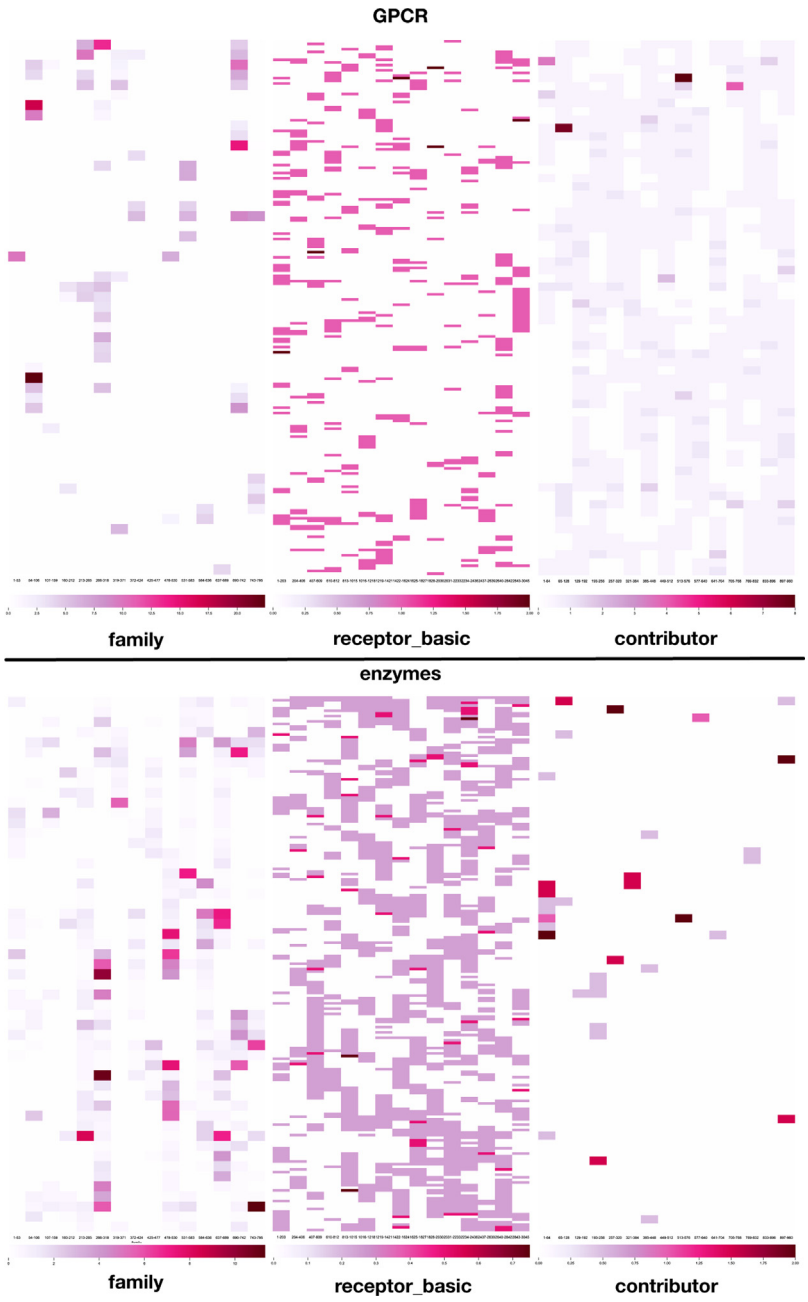

Fig. 4. Heat-maps obtained from three tables (family, receptor_basic and contributor) with a uniform distribution – every family had 1 citation. Two families are represented: GPCR and enzymes.

The heat-map therefore describes the volume of information contained in each family. Families with higher credit have more content. The same is true for the table `family` of the enzyme type. Here, different tuples of the table are highlighted, since different families belong to this type. They are also greater in number than those in the GPCR family. However, if we focus on the credit assigned, GPCR families generally have more credit (e.g., the family with identifier 16 gets a credit of 22), while those of the enzyme family receive less (they never get more than 6). Note that credit values have no meaning in absolute terms but, to establish which family gets more credit, they should be compared.

For the `receptor_basic` table in the GPCR scenario, credit is uniformly distributed over all the tuples that have a role in the GPCR families: the cell color is quite uniform. This means that all these receptors are found in one and only one family, and receive the same quantity of credit only once. In this case, the heat-map reveals which receptors belong to the GPCR families.

Looking closely, six receptors get more credit than the others (i.e., darker cells in the heat-map). These receptors (such as the *GPR*55, identifier 109), belong to two different families (in the case of *GPR*55, family 16 and 114), as they present similarities or traits that make cataloging into only one family difficult. In this sense, the heat-map is also a useful tool to quickly spot receptors that belong to more than one family. The same is true for the Enzymes, where more receptors belong to two or more families.

Let us conclude with the `contributor` table. For both GPCR and Enzymes, the heat-maps highlight the authors that have contributed to the corresponding families. As for GPCR, many more authors contributed to the pages of this family than the page on Enzymes, since almost all the tuples in the table have non-zero credit. However, some contributors get more credit than others. In particular, the three cells with intense color show the contributors with high credit since they contributed to more than one family, receiving credit from more than one query. The heat-map can be regarded as an indicator of how much work a contributor has done within the GPCR families. For instance, one author is a curator in sixteen different families, therefore obtaining the highest credit. The Enzymes case is similar, with fewer authors contributing to Enzymes than to GPCR, but on average having a higher credit. As can be seen in fact, there are many white cells (without credit), but several have an intense color. This means that fewer authors work on Enzymes, and they often work together sharing the same quantity of credit. Moreover, the overall credit for Enzymes is lower than for GPCR even though there are more pages. This means that there is less information on the pages, and therefore less credit to be distributed.

The synthetic scenario also illustrates how credit distribution highlights different parts of the database that belong to different regions, corresponding to different families. Moreover, based on the queries used for credit distribution, different heat-maps can be interpreted specifically. In the case of `family`, the uniform distribution of credit allows us to find pages that contain more raw information (in our case, more receptors and related works). In `receptor_basic`, the receptors belonging to more families can be identified. While in `contributor`, we identify the authors who contributed to a specific topic, and to what extent.

5.3. synthetic scenario: Pareto distribution

The second synthetic scenario expands the real-world case where a few resources are cited several times while many others are never cited or just once. A Pareto distribution is used to generate the number of times each family is cited (therefore the number of times the corresponding queries are issued). Every time a query is issued, corresponding credit is distributed.

The adopted Pareto distribution is implemented in the Apache commons library.[12] The probability distribution of the random variable x is defined as follows:

$$f_X(x) = \begin{cases} \dfrac{\alpha k^{\alpha}}{x^{\alpha+1}} & x \geq k \\ 0 & x < k \end{cases}$$

where k is the *scale parameter* and α is the *shape parameter* or also *tail* parameter. The scale parameter defines the minimum values produced by the distribution with a non-zero probability. Tail parameter describes the shape of the distribution. The higher α, the faster the distribution converges to zero (the less probable it is to obtain higher values).

For every family a value x is generated from the distribution, and we considered $\lceil x \rceil$ as the number of citations received by that family. We set $k = 0.9$ and $\alpha = 0.95$. k guarantees that every page has at least one citation (so there are not families with zero citations). With $\alpha = 0.95$ a fair and contained number of families received many citations. We set thirty as citation threshold for a single family. Therefore, every time the distribution returned a value higher than thirty, it was cut to thirty.

After computing the citation numbers, we distributed the credit as many times as indicated. For instance, if a webpage receives five citations, the queries returning the data on the family were used five times in credit distribution.

Fig. 5 shows the heat-maps obtained with this distribution. Starting from the GPCR family, the maximum level of credit reached by one tuple is much higher (more than 140, against the value 20 obtained with the uniform distribution) as many more queries were issued in this case.

According to Pareto distribution, only certain families receive many queries, the rest obtain only few citations. We still see all the GPCR-related tuples highlighted in the `family` table as in the uniform distribution case (Fig. 4). Yet, many of them now belong to the lower-part of the spectrum, whereas a few get high credit value.

Credit assumes another role using Pareto distribution, since it is more sensitive to the information of the tuples in the database. In fact, the tuples that correspond to information cited more frequently may be considered more important in this context, and are rewarded with more credit than the others. The quantity of credit they are annotated becomes a proxy of their importance, as recognized by the research community.

Similar observations are true for the other tables in the figure. In particular, `receptor_basic` does not present a "uniform" credit distribution among the tuples as before. The Pareto distribution rewards certain receptors more than others, depending on their relation to the families that obtain a higher reward. In this simulation, credit is therefore given to receptors that the research community considers more important.

[12] https://commons.apache.org/proper/commons-math/javadocs/api-3.4/org/apache/commons/math3/distribution/ParetoDistribution.html

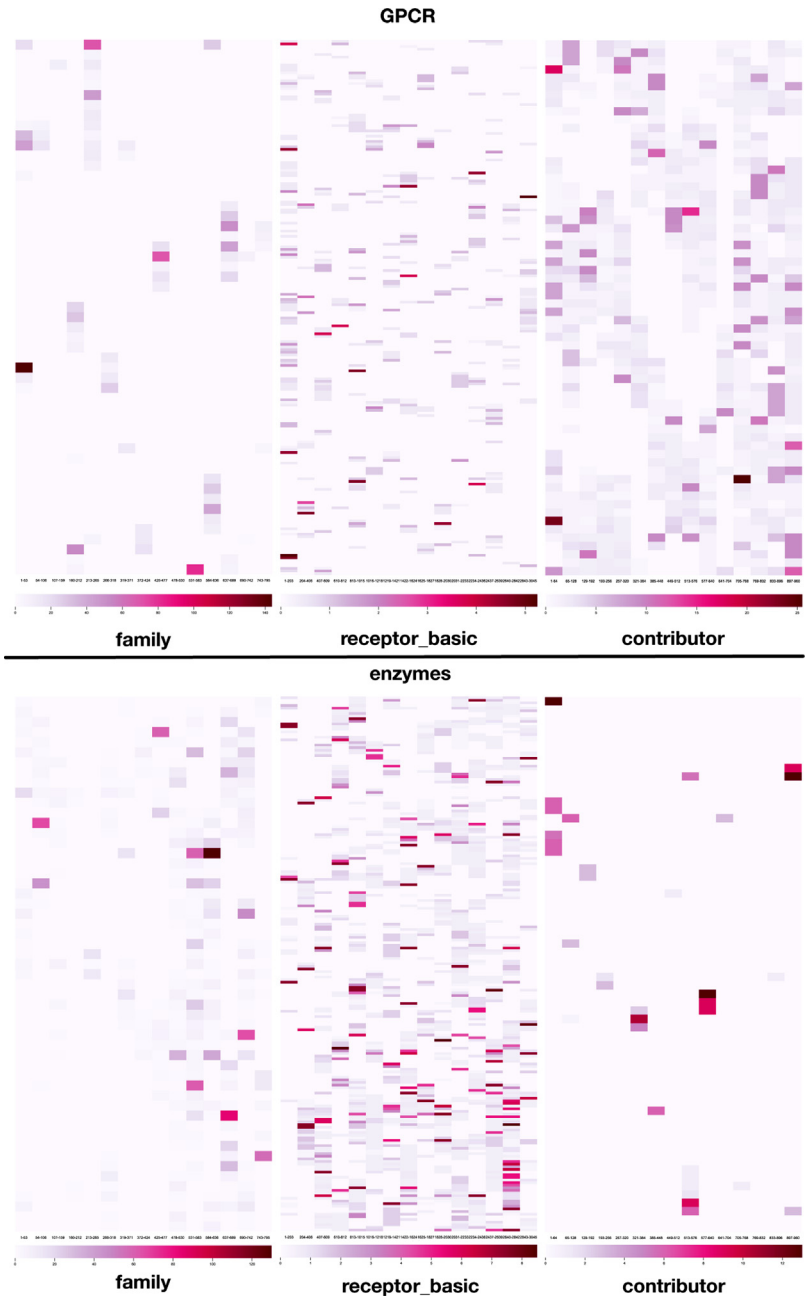

Fig. 5. Heat-maps obtained from three tables (family, receptor_basic and contributor) with a Pareto distribution of scale 0.9 and shape 0.95. Two families are represented: GPCR and enzymes.

Similarly, for the `contributor` table, there is a change in distribution compared to the one obtained through uniform distribution in Fig. 4. Authors are not rewarded for their role in creating certain types of pages, but for the recognition, on the part of the research community, of their work. Since their names appear on the pages they have curated, the more a page is cited, the more the authors are rewarded. The authors that receive more credit are not necessarily those with more pages in their curriculum, but the ones whose pages were used more, thus producing more credit by the community. With credit distribution, some authors are credited even though the data they have helped to generate is not cited directly, but used to produce the cited data.

This new distribution allows us to highlight how credit, depending on the situation, can be used to infer different information on the roles of tuples and agents within a database.

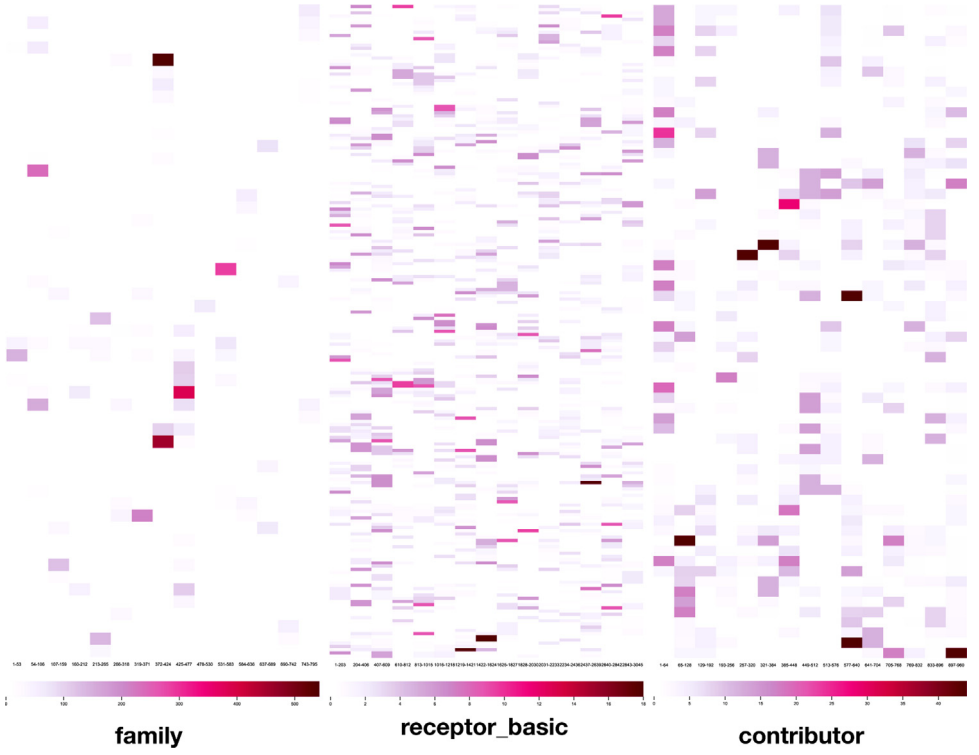

Fig. 6. Heat-maps obtained from three relations (family, receptor_basic and contributor) with the distribution produced from the papers citing GtoPdb 2018.

5.4. Real scenario: distribution generated from papers

As already stated, many papers that draw information from the GtoPdb website[13] cite papers published every two years by the GtoPdb Committee on Receptor Nomenclature and Drug Classification (NC-IUPHAR). To obtain a set of citations capable of representing what actually happens, we consider a paper subset citing the 2018 GtoPdb (Harding et al., 2018) data paper. At the time of writing this paper received more than 900 citations.

As explained in Section 3, in the papers published in the *British Journal of Clinical Pharmacology* (*BJCP*), which cite GtoPdb, the name of families are hyperlinks that point to the corresponding webpages in GtoPdb. We considered all the 460 papers in BJCP citing (Harding et al., 2018) as of February 2020. URL references to family pages were automatically extracted to guide in building the queries to produce corresponding webpages. A total of 5945 different queries were built in this way. [14]

Fig. 6 shows heat-maps obtained by credit distribution performed using the queries described above. As it can be seen, the families are not treated separately, since the queries are citing pages from all eight family types of GtoPdb. Heat-maps are therefore representing the distribution of credit defined by the research community that cited the latest available version of GtoPdb.

Focusing on `family`, it is apparent that our experiment with the Pareto distribution is actually rather accurate in depicting the credit distribution trend in the real-world. The distribution in Fig. 6 shows a few tuples with a high credit and many others with almost no credit. A very similar one was obtained with Pareto distribution and it seems that the α parameter is even higher in reality than in our synthetic hypotheses; the real citation distribution is more skewed than in the synthetic case we created. This means that in the real case fewer tuples compared to the one of the Pareto distribution receive almost all the credit.

In the `contributor` table very few authors receive high credit, as they correspond to the more cited families in the considered papers. Since these pages receive most of the credit, the corresponding tuples in `contributor` also receive a great quantity of credit.

Credit therefore adapts to the different ways in which citations are distributed, becoming an effective and immediate tool to find the tuples highly used and valued by the research community.

[13] https://www.guidetopharmacology.org

[14] For reproducibility purposes, the code we used for our experiments and all the produced queries can be found at the following link: https://bitbucket.org/dennis_dosso/credit_distribution_project.

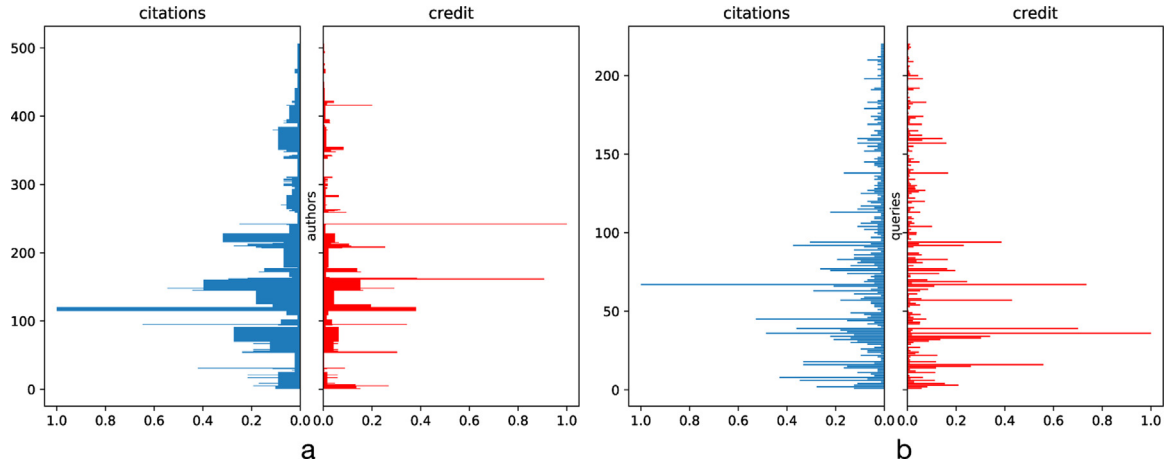

Fig. 7. Bar plots representing the quantity of credit and citations respectively received by the authors of virtual documents and by virtual documents. All the quantities were normalized between 0 and 1.

Our last set of experiments still uses real-world distribution from BJCP, now changing the level of granularity. So far we have considered data at tuple level, i.e. every tuple was a potential recipient of credit. Now, every family in GtoPdb is considered as one full-fledged paper. As already mentioned, this is a sensible choice because every GtoPdb family is represented as a webpage with the characteristics of a standard publication. The pages of GtoPdb are therefore now considered as publications of a journal, which is the whole database. We call these publications *virtual papers* and consider them as recipients of credit.

Every time a paper from BJCP refers to one GtoPdb family, that reference becomes a citation to the corresponding virtual paper. Every time a family is cited, the virtual paper receives a new citation and all the credit generated by the set of queries used to create the corresponding webpage.

Since every family has its set of contributors, the latter are considered as the authors of the virtual paper. This implies that whenever a virtual paper is cited, every author receives one citation. Every time some credit is allocated to a paper, that credit is *equally distributed* to the authors. Up until now, credit was allocated solely to the data, but nothing prevents us to assign it to the authors of that data as well. This possibility was already raised in Katz (2014). In this case, we assume that every author equally contributed to one family. Such process differs from that of credit distribution to the tuples in the `contributor` table in Figs. 4–6. In those cases, authors correspond to tuples in a table used by a query. They therefore receive credit being part of a computation. In this case we are closer to traditional citations among papers. First we attribute credit to a virtual paper of one family and that same credit is equally distributed to the authors of that paper.

In this experiment the number of citations and the quantity of credit given to each virtual paper and to each author have been counted. Dividing by the highest value of citations and credit obtained by an entity (a paper or an author), we normalized and compared them.

Fig. 7 shows a first comparison between the two metrics. Fig. 7.a represents the authors on the y-axis (more than 500 for all the GtoPdb families), and number of citations and of credit on the x-axis. As we can see, there is a non-negligible correlation between the two measures and it is easily comprehensible: the more one author is cited, the more s/he will receive credit. The linear correlation among citations and credit for authors is 0.62, with a p-value of 2.07×10^{-54}; while Kendall's tau correlation is 0.75, with a p-value of 2.06×10^{-124}. This confirms a high correlation between citation and credit distributions.

A similar observation seems true for the virtual papers in Fig. 7.b. The more a paper is cited, the more credit it receives. In this case, the linear correlation between the two is 0.74, with a p-value of 3.05×10^{-39}; the Kendall's tau correlation is 0.59, with a p-value of 1.55×10^{-41}.

Fig. 8 shows how credit can highlight certain aspects of an author's role whereas the information generated by citations cannot. A radar plot illustrates the top 10 authors, ranked by the number of citations they received from papers citing GtoPdb, together with their credit (all values are normalized between 0 and 1). The identifiers in the figure are randomly given to the authors to anonymize them. In the case of papers, they correspond to the identifiers of the respective families.

The first six authors in Fig. 8(a) (who worked together in the same family) receive most of the citations, while the others just a few. However, even these particularly successful authors do not get a very high credit, since it must be divided among themselves. Additionally, the quantity of credit generated from their citations is not that high enough to guarantee a top ranking credit position.

This key aspect distinguishes credit distribution from citations counting. In fact, as already stated, these experiments have the underlying assumption that every tuple of the output carries is assigned with credit equal to one. This implies that the larger the query result ($|Q(I)|$), the higher the credit that will be distributed. Credit is therefore also sensitive to the quantity of information cited, not only for it being cited or not. The more one author is cited, the more credit s/he receives.

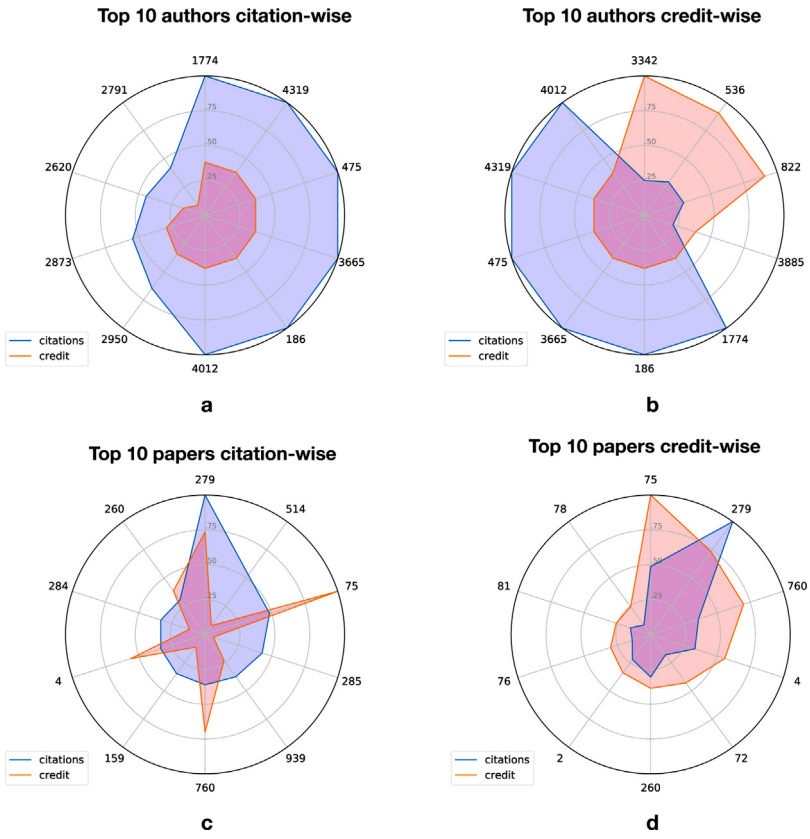

Fig. 8. Radar plots representing the top 10 authors by their citations together with their credit (a); the top 10 authors by their credit together their citations (b), the top 10 virtual papers by their citations, together with their credit (c) and the top 10 virtual papers by their credit together with their credit. All values are normalized between 0 and 1.

But this is not all. If an author contributes to a query output that carries a lot of credit, s/he will receive more credit than an author who contributes to data that carries less credit.

Moreover, the more an author shares the effort with other authors, the less credit s/he will receive. Therefore authors who worked individually without sharing their effort receive more credit, as reward for their effort.

Fig. 8(b) shows the top-10 credited authors, who are not necessarily also the top-10 cited authors. Credit in such case recognizes the authors role in work which is cited fewer times, but pivotal in terms of results and impact within the reference community. This type of crediting gives a different point of view on the roles and contributions of authors.

Similar observations can be made for Fig. 8c and d where we consider the top-10 virtual papers in terms of citations and credit. In Fig. 8c the top virtual documents do not always have the highest credit. These documents correspond to families with less information, i.e. are composed of less tuples, thus generating less credit, despite the many citations.

On the other hand, as shown in Fig. 8d, virtual papers with high credit do not necessarily get the highest number of citations. These papers correspond to families with many tuples, which therefore generate a lot of credit every time they are cited.

6. Discussion

6.1. More characteristics of credit

While the experiments disclosed the many characteristics of credit and their difference from citations, some critical aspects should nonetheless be pointed out. Due to the conditions in which these experiments were set, in particular the nature of GtoPdb and the queries we used, some aspects of credit may not be evident as shown below:

1. There may be cases where a query output displays only part of the information which is used by the query itself. Certain query operators may project out information contained in tuples which were used for joins. Another case may refer to aggregations, such as the `sum`. In this way, many tuples may be used, but none displayed to the user nor cited. Credit can therefore reward information which is not *visible* in the query output. If it were not for credit distribution, this information and its authors would never been rewarded.

2. Credit can be assigned through different strategies. For instance, a particularly useful tuple for the results of a citing paper can bring more credit than a tuple which is only mentioned as an example. This means that credit not *only* reflects the quantity of work which is used, but also how it is used. This is not the case with citations, since every paper in the reference list of a citing paper always receives one citation, no matter how it is used. Credit instead can be adapted to reflect the role of data within a citing paper.
3. One output tuple might be generated using many different tuples, thus its lineage is quite big. Using the DDS proposed in this paper, the tuples of the lineage equally share the credit. The bigger the cardinality of the lineage of an output tuple, the less credit the single tuples receive. This may not be the best outcome, in particular when the output tuple is the result of a complex query and has an important role in the citing source. In this paper, we assumed $k = 1$ for each output tuple simply to achieve a relative simplicity in our discussion, without deviating too much from the main focus, i.e. the distribution process. The decision of the quantity of credit carried by one tuple is a problem by itself, one which is not discussed in this paper, since it presents various aspects that have their own research dignity. A solution to this problem can be achieved by modifying the quantity k of credit carried by a specific output tuple to more accurately reflect its importance in the context of the query.

6.2. About other types of databases

We focused on credit distribution on curated RDBs. Even so research is also carried out by using data stored in other forms of databases, such as RDF and XML databases. Moreover, on one of its latest report, the RDA WG on Data Citation reported that there is an enormous amount of CSV data in the world of research that represent a challenge for Data Citation (RDA Data Citation Working Group, 2020). CSV files and spreadsheets are not databases in a strict and formal sense; however, they do not represent a particularly difficult challenge. A CSV file, or more generally a spreadsheet, could be seen as one unique relational table, where the rows are the tuples and the columns represent the single attributes. A query over a CSV file is usually a selection of rows and columns, which can be easily converted into one or more SQL queries composed by selection and projection operators. Therefore, the proposed DCD methodology can be easily adapted to work with CSV files.

For RDF databases it is often the case that they are stored in relational databases composed of one unique table, made of three attributes: subject, predicate, and object. Other, more sophisticated solutions are also adopted, which still rely on relational databases (Weiss, Karras, & Bernstein, 2008). In this case, a SPARQL query (the standard structured query language for RDF datasets (Pérez, Arenas, & Gutierrez, 2009)) can still be converted, through careful transformations, in an equivalent SQL query on a relational database (Rodriguez-Muro & Rezk, 2015). This still provides all the means to distribute data credit. This case is thus less straightforward with respect to CSV files, but the implementation of DCD on RDF databases is still possible without developing specific techniques.

It is also the case that there are other curated RDF databases that are exposed to the users through Webpages, like DisGeNET[15] (Queralt-Rosinach, Pi nero, Bravo, Sanz, & Furlong, 2016), Eagle-i[16] or UniProt[17] . For instance, DisGeNET is a database containing information about genes and the diseases that they cause and, while it exposes a SPARQL end-point where users can directly query the database, it also can be accessed through a web interface and search and browse functionalities. One example of a DisGeNET Webpage, similar in structure to the ones of GtoPdb, is for example https://www.ncbi.nlm.nih.gov/medgen/C0027651 for “Neoplasm”. This makes this database very similar to GtoPdb in this regard. DisGeNET also presents an SQLite and CSV version, making even easier to implement our strategy using what is already proposed in the present work.

For citations to XML databases, if they follow the RDA guidelines (RDA Data Citation Working Group, 2020), it is possible to get the queries (e.g. a WQuery) corresponding to the citations and the version of the database where the query was computed. Thus, two of the main components of DCD are present: the database and the query. What is missing is a form of data provenance, such as the lineage. Since there are forms of lineage for XQuery on XML databases (Steiger, 2010), and others can be produced, it is actually possible to map DCD on XML datasets. The basic mechanism and guidelines of DCD presented in the paper remain the same, and they can be followed to implement new Distribution Strategies on different types of databases.

This means that, no matter the type of the database, it is still possible to extend and implement DCD to other databases with a relative small effort once the basic guidelines expressed by the RDA WG (or also from the CODATA (CODATA-ICSTI Task Group on Data Citation Standards & Practices, 2013) and FORCE 11 (Martone, 2014) initiatives) are followed.

6.3. On the variability and quality of data

One of the main differences between data and traditional papers is the higher variability of datasets. Data are updated more frequently than published papers, and thus may present many different versions. Citations to datasets are harder to be dealt with since it is not always evident how and if a citation should be transferred from one version to the other. On the

[15] www.disgenet.org/
[16] https://www.eagle-i.net/
[17] https://www.uniprot.org/

other hand, credit, being assigned to single tuples, can be "propagated" from one version of the dataset to another by merely maintaining the credit to the tuples that were not updated, and designing functions to update the credit assigned to tuples that were changed.

Also, as noted in Mayernik, Callaghan, Leigh, Tedds, and Worley (2015), the introduction of new methods for peer-review of datasets can increase the trustworthiness and the value of individual datasets and strengthen research findings. However, there are still questions about the right place of data in the value chain of scholarship, at what point in their life cycle the review should occur, and what concretely means to peer review data (Borgman, 2010; Mayernik et al., 2015). There is no clear, shared, and universally accepted way to peer-review data that can guarantee the quality of data itself. There are not ranking systems for data venues, e.g., repositories. Thus we cannot infer the quality of a database from where it is published. Data credit, together with data citations, can work as an indirect means to assess the quality of data. The impact a particular dataset, or part of it, may function as an indicator of how much the research community is finding that data reliable or authoritative.

6.4. Limitations

DCD is a new tool built on top of the mechanisms of Data Citations. More precisely, to be implemented effectively, DCD needs: a database, a query, its output, and the provenance of the output.

In the current world of research it is often the case that researchers do not follow an optimal data citation behavior. These misbehaviors may manifest themselves in different ways, and cause different problems in the DCD pipeline and more in general to data citation.

One example is when authors publish their data in local repositories, which are not accessible or do not follow the data citation guidelines. Another example is when researchers publish their data as unstructured files that cannot be queried. In these cases there is not much that we can do to perform DCD because the data are in the first place not properly citable.

In other cases, when writing a paper researchers do not cite data or limit themselves to cite the whole database instead of the data subset that they actually used. Without a query it is not possible to perform Credit Distribution. However, it is often the case, as shown in Section 5.4, that useful information can be inferred from the citing paper to build the original query producing the used data. In the case of the BJCP papers we were able to extract the URLs of the GtoPdb Webpages and, with them, reverse-engineer the queries that build the corresponding Webpages. In other cases it can still be possible to infer the queries from other types of information, such as keywords contained in the text. This strategy can be applied to all the papers citing the databases discussed in Section 2 which follow the RDA specifications.

More importantly, the more data citation becomes a shared practice, the more the queries used by researchers are accessible. This will also pave the way for DCD and make these inference practices less and less necessary.

Our work is placed in a series of efforts toward a better understanding and adoption of data citation practices by research communities; we focused on life science communities and data because in this realm data citation is more present than in other domains such as the humanities.

7. Conclusions and future works

This paper defines data credit distribution (DCD) for relational databases. Given a database instance I and a query Q, DCD takes as input its result set $Q(I)$ and a real positive value k called credit. Fractions of k are distributed to data in I which plays a role in generating $Q(I)$. Since this work considers relational databases, queries are written in SQL. When dealing with about data, we refer to tables and tuples of a database.

A methodology to solve this problem has been defined through the use of lineage, that is a kind of data provenance method for relational databases. The lineage method helps define a function called credit distribution strategy (CDS), to solve DCD. The CDS proposed herein equally distributes credit to all tuples that have a role in generating a query result set.

We have designed a system that implements the CDS function and tested it on GtoPdb, a widely used curated relational database. GtoPdb provides data via a website with a hierarchical organization where each webpage represents a "family". In turn, such content is generated by issuing a number of queries on the database. Citing a family webpage is a proxy of citing the output of queries used in generating the family webpage.

Different groups of queries were identified to select data from the database. These queries identify families of receptors in the dataset and all related information (name, overview, list of receptors, references, comments, curators). We then used these queries to simulate different DCD scenarios: two synthetic and one based on real data citations.

In the first synthetic scenario (i.e., uniform distribution), we consider that every GtoPdb family is cited one and only one time, meaning that all the queries used to create family webpages are issued only once.

The paper shows how credit can highlight parts of the database that cover a certain topic instead of another.

The second synthetic scenario simulates the case of cited webpages following a Pareto distribution. In this typical "power law" condition, a few resources are highly cited, while many others have limited or no citations. In this scenario, credit rewards the tuples that correspond to more frequently issued queries, and to the tuples that correspond to queries that generate high volume of data.

The real scenario considers a set of papers belonging to the *British Journal of Clinical Pharmacology* that cite families in GtoPdb. Every time a query is used in a given paper, we count a citation going from such paper to the data generated by

the query. More than five thousands citations from papers to GtoPdb families were collected. We then distributed the credit generated by the citations to the tuples of the database and to the related authors.

Even though credit and citations count are correlated measures, credit behaves differently from citation count, offering new perspectives to evaluate the impact of both data and authors. Credit has different applications depending on the way it is employed. When a uniform distribution is applied, it can highlight parts of the database related to certain "topics". Database "hotspots" can be highlighted when tuples are particularly used by a set of queries.

Differently from classical counting of citations, credit directly rewards all tuples (and their authors) that contribute to produce cited data, even those that are not in the output itself. Moreover, it can be used to proportionately reward tuples with a prominent role in a set of queries. In fact, the more a tuple is used, the more it is rewarded.

In this paper, we assumed that every output tuple carries credit equal to one as a starting point for the distribution. Different assumptions and algorithms for the computation of the credit on the output tuples are possible. For instance, these algorithms may take into account the role of cited data in a paper and weight credit differently among output tuples.

What emerges is that the more an information is useful for a paper, the more credit it produces and distributes. Hence, credit takes the impact of cited data into account.

Credit can also be distributed to the authors of the cited entity, be it paper or data. When citing a paper, every author of that paper receives one citation. Credit, instead, can also be distributed among the authors of the cited entity, emphasizing their specific role in that entity.

In this work, we decided to equally distributed credit to the authors of a cited entity. But, even in this case, different strategies can be followed. It is however true that the more authors a cited entity has, the more the credit will be shared, therefore reducing their individual payoff. Having shared the effort in producing a final output, they will be less rewarded than the author that alone is responsible for a paper or some data.

We have shown that credit differs from citations in this context as well. Certain authors receive more credit than others with more citations because their cited contributions generate more credit, rewarding them more.

All of the above suggests that credit can be an effective new tool, used alongside traditional paper and data citations, to better understand the role and impact of cited entities and authors in the literature. Credit can be the base of new bibliometrics, which not only take into account whether a research entity is cited or not (as in traditional citations), but also specifies: (i) the role of the cited entity in the citing ones; (ii) *all* the elements that contributed in generating the cited entity, not only the *visible* ones in the entity itself; and, (iii) the authors' roles in producing the information that generated the cited entity.

Future work will focus on developing new data distribution strategies using different kinds of data provenance, in particular why and how-provenance. They are more sophisticated forms of provenance, and take into account why and how data was used when generating information. With such additional information, credit distribution will undoubtedly be richer, more sophisticated and potentially more fair.

While this paper was centered on a tuple level, our plan is to also work at the attribute level. To do so, new forms of attribute-based provenance, such as where-provenance (Buneman et al., 2001), will be tested and developed.

We also intend to explore different databases, like UNIPROT, which contains cases of data citing papers and data citing data. Graphs where papers and data can interchangeably cite one another can therefore be designed. Potential paths, through which credit can be transitively transmitted, can also be created to study the propagation of the influence of a database or a paper within a chain of citations, as already hypothesized in other papers.

Moreover, we will study new possibilities for bibliometrics based on credit uses in tandem with more classical measures such as impact factor and h-index. New bibliometrics can therefore lead to better recognition of the role of data and authors in the scientific literature.

Author contributions

Dennis Dosso: Conceived and designed the analysis, collected the data, contributed data or analysis tools, performed the analysis, wrote the paper.

Gianmaria Silvello: Conceived and designed the analysis, wrote the paper.

Acknowledgments

This work is partially supported by the Computational Data Citation (CDC-STARS) project of the University of Padua and by the ExaMode project, as part of the European Union Horizon 2020 program under Grant Agreement no. 825292.

The authors would like to thank Yinjun Wu, for providing the code in Wu et al. (2019) to compute provenance. Part of this research has been carried out when Dennis Dosso was visiting the University of Pennsylvania under the supervision of Susan Davidson.

The authors wish to warmly thank the anonymous reviewers for the useful suggestions that helped us to improve the paper.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.joi.2020.101080.

References

Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bernstein, P., et al. (2020). The Seattle report on database research. *ACM SIGMOD Record*, *48*(4), 44–53. http://dx.doi.org/10.1145/3385658.3385668

Abiteboul, S., Hull, R., & Vianu, V. (1995). . *Foundations of databases* (Vol. 8) Reading: Addison-Wesley.

Alawini, A., Davidson, S. B., Hu, W., & Wu, Y. (2017). Automating data citation in CiteDB. *Proceedings of the VLDB Endowment*, *10*(12), 1881–1884.

Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., & Wu, Y. (2018). Data citation: A new provenance challenge. *IEEE Data Engineering Bulletin*, *41*(1), 27–38.

Altman, M., Borgman, C. L., Crosas, M., & Martone, M. (2015). An introduction to the joint principles for data citation. *Bulletin of the Association for Information Science and Technology*, *41*(3), 43–45.

Asmi, A., Rauber, A., Pr“oll, S., & van Uytvanck, D. (2016). Citing dynamic data-research data alliance working group recommendations. *EGU general assembly conference abstracts (Vol. 18)*,

Baggerly, K. (2010). Disclose all data in publications. *Nature*, *467*(7314), 401.

Ball, A., & Duke, M. (2011). *How to cite datasets and link to publications*. Digital Curation Centre.

Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D., Bhagat, J., et al. (2013). Why linked data is not enough for scientists. *Future Generation Computing Systems*, *29*(2), 599–611.

Belter, C. W. (2014). Measuring the value of research data: a citation analysis of oceanographic data sets. *PLOS ONE*, *9*(3), e92590.

Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., & Lamere, P. (2011). *The million song dataset*.

Borgman, C. L. (2010). *Scholarship in the digital age: Information, infrastructure, and the Internet*. MIT Press.

Buneman, P. (2006). How to cite curated databases and how to make them citable. *18th international conference on scientific and statistical database management, SSDBM*, 195–203.

Buneman, P., Cheney, J., Tan, W.-C., & Vansummeren, S. (2008). Curated databases. *Proc. of the 27th ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, PODS 2008*, 1–12. http://dx.doi.org/10.1145/1376916.1376918

Buneman, P., Davidson, S. B., & Frew, J. (2016). Why data citation is a computational problem. *Communications of the ACM*, *59*(9), 50–57.

Buneman, P., Khanna, S., & Tan, W. C. (2001). Why and where: A characterization of data provenance. *8th International conference om database theory – ICDT 2001*, 316–330.

Buneman, P., & Silvello, G. (2010). A rule-based citation system for structured and evolving datasets. *IEEE Data Engineering Bulletin*, *33*(3), 33–41.

Burton, A., Koers, H., Manghi, P., Stocker, M., Fenner, M., Aryani, A., et al. (2017). *Scholix metadata schema for exchange of scholarly communication links*. Geneva, Switzerland: CERN.

Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N., Kirsch, P., et al. (2012). making data a first class scientific output: Data citation and publication by NERC's environmental data centres. *International Journal of Digital Curation*, *7*(1), 107–113. http://dx.doi.org/10.2218/ijdc.v7i1.218

Candela, L., Castelli, D., Manghi, P., & Tani, A. (2015). Data journals: A survey. *Journal of the Association for Information Science and Technology*, *66*(9), 1747–1762.

Cheney, J., Chiticariu, L., & Tan, W. (2009). Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, *1*(4), 379–474.

CODATA-ICSTI Task Group on Data Citation Standards and Practices. (2013).. *Out of cite, out of mind: The current state of practice, policy, and technology for the citation of data* (Vol. 12) http://dx.doi.org/10.2481/dsj.OSOM13-043

Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., & Simons, N. (2019). Bringing citations and usage metrics together to make data count. *Data Science Journal*, *18*(1).

Cronin, B. (1984). *The citation process. The role and significance of citations in scientific communication*. London: Taylor Graham.

Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evidence of a structural shift in scholarly communication practices? *Journal of the American Society for Information Science and Technology*, *52*(7), 558–569.

Cui, Y., Widom, J., & Wiener, J. L. (2000). Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems*, *25*(2), 179–227.

Davidson, S. B., Deutch, D., Milo, T., & Silvello, G. (2017). A model for fine-grained data citation. *8th biennial conference on innovative data systems research – CIDR 2017*, www.cidrdb.org

Dubernet, M. L., Antony, B. K., Ba, Y. A., Babikov, Y. L., Bartschat, K., Boudon, V., et al. (2016). The Virtual Atomic and Molecular Data Centre (VAMDC) Consortium. *Journal of Physics B: Atomic, Molecular and Optical Physics*, *49*(7), 074003. http://dx.doi.org/10.1088/0953-4075/49/7/074003

Fang, H. (2018). A discussion of citations from the perspective of the contribution of the cited paper to the citing paper. *Journal of the Association for Information Science and Technology*, *69*(12), 1513–1520.

Force, M., Robinson, N., Matthews, M., Auld, D., & Boletta, M. (2016). Research data in journals and repositories in the web of science: Developments and recommendations. *Bulletin of IEEE Technical Committee on Digital Libraries*, *12*(1), 27–30. Special Issue on Data Citation.

Gößwein, B., Miksa, T., Rauber, A., & Wagner, W. (2019). Data identification and process monitoring for reproducible earth observation research. *2019 15th international conference on eScience (eScience)*, 28–38.

Green, T. J., Karvounarakis, G., & Tannen, V. (2007). Provenance semirings. *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems*, 31–40.

Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson, A. J., Ireland, S., et al. (2018). The IUPHAR/BPS guide to PHARMACOLOGY in 2018: Updates and expansion to encompass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Research*, *46*(Database-Issue), D1091–D1106.

Hartley, J. (2017). Authors and their citations: A point of view. *Scientometrics*, *110*(2), 1081–1084.

Honor, L. B., Haselgrove, C., Frazier, J. A., & Kennedy, D. N. (2016). Data citation in neuroimaging: Proposed best practices for data identification and attribution. *Frontiers in Neuroinformatics*, *10*, 34.

Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., et al. (2005). Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research*, *33*(Database-Issue), 428–432. http://dx.doi.org/10.1093/nar/gki072

Katz, D. (2014). Transitive credit as a means to address social and technological concerns stemming from citation and attribution of digital products. *Journal of Open Research Software*, *2*(1).

Kosten, J. (2016). A classification of the use of research indicators. *Scientometrics*, *108*(1), 457–464.

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

Longo, D. L., & Drazen, J. M. (2016). Data sharing. *The New England Journal of Medicine*, http://dx.doi.org/10.1056/NEJMe1516564

Martone, M. (2014). *Joint declaration of data citation principles. FORCE11*. San Diego, CA: Data Citation Synthesis Group. http://dx.doi.org/10.25490/a97f-egyk https://www.force11.org/datacitationprinciples

Mayernik, M. S., Callaghan, S., Leigh, R., Tedds, J., & Worley, S. (2015). Peer review of datasets: When, why, and how. *Bulletin of the American Meteorological Society*, *96*(2), 191–201.

Meho, L. I., & Yang, K. (2007). Impact of data sources on citation counts and rankings of LIS faculty: Web of Science versus Scopus and Google Scholar. *Journal of the American Society for Information Science and Technology*, *58*(13), 2105–2125.
Mooney, H., & Newton, M. P. (2012). The anatomy of a data citation: Discovery, reuse, and credit. *Journal of Librarianship and Scholarly Communication*, *1*.
Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., et al. (2015). Promoting an open research culture. *Science*, *348*(6242), 1422–1425.
Pérez, J., Arenas, M., & Gutierrez, C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, *34*(3), 1–45.
Peters, I., Kraker, P., Lex, E., Gumpenberger, C., & Gorraiz, J. (2016). Research data explored: An extended analysis of citations and altmetrics. *Scientometrics*, *107*(2), 723–744.
Pröll, S., & Rauber, A. (2013). Scalable data citation in dynamic, large databases: Model and reference implementation. *Proceedings of the 2013 IEEE international conference on big data*, 307–312.
Queralt-Rosinach, N., Pi nero, J., Bravo, À., Sanz, F., & Furlong, L. I. (2016). DisGeNET-RDF: Harnessing the innovative power of the semantic web to explore the genetic basis of diseases. *Bioinformatics*, *32*(14), 2236–2238.
Rauber, A., Ari, A., van Uytvanck, D., & Pröll, S. (2016). Identification of reproducible subsets for data citation, sharing and re-use. *Bulletin of IEEE Technical Committee on Digital Libraries*, *12*(1), 6–15. Special Issue on Data Citation.
RDA Data Citation Working Group. (2020). *Mtg@p15*. https://rd-alliance.org/system/files/documents/RDA-DC-Recommendations_151020.pdf
Rodriguez-Muro, M., & Rezk, M. (2015). Efficient SPARQL-to-SQL with R2RML mappings. *Journal of Web Semantics*, *33*, 141–169.
Silvello, G. (2018). Theory and practice of data citation. *Journal of the Association for Information Science and Technology*, *69*(1), 6–20.
Spengler, S. (2012). Data citation and attribution: A funder's perspective. In N. A. Information (Ed.), *Report from developing data attribution and citation practices and standards: An international symposium and workshop of sciences' board on research data* (pp. 177–178). Washington, DC: National Academies Press.
Steiger, B. (2010). *Data lineage/provenance in XQuery (Master's thesis)*. ETH Zurich, Systems Group.
Weiss, C., Karras, P., & Bernstein, A. (2008). Hexastore: Sextuple indexing for semantic web data management. *Proceedings of the VLDB Endowment*, *1*(1), 1008–1019.
Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific Data*, *3*.
Wu, Y., Alawini, A., Davidson, S. B., & Silvello, G. (2018). Data citation: Giving credit where credit is due. *Proceedings of the 2018 international conference on management of data*, 99–114.
Wu, Y., Alawini, A., Deutch, D., Milo, T., & Davidson, S. B. (2019). ProvCite: Provenance-based data citation. *Proceedings of the VLDB Endowment*, *12*(7), 738–751.
Zeng, T., Wu, L., Bratt, S., & Acuna, D. E. (2020). Assigning credit to scientific datasets using article citation networks. *Journal of Informetrics*, *14*(2).
Zou, C., & Peterson, J. B. (2016). Quantifying the scientific output of new researchers using the zp-index. *Scientometrics*, *106*(3), 901–916.
Zwölf, C. M., Moreau, N., Ba, Y. A., & Dubernet, M. L. (2019). Implementing in the VAMDC the new paradigms for data citation from the research data alliance. *Data Science Journal*, *18*(1).
Zwölf, C. M., Moreau, N., & Dubernet, M.-L. (2016). New model for datasets citation and extraction reproducibility in VADMC. *Journal of Molecular Spectroscopy*, *327*, 122–137.