Credit Distribution through Data Provenance in Relational Scientific Databases

Dennis Dosso[a], Susan B. Davidson[b], Gianmaria Silvello[a]

[a]Department of Information Engineering, University of Padua, Italy
[b]Department of Computer and Information Science, University of Pennsylvania, United States

Abstract

In the current world of research data is a fundamental method to disseminate scientific knowledge, to determine scholarship, and to provide credit and recognition to the authors of research endeavors. However, issues like data citation, handling and counting the credit generated by such citations are still open research questions.

In this context, data credit has recently emerged as a new measure of value, defined and built on top of the data citation theory. Data credit is a real value that represents the importance of data cited by a paper, or by another research entity. As such, credit can be used to annotate data contained in curated scientific databases, and it can be considered as a measure for their importance and impact in the research world. As such, it is a new method that, together with traditional citations, helps to recognize the value of data and its creators in a world more and more dependent on data.

In this paper we explore the problem of Data Credit Distribution, the process by which credit is divided and assigned to the data in a database that are responsible for the production of data being cited by a research entity.

We adopt as use case the IUPHAR/BPS Guide to Pharmacology (GtoPdb), a curated and well-known scientific relational database. We define two new distribution strategies, functions that perform this task, based on two form of data provenance, why-provenance, and how-provenance.

Using different distribution strategies, we show how credit can highlight areas of a database that are frequently used, and how it can work as a new bibliometric measure for data and their corresponding curators. Credit in particular rewards data and authors based on their research impact, and not merely on the number of citations. Also, we show how different distribution strategies, based on different types of data provenance, can be more sensible to the role of an input tuple in the generation of the output, and thus rewarding it differently.

Keywords: Data Citation, Data Credit

---

1. Introduction

Citations are an essential component of scientific research, enabling research products to be found as well as the relationships between research products to be understood. They form a basis on which to give credit to authors, papers, and venues [55, 19, 20]. Citations are used, among other things, to decide on tenure, promotion, hiring, and funding of grants for researchers [41, 21, 32, 38].

Nowadays, science and research are increasingly digital. There are numerous curated databases that are at the core of scientific research efforts [12]. It is therefore generally accepted that data must be cited and citable [39, 15], and that data citations should contribute to the scientific reputation of researchers, scientists, data curators, and creators [4, 50]. It is also accepted that data citations should be counted alongside of traditional citations, and contribute to bibliometrics indicators [7, 44].

A central problem in data citation is how to attribute credit to data creators and curators [11]. How to handle and count the credit generated by data citation, and how it contributes to traditional and new bibliometrics, are long-standing research issues Garfield [28], Borgman [9]. However, even when correctly applied, data citations and the bibliometric computed using them do not always correctly reward the creators of data used in a database. Data, in fact, is often cited at the "database level" or the "webpage level". In the first case, the whole database is cited and therefore all credit goes to the key personnel of the database. In the second case, the database has a website with webpages that can be individually cited. The webpages use data extracted from the database, which is aggregated by topic and built to resemble a traditional research paper. Often the creators and curators of the webpage's data are not credited or only marginally credited for their work [3].

Recently, the concepts of *data credit* and *Data Credit Distribution* (DCD) [26, 36, 54] have emerged, built on top of methodologies for data citation. Data credit is a value that is computed based on the importance of the data being cited in a paper, and represents the impact of the data on the citing paper. The Data Credit Distribution problem consists of distributing this credit to elements in the databases in the citation graph that are responsible for the generation of the data being cited. The goal of DCD is to improve and expand the reach of data citation, rather than being an alternative to it. This means that to employ DCD techniques, we need data citations in some form.

[37] defined credit as a "quantity" that describes the importance of a research entity, such as papers or data mentioned in a citation, and proposed the idea of a *distribution* of credit from research entities, such as papers or data, to other research entities through citations. This can be done by exploiting the structure of the *citation graph*, a directed graph whose nodes are publications and edges are citations. This graph is the model at the core of systems such as Google Scholar and the Web of Science. Zeng et al. [54] and Fang [26] further explored this concept by defining frameworks for the computation and distribution of credit between papers, authors, and data used by papers in the citation graph.

In this paper, we consider data credit as a data value measure in a (curated) scientific database; credit can be assigned to data of any kind and at any level of granularity. Therefore the concept of "data" is left intentionally vague, although in this paper we focus on relational databases. Credit is a positive *real* value, acting as a proxy for the value of data based on the measure of citations, accesses, clicks, downloads, or other surrogates for data use. We call Data Credit Distribution the process, method, or algorithm used to assign credit to a given datum or dataset.

The DCD problem differs from the traditional citation setting since:

1. In a traditional setting, when a paper cites another paper, a +1 "credit' is given to the cited paper (and to its authors). It does not matter why or how paper p_1 cites paper p_2[1], the result is always +1 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. Hence, we can weigh the importance of the cited entities and assign credit according to their role.

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

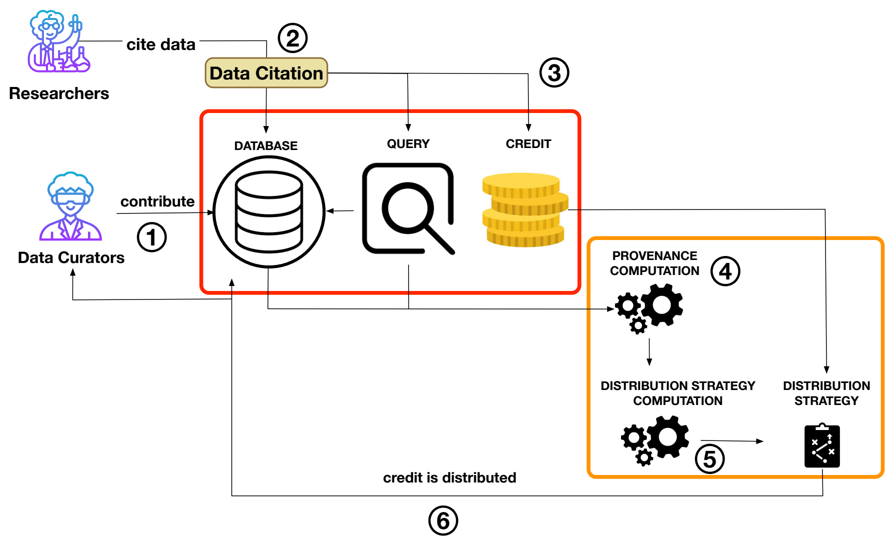

Figure 1: Overview of the credit distribution pipeline.

2. Traditional citations are considered to be *atomic*. A 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 that 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. In contrast, we consider data credit to be a *non-atomic* real value, which can be divided and distributed to multiple components of a database.
3. Credit can be *transitive*, that is, it can be propagated through one cited entity to other entities cited by it that contributed to its content.

We study the DCD problem in the context of relational databases (RDBs) since they are widely used [2] and are the main focus of current work in data citation methods [14, 12, 45]. RDBs are also frequently a test-bed for new methods that can be adapted to other databases, e.g., graphs or document databases. Furthermore, the "portions" of data in an RDB that can be credited can be defined at different levels of granularity, in particular: (i) the whole database, (ii) tables, and (iii) tuples.

The DCD process is summarized in Figure 1:

[2] The "relational database market alone has revenue upwards of $50B" [1].

Step 1 Scientists and experts contribute the curated information contained in a scientific database. These are called the "Data Curators".

Step 2 Other researchers use the data in their research, and when possible, cite them.

Step 3 The citation to the data generates credit, that can be used as a proxy for the impact of the data on the citing paper. This credit is represented as a real value $k \in \mathbb{R}_{>0}$.

Step 4 Given the database instance I and the query Q, it is possible to compute the *data provenance* of $Q(I)$. The provenance of $Q(I)$ is a form of metadata that describes the generation process undertaken by Q, and the data used in I to generate the output [17]. Many different notions of provenance have been proposed in the literature for data in database management systems [22, 13, 30], describing different kinds of relationships between data in the input and the output of a query. As reported in [17], these provenances have been used in several applications beyond giving information on how queries work, for example, annotation propagation and the view update problem. In this paper, we consider three types of provenance: lineage, why-provenance, and how-provenance.

Step 5 Provenance is input to the CDC problem, whose aim is to compute the *Credit Distribution Strategy* (CDS, also referred only as Distribution Strategy, DS). The CDS is a function that distributes k to the data in the input database I, and is defined on the basis of citation policies decided at the database administration level or at the domain community level. In this paper, since we base CDS on data provenance, we describe three CDS, each one based on a different form of provenance.

Step 6 Once the CDS is computed, it is used to distribute the given credit k to the parts of the database that are responsible for the generation of $Q(I)$. Transitively, this credit is also divided and given to the corresponding authors of those data.

This paper expands our recent work in [24], which addressed the problem of how to reward data and data curators who are typically overlooked in current citation systems. In that work, we first defined the problem of DCD in relational databases, and proposed a viable Distribution Strategy (DS) based on *lineage*, which is the simplest form of *data provenance*. The lineage of a tuple t in the output $Q(I)$ is defined as the set of all and only the tuples in the database instance I that are "relevant" to the production of t, that is the tuple that are used by Q in the production of t. The lineage-based strategy equally redistributes the credit k to the tuples in the lineage set, thus each tuple receives credit $k/|L_t|$, where L_t is the lineage set of t.

One may argue that this DS is too simplistic, since lineage only tells the relevant tuple used to produce the output, and does not convey any information about their role or importance in the query. Therefore, one may desire to give more credit to the tuples that are more relevant or *essential* to the production of the output, i.e. those tuples that, if removed, would prevent the output tuple from appearing in the final result, or those tuples used more than once by the query.

Therefore, in this paper, we expand the ideas in [24] by proposing two new DSs based on other forms of data provenance: why-provenance [13] and how-provenance [30]. We compare them with the lineage-based solution, and discuss why one may be preferred to another depending on the application and its goals. In particular, we show that why-provenance and how-provenance are more sensitive to the *role* of a tuple in a query, i.e. how many times the tuple is used and how it is used. The DS based on why-provenance give more reward to tuples that are essential to the production of the result set, whereas the DS based on how-provenance also takes into consideration the different ways that a tuple is used.

For evaluation, we use a well-known curated database, the IUPHAR/BPS[3] Guide to Pharmacology [31], also known as GtoPdb[4], 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.

We perform three sets of experiments. In the first one, real queries are ex-

[3]International Union of Basic and Clinical Pharmacology/British Pharmacology Society

[4]`https://www.guidetopharmacology.org/`

tracted from papers published in the British Journal of Pharmacology (BJP), that represent data citations to GtoPdb, and are used to distribute credit in the database using the three different provenance-based DSs. In the second and third experiment we analyse the behaviour of the different DS when complex citation queries are employed.

Contributions. Contributions of this work include:

- The definition of new distribution strategies for the problem of Data Credit Distribution, based on why-provenance and how-provenance;
- An in-depth analysis of the effects of credit distribution on real-world curated data and of the differences between the three proposed Distribution Strategies.

Outline. The rest of the paper is organized as follows: Section 2 presents the background and related work. Section 3 describes the use case we adopted. Section 4 briefly presents the forms of provenance used in the paper. Section 5 describes the problem of DCD and the proposed DS. In Section 6 we present the experimental evaluation. Finally, Section 7 draws some conclusions and outlines future work.

2. Background

Data in Research. As described by Jim Gray in his last talk [33], the world of research is rapidly transitioning towards the *fourth paradigm of science*, that is, data-intensive scientific discovery, where data are important for scientific advances as well as for traditional publications [6].

The scientific community is promoting an *open research culture* [43], founded on methods and tools to share, discover, and access experimental data. The community has identified the FAIR principles (Findable, Accessible, Interoperable, and Reusable) [52], that should be enforced by every database. In particular, data should be accessible from the articles, journals, and papers that cite or use them [19]. Aspects such as the need for the *reproducibility* of experiments through the used data; the *availability* of scientific data; the *connections* between data and the scientific results are all needed aspects for the fourth paradigm, and are all relevant to the domain of *data citation* [34].

Data Citation: Principles and Motivations. Data Citation principles were first described in detail in [18], and later summarized and endorsed by the Joint Declaration of Data Citation Principles (JDDCP) [40]. The principles are divided into two groups [48]. The first one contains principles concerning the role of data citation in scholarly and research activities such as the (i) *importance* of data (why data citation is important and why data should be considered as first-class citizens); (ii) *credit* and *attribution* to the creators and curators of the data; (iii)*evidence*; (iv) *verifiability*; and *interoperability*, with these last three requiring data citation methods to be flexible enough to operate through different communities. The second group defines the main guidelines to establish a data citation systems, and contains principles such as the (i) *unique identification* of the data being cited; (ii) *(open) access* to data; (iii) guarantee of *persistence* and *availability* of citations even after the lifespan of the cited entity; the (iv) *specificity* of a citation, i.e. it must lead to the data set originally cited.

It is possible to outline six main motivations for data citation [48]:

- *Data attribution*: identify the individuals that should be credited for data with variable granularity.
- *Data connection*: connect papers to the data being used.
- *Data Discovery*: citations helps to find data records and subsets that would be otherwise not findable via search engines.
- *Data Sharing*: share data obtained by researchers within the whole community.
- *Data Impact*: highlight the results obtained in writing papers using specific data, the frequency and modality data were used.
- *Reproducibility*: data citation greatly impacts the reproducibility of science [5]. Many authoritative journals ask to share data and provide valid methodologies to reproduce experiments.

2.1. Data Citation in Relational Databases

In this paper, we develop our methods and experiments on relational databases. RDBs have been the main target of data citation methods since the surge of the data-centric research paradigm. The RDA "Working Group on Data Citation: Making Dynamic Data Citable"[5] [46] has been working in the last years on large, dynamic, and changing datasets. The working group has finished the development of its guidelines and has now moved on into an adoption phase. The datasets considered by the WG are often relational.

In one of its most recent sessions [47], 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) [29]; LNEC (River dam monitoring); MDS (Million Song Database) [8]; CBMI[7] (Center for Biomedical Informatics); VMC (Vermont Monitoring Cooperative); CCA[8] (Climate Change Center Austria); VAMDC (Virtual Atomic and Molecular Data Center) [25, 56].

More examples of work on data citation in relational databases are [12, 53, 2, 23]. 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 compliant with the most recent guidelines, and they are in the vast majority of cases accessible via dynamically created Webpages. In all these databases is, therefore, possible to implement DCD on top of the existing infrastructures for citing data.

Data citation techniques are primarily applied to relational databases 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 even an attribute. Many papers [10, 12, 2] 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, [53] suggested decomposing the database in a set of views, where each view is associated with its citation.

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 pa-

[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`

pers" can be cited, being traditional papers that describe a database [16]. In this case, all the credit from the citations goes to the database administrators or to the authors of the data papers.

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 [46]. In this case, the credit generated from citations can be distributed among the contributors of the portions of data being cited, 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 [35], the GtoPdb [31], and the VAMDC [56]. Every single Webpage is unequivocally identifiable and can be individually cited.

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) [27] is an exception, since its infrastructure tracks 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 does not count citations to database portions such as views, tables, or tuples.

2.2. Data Credit

Data credit is related to data citation: they both aim to recognize the work of data creators and curators. Data credit can therefore also be seen as a by-product of data citation, since credit attribution is impossible without the presence of data citations.

[36] 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. In the paper two considerations are made: (i) research objects such as data and software are currently not formally rewarded or recognized by the community; (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 even more true for data, where different groups of people work on the same database.

In [36] credit is defined as a "quantity" that describes the importance of a research entity, such as papers, software, or data, mentioned in a citation.

We add that the concept of credit can be built on top of the existing infrastructure handling traditional and data citations. [36] further explores the idea of a *distribution* of credit from research entities (i.e., papers and data) to other research entities through citations that connect them. Thanks to traditional citations and now also to data citations, this distribution is finally possible, at least between papers and data. Some problems related to traditional citations can thus be solved by citations:

1. Credit rewards research entities that to date are not (formally) recognized (a goal shared with data citation).
2. Credit can reward authors *proportionally* to their role in generating the entity. The more an author contributes to a paper, the more credit is given to him. [55] work on something similar with their zp-index, which includes in its formulation the position (and thus the role) of a publication author to represent its impact in the work itself.
3. Credit can be *transitively* channeled through a chain of papers citing each other, thus enabling the rewarding of older papers **that are no more cited, since other papers summarize or report their content. Gianmaria: I do not understand this token, what do you mean with: papers that are no more cited?** but are nevertheless crucial in a research area for the influence of their content.

[26] 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 among 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 including both papers and data.

[54] proposes the first method 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 the first step towards an automatic credit computation procedure. This proposal is, however, limited to assigning credit to whole datasets, and it does not deal with the granularity of data. It does not work to assign credit to a single research entity within a dataset. Differently from [54], we do not treat the credit computation process, but we focus on the distribution process.

2.3. Data Provenance

To distribute credit, we base our methods on *data provenance*. Data provenance is information that describes the origin and the process of creation of data. It can also be seen as metadata pertaining to the derivation history of the data. It is particularly useful to help users to understand where data are coming from, and the process they went through. Data citation and data provenance are closely linked [3] since both are forms of annotations on data retrieved through queries. Data provenance has been widely studied in different areas of data management. In this paper, we focus on provenance for database management systems (DBMS). For further details on data provenance, please refer to surveys like [17] and [49].

[17] presents four main types of data citation for DBMS: *lineage* [22], *why-provenance* [13], *how-provenance* [30] and *where-provenance* [13].

Let us start with the first three provenances. Given a database instance I, a query Q, and the result $Q(D)$, consider one tuple t of the output. Its provenance is information about its generation through the tuples of the input that are used by Q. Different types of provenance convey different levels of information. Since these three provenances are computed for each tuple of the output, they are also referred to as *tuple-based*.

Lineage is somehow the simplest among the forms of provenance. It has been defined in different ways [17], but it can be thought of as the set of all the tuples that are used in some way by the query to produce the output tuple, the ones that are somehow *relevant* to its generation.

The definition of why-provenance is based on the notion of *witness set*. A witness is a set of relevant tuples that guarantees the existence of t in $Q(D)$. The lineage is therefore an example of a witness. The why-provenance of a tuple t is a peculiar set of witnesses – described in [13] – that are computed from the query, called *witness basis*. A witness basis may be composed of more than one witness. Therefore, the why-provenance contains more information than the lineage, since it describes *alternative* ways in which the same output may be generated.

The how-provenance takes the form of a polynomial, called *provenance polynomial*, where the variables are taken from the set of identifiers of the tuples (provided that each tuple in I has an identifier) and the coefficients are taken from $\mathbb{N}$. This provenance also contains information on *how* the input tuples are used. For example, when two tuples are combined by a join, they are also combined in the polynomial by the $\cdot$ operator. When two or more

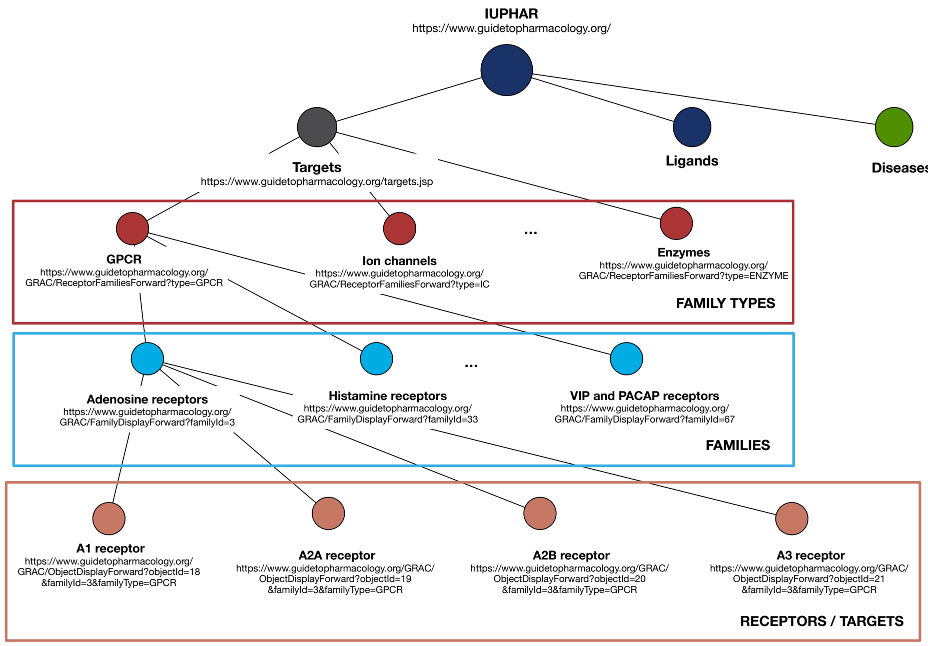

Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

tuples become equivalent due to a union or a projection, the corresponding monomials are combined by the + operator.

It has been shown in [17] that the how-provenance is the more general and informative of the three, containing the other two.

Where-provenance, differently from the other three, is *attribute-based*, so we do not take it into account in this work since we consider the tuple as the finest citable unit.

3. Use Case: GtoPdb

As use case we refer to the IUPHAR/BPS Guide to Pharmacology [31] or GtoPdb[9]. GtoPdb is a well-known and well structured scientific 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. It is curated and maintained by the GtoPdb Committee, and

[9] `https://www.guidetopharmacology.org/`

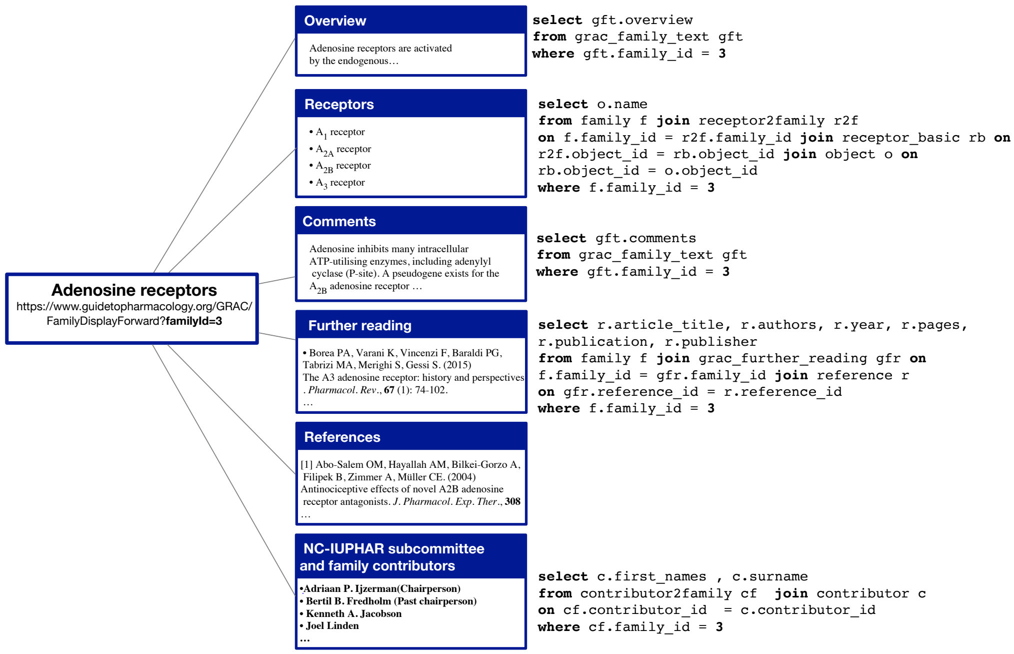

Figure 3: Basic web-page structure of "Adenosine receptors" family (ID 3), with queries used to retrieve the information contained in every section, except references.

by 96 subcommittees, comprising 512 scientists collaborating with in-house curators who draw the information contained in the database from high-quality pharmacological and medicinal chemistry literature. Roughly 1000 researchers from all over the world have contributed to the database, and the curators wanted to give recognition to these contributors. This led to some early work on data citation [10].

GtoPdb is relational, but its logical structure is hierarchical as shown in Figure 2. The information contained in the database is also organized into webpages focused on specific diseases, targets or ligands, and families for easier access by users. As depicted in Figure 2, the database can be thought of as a tree where the root is the database; the first level consists of all targets, ligands, and diseases; and the lower levels consists of specific targets, ligands and diseases. In this paper, we focus on targets; thus at the third level in the figure we show examples of family types, at the fourth level we show specific families of targets (a finer level of granularity), and finally, at the last level, the single targets (also known as receptors).

GtoPdb provides access to the webpages corresponding to all these nodes through URLs. The webpages corresponding to target families all present a similar structure, as shown in Figure 3 for the “Adenosine receptors” family. Each page has an *Overview*, a brief text describing the content of the page; a list of *Receptors* comprising the family; a section of *comments* about the family; the *References*, a list of the papers consulted by the curators of the page, similar to a reference list of a paper; the *further reading* list, reporting papers that an interested reader may want to consult to obtain more insight on the family; and a final section called *How to cite this family page*, containing text snippets useful to cite the specific page or the whole database. Figure 3 shows the SQL code that retrieves the information used to build the corresponding sections (apart from the References section). Therefore, each family page can be considered a full-fledged traditional publication, consisting of title, authors, abstract (the overview), content, and references.

In practice, many papers in the literature only reference GtoPdb (the root) without including a reference to the specific page being cited. That is, they only cite a paper describing GtoPdb as a whole (e.g., [31]) and refer to targets, ligands, diseases, etc. only by name. Thus, citations to specific families are *de-facto* “hidden” to citation systems such as Google Scholar, and useless for the computation of bibliometrics.

In certain “lucky” cases, as with papers available in PDF and published in the British Journal of Clinical Pharmacology [10] (BJCP), when a family, ligand, receptor name, etc. are used, they have a hyperlink pointing to the corresponding webpage in GtoPdb. Therefore, the citations to the families can be detected and counted using the URLs reported in the papers. However, these citations to GtoPdb webpages are not counted as such by citation systems, so they are not converted into credit for curators and collaborators.

For our running example, consider Table 1. This simplified version of GtoPdb illustrates three tables: `family`, `contributor` and `contributor2family`. The first table, `family`, has tuples representing families with three attributes: the id of the family, its name, and type. Table `contributor` consists of people who have helped generate the data of the database. The third table, `contributor2family`, serves as a link between the families and the people who contributed to them. For instance, “John Smith” ($\mathtt{c}_1$) contributed to “Dopamine Receptors” ($\mathtt{f}_1$) as well as to the “YANK Family” ($\mathtt{f}_4$). We use this example throughout the rest of the paper. In particular, we are using

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

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

Table 1: Example of a database consisting of three tables. `family` includes some receptor families in the database; `contributor` contains the name and country of contributors; `contributor2family` connects contributors to the families they contributed to.

the `id` attribute of the tables as *provenance token* of its corresponding tuples, that is, as a symbol that serves to identify a tuple when talking about provenance.

4. Data Provenances

In this section, we present the three types of provenance used in this paper: lineage, why-provenance, and how-provenance.

4.1. Lineage

Lineage was first introduced by Cui et al. [22]. Given a database instance I and query Q, lineage associates with each tuple $o \in Q(I)$ the set of tuples in the input that helped "produce" it [17]. As an example, consider the following SQL query `Q1`, applied to the database described in Table 1, that asks for the names of families curated by researchers based in the United Kingdom (UK):

```
Q1: SELECT DISTINCT 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’
```

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

Table 2: Result of an SQL query applied to the database instance in Table 1, which asks for the names of families curated by a researcher based in the UK. Attribute `id` is not part of the output and was added to succinctly identify each tuple as provenance token. Each tuple is also annotated with its lineage.

Table 2 shows the query result, which consists of two tuples. We add an extra attribute `id` so that we can easily refer to each result tuple. The lineage for tuple o_1 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, and was also joined with $c2f_2$ and c_2. No other tuple is used in the database to produce o_1. For tuple o_2 the lineage is $\{f_4, c2f_4, c_1\}$. Lineage is defined for each tuple of the output, and can differ between tuples.

4.2. Why-Provenance

Why-Provenance was first defined in terms of a deterministic semistructured data model and query language [13]. While why-provenance can be defined in many ways, we refer to [17], where it is expressed in terms of the relational model using the relational algebra.

In particular, while lineage aims to find all and only the tuples in the input relevant to the production of an output tuple, why-provenance aims to find sub-instances of the input that “witness” a part of the output. Given a tuple t in the query’s output, a *witness* is any sub-instance of the database that produces t. In particular, the whole database and the lineage of t are both witnesses of t. Since the definition of witness allows for the presence of “irrelevant” tuples, the set of all witnesses is finite (since the database instance I is finite), but it is potentially exponentially large [17].

Buneman et al. [13] defined the why-provenance of an output tuple t in the result $Q(I)$ as a special *subset* of the set of witnesses called the *witness basis*. The witnesses of the basis depend on Q; thus, each basis’s size is bounded by the size of Q. The witnesses of the basis exclude tuples that are irrelevant to t being produced by Q, and thus the basis tends to be very small compared to the set of all possible witnesses [17]. The witnesses are also *minimal*, in the sense that if one tuple is removed from one of these witnesses, it cannot produce the output.

id	name	why-provenance
o_1	Dopamine Receptors	$\{\{f_1, c2f_1, c_1\}, \{f_1, c2f_2, c_2\}\}$
o_2	YANK Family	$\{\{f_4, c2f_4, c_1\}\}$

Table 3: Result of a SQL query applied on the database of Table 1 with the why-provenance of the corresponding results.

In a sense, each witness in the witness basis captures one possible way in which the query can generate the output. To better understand this, consider the example in Table 3, where each tuple in the result of query `Q1` is annotated with its why-provenance.

The why-provenance of output tuple o_2 has only one witness, which coincides with its lineage. This happens because there is only one way this output tuple can be produced, i.e., for tuple f_4 to be joined with $c2f_4$ and c_1. On the other hand, o_1 has a witness basis with of two witnesses, since there are two possible ways in which the query can generate o_1. One possibility is that f_1 is joined with $c2f_1$ and c_1 (the first witness), and the second possibility is that f_1 is joined with $c2f_2$ and c_2 (the second witness). This means that to generate o_1, it is sufficient that only one of the two witnesses is present in the input database.

4.3. How-Provenance

While why-provenance describes the source tuples that witness an output tuple in the result of the query, it leaves out information about how the source tuples are used. How-provenance was therefore defined in [30] to capture this information using a *semiring* algebraic structure, and is a form of provenance that takes the form of a *polynomial*.

The key idea in Green et al. [30] is to use the two operators $+$ and $\cdot$ to represent two basic transformations that source tuples undergo as a result of applying a relational query to a database [17]. Two tuples may either be joined together, as an effect of a join (represented with the $\cdot$ operator) or merged via union or projection (represented with the $+$ operator).

Table 4 shows a simple example in which the two output tuples of our running example are annotated with their respective how-provenances. Tuple o_2 was produced through the join among the input tuples $f_4, c2f_4$, and c_1. The three provenance tokens are, therefore "multiplied" together. The case of o_1 is slightly more complex. This tuple, as already discussed, can be obtained through two different joins. The two monomials composing the polynomial

id	name	how-provenance
o_1	Dopamine Receptors	$f_1 \cdot c2f_1 \cdot c_1 + f_1 \cdot c2f_2 \cdot c_2$
o_2	YANK Family	$f_4 \cdot c2f_4 \cdot c_1$

Table 4: Result of the example SQL query `Q1` with the corresponding how-provenances of the output tuples annotated.

represent these two alternatives. They correspond, in a way, to the witnesses of the why-provenance of o_1. The $+$ operator represents the fact that the two monomials describe alternative derivations. The output tuple is the result of a merge of two distinct tuples after the projection on the attribute `name`. This merge is due to the fact that the result of a relational algebra expression is always a *set* of tuples, which corresponds to the presence of the `DISTINCT` operator in an SQL query. This simple example gives the basic idea behind how-provenance and how it allows us to track the operations that produced an output tuple.

Provenance polynomials may also have monomials whose exponents and/or coefficients are greater than one, for example, $3f_1 \cdot c2f_1 \cdot c_1 + f_1 \cdot c2f_2^3 \cdot c_2^3$. This is a polynomial of a tuple produced by a query where the result of the join between the tuples f_1, $c2f_1$, and c_1 is produced three times and then merged (e.g. as the result of a projection), and the tuples $c2f_2$ and c_2 are used three times in the operation described by the second monomial (e.g., with nested queries). *** Why would the join tuple be produced 3 times? Perhaps as a result of a union? Projection doesn't make sense ***

5. Credit Distribution and Distribution Strategies

We now give formal definitions of data credit and Data Credit Distribution (DCD), and present three different Distribution Strategies (DSs) based on the forms of provenance discussed earlier: Lineage-based DS, Why-Provenance-based DS, and How-Provenance-based DS. We also show how these strategies distribute credit in the IUPHAR example discussed earlier.

5.1. Data Credit and Data Credit Distribution

Given a database instance I, a *recipient of credit* is a unit of information within I. In the case of relational databases, recipients may be (i) the whole database; (ii) a table; (iii) a tuple; or (iv) an attribute.

Data credit is a value $k \in \mathbb{R}_{>0}$. Every recipient in a database is annotated with a quantity of credit as a proxy for its importance. In this paper, we focus on *tuples* as recipients of credit.

Given a *distribution strategy* (DS), *Data Credit Distribution* (DCD) takes a database instance I, quantity of credit k, and query Q over I, and splits k among the recipients of credit in I.

In the following, we use the notation in Cheney et al. [17]: Given an instance I, a *tuple location* (R, t) is a tuple t in relation R. With reference to the running example, `(family,` $\langle f_1$`, Dopamine Receptors, gpcr`$\rangle$`)` is the tuple location of the first tuple in the `family` relation. The set of all tuple locations in I is called *TupleLoc*. We use this to formally define DCD at the *tuple level*.

Definition 5.1. ***Tuple Level Data Credit Distribution (DCD)*** *[24]* *Given a query Q over I and $k \in \mathbb{R}_{>0}$, DCD is defined by the function $f_{I,Q} : TupleLoc \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$ such that $f_{I,Q}(t, k) = h$ where $0 \leq h \leq k$ and $\sum_{t \in TupleLoc} f_{I,Q}(t, k) = k$. The function f_{IQ} is the distribution strategy (DS).*

As we can see, the DS is a function that annotates each tuple in the database with a real value, which is a fraction of the given quantity k. The only constraint is that the sum of the credit annotations on tuples must be k, i.e. that no credit is generated or destroyed during the distribution. Given I and Q, many different DSs may be defined as long as they sum up to k.

In what follows, we use information provided by data provenance to define distribution functions. For simplicity, we assume that the credit k is distributed equally across the set of output tuples (i.e. the result of a query), and discuss how the credit of one output tuple o, k_o, is distributed across the instance I.

5.2. A Lineage-based Distribution Strategy

In the lineage-based distribution strategy, each tuple in the output of a query distributes credit equally to each input tuple that appears in its lineage. More formally:

Definition 5.2. *Lineage-based Distribution Strategy [24]*
Let I be a database instance, Q a query over I, $o \in Q(I)$ an output tuple and k_o the credit associated to o. Let L be the lineage of o and t be a tuple in I,

then t receives credit equal to:

$$f_{I,Q}(t, k_o) = \begin{cases} 0 & \textit{if } t \notin L \\ \frac{k_o}{|L|} & \textit{if } t \in L \end{cases}$$

Note that lineage-based DS distributes credit only to input tuples that have a role in creating o by the query Q, and that each receives an equal share of credit via o. Thus, the more tuples in a lineage set, the less credit each tuple receives.

As an example, consider the output tuples of Table 2, and assume that each output tuple has credit $k_o = 1$. The lineage of the first tuple, o_1, 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 zero credit. The lineage of the second output tuple is $\{f_4, c2f_4, c_1\}$, therefore each of these tuples receives credit 1/3.

At the end of the process, tuples f_1, $c2f_2$ and c_2 each receive credit 1/5, tuples f_4 and $c2f_4$ receive 1/3, while tuple c_1 receives 8/15. Note that if a tuple appears in more than one lineage set, then it will accumulate credit from the distribution associated with each one of these sets, implying that it has a more significant role in the context Q, as is the case with c_1 in this example.

Not all of the tuples in the lineage of an output tuple are necessary to be present at the same time for the output tuple to appear in the query results. 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, while f_1 is always needed for o_1 to appear in the output, only one of the sets of tuples $\{c2f_1, c_1\}$ and $\{c2f_2, c_2\}$ is required. One could therefore argue that it would be more fair for f_1 to receive more credit than the other four tuples, given its role in producing o_1.

This highlights one limitation of the lineage-based DS: while able to find all and only the relevant tuples of the output, it does not distinguish the *importance* of tuples in the query computations. We therefore present two other, more sophisticated, forms of distribution strategies based on why- and how-provenance.

5.3. A Why-Provenance-Based Distribution Strategy

The distribution strategy based on why-provenance first equally distributes the credit k_o among the witnesses of the witness basis for o, and then equally

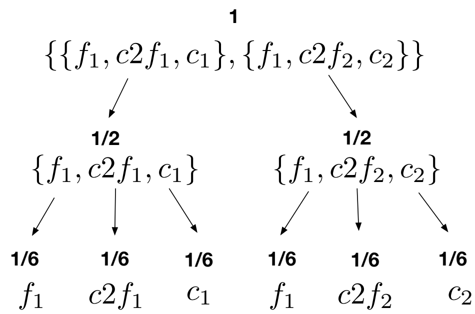

Figure 4: Distribution of credit using why-provenance-based DS for tuple o_1.

divides the credit of a witness among the tuples in the witness. Since a tuple may appear in more than one witness, it will receive more than one portion of credit from the same distribution. More formally:

Definition 5.3. *Why-Provenance-based Distribution Strategy*
Let I be a database instance, Q a query over I, $o \in Q(I)$ an output tuple and k_o the total credit associated to o. Let $\mathcal{W} = Why(Q, I, o)$ be the witness basis of o according to Q and I, and $W \in \mathcal{W}$ be a witness.

Then tuple t in I receives credit equal to:

$$f_{I,Q}(t, k_o) = \frac{k_o}{|\mathcal{W}|} \sum_{W \in \gamma(\mathcal{W}, t)} \frac{1}{|W|}$$

where γ is a function which returns all witnesses W in which t appears:

$$\gamma(\mathcal{W}, t) = \{W \in \mathcal{W} : t \in W\}$$

Figure 4 shows the distribution of credit with why-provenance-based DS for tuple o_1. The credit is first equally divided between the two witnesses, so that both receive credit 1/2. The credit is then further divided among the tuples in each witness. Since each witness has three tuples, each tuple in a witness receives 1/6 of credit. At the end of the distribution, f_1 receives a total credit of 1/3, and the other tuples receive 1/6 each. This distribution better reflects the role of f_1 in the generation of o_1 since, as discussed earlier, it is the only mandatory tuple for o_1 to appear in the output; only one of the two other pairs of tuples are necessary for o_1 to appear in the result.

This example illustrates that why-provenance can better reward input tuples depending on their role. Tuples that appear in more than one witness are rewarded more than others.

$$\mathcal{H} = \underbrace{3f_1 \cdot c2f_1 \cdot c_1}_{M_1} + \underbrace{f_1 \cdot c2f_2^3 \cdot c_2^3}_{M_2}$$

$$\begin{array}{cc} c(\mathcal{H}) = 4 & c(M_2) = 7 \\ mc(M_1) = 3 & mc(M_2) = 1 \\ e(c_2, M_2) = 3 & \gamma(c_1, \mathcal{H}) = \{M_1\} \end{array}$$

$$\gamma(f_1, \mathcal{H}) = \{M_1, M_2\}$$

Figure 5: Illustration of notation used to define the how-provenance based DS in Definition 5.4.

5.4. A How-Provenance Based Distribution Strategy

How-provenance conveys more information than why-provenance since it not only captures what tuples are relevant to the output and in which combination, but also how they are used. The "how" is captured through the provenance polynomials.

The how-provenance-based DS therefore first distributes the credit to the monomials of the polynomial accordingly to the weight represented by their coefficients, then to the tuples of each monomial accordingly to the weights represented by their exponents.

To define the DS more formally, we introduce some notation and illustrate it using the provenance polynomial $\mathcal{H}$ shown in Figure 5.

We call c the function that, given a polynomial, returns the sum of the coefficients of the polynomial; thus $c(\mathcal{H}) = 3+1 = 4$. We use the same name for the function that, given a monomial, returns the sum of its exponents; thus $c(M_2) = 1+3+3 = 7$. mc is the function that takes as input a monomial and returns its coefficient. e is a function that takes as input a tuple and a monomial, and returns the exponent of the tuple in the monomial, if present; thus $e(c_2, M_2) = 3$. γ takes as input a tuple and the whole polynomial, and returns a set containing the monomials containing that tuple, if present in the polynomial; thus $\gamma(f_1, \mathcal{H}) = \{M_1, M_2\}$.

Definition 5.4. *How-Provenance-Based Distribution Strategy*
Let I be a database instance, Q a query over I, $o \in Q(I)$ an output tuple, $\mathcal{H}$ be the provenance polynomial for o, and k_o the credit given to o. The credit given to tuple t in I is:

$$f_{I,Q}(t, k_o) = \frac{k_o}{c(\mathcal{H})} \sum_{M \in \gamma(t,\mathcal{H})} mc(M) \frac{e(t, M)}{c(M)}$$

id	name
oxs_1	Dopamine Receptors

lineage	why-provenance	how-provenance
$\{f_1, c2f_1, c_1, c2f_2, c_2\}$	$\{\{f_1, c2f_1, c_1\}, \{f_1, c2f_2, c_2\}\}$	$f_1^2c2f_1c_1 + f_1^2c2f_2c_2$

Table 5: Result of query `Q2` applied on the database of Table 1 and its different provenances. The reported numbers are the credit distributed through the process.

Going back to the example of Table 4, consider o_1 with provenance polynomial $f_1c2f_1c_1 + f_1c2f_2c_2$. The how-provenance-based DS firstly divides the credit between the two monomials. Since the coefficients of each monomial are 1, the credit is split in half. If they were, for example, 1 and 2 respectively, 1/3 of the credit would go to the first monomial, and 2/3 to the second. Since in our example each variable has exponent 1, the credit is further divided equally among the three variables. Thus, at the end of the computation, f_1 receives 1/3, and the other tuples receive 1/6. If, for example, the first monomial was $f_1^2c2f_1c_1$, then the portion of credit of this monomial would be divided in this way: 1/2 to f_1 and 1/4 to each of the other two tuples.

In this specific example, the how-provenance-based DS has the same outcome as the one based on why-provenance. We therefore consider another query over GtoPdb, `Q2`, that asks for the families of type `gpcr` that have as contributor a researcher located in the UK:

```
Q2: SELECT DISTINCT F.name
FROM family as F JOIN
(SELECT DISTINCT f.name AS 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") AS R ON F.name = R.name
WHERE F.type = "gpcr"
```

The result of `Q2` is shown in Table 5, and consists of one tuple, annotated with each of the three provenances. As can be seen, lineage and why-provenance are identical to those of the tuple o_1 in the previous example. The how-provenance, however, is different since tuple f_1 is used twice: first in the join of the inner query, and second in the join of the outer query. This information is lost in the first two forms of provenances since they are sets, but it is captured in how-provenance through the use of the operator '$\cdot$'.

Figure 6 shows the differences between the three DS for the tuple o_1 of

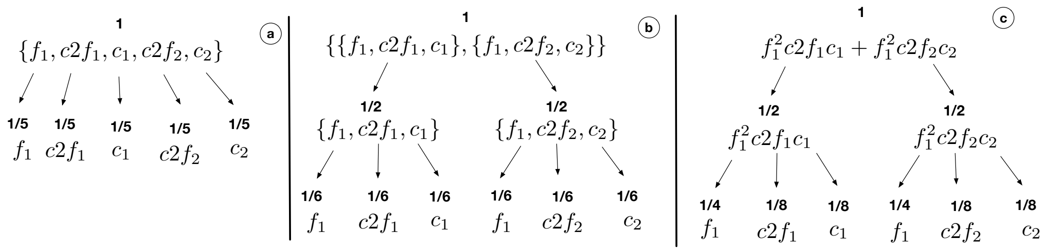

Figure 6: Comparison of different distributions strategies for tuple o_1 produced by query `Q2`.

Table 5. Subfigure 5.a uses lineage, sub-figure 5.b uses why-provenance, and sub-figure 5.c uses how-provenance. The DS based on the provenance polynomial gives credit $1/2$ to f_1, and $1/8$ to the other tuples. This is reasonable since `Q2` relies on f_1 even more than `Q1` does. The distribution based on how-provenance can reward f_1 more, showing that how-provenance is even more sensitive to the tuples' role in a query than why-provenance. This is a direct consequence of the fact that, as proven in [30], how-provenance is more general than why-provenance and lineage, in the sense that it contains more information.

6. Experimental Evaluation: comparing provenances

We evaluate the proposed distribution strategies on GtoPdb, and in particular we focus on target families, all of those are described in webpages. GtoPdb in particular 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 the data of a family, it is 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 Figure 3. In the figure, we can see how the structure of one family, "Adenosine receptors", is mapped into several queries to obtain the information to build the corresponding webpage. In GtoPdb, all family pages share a similar structure (the only differences may be the presence/absence and length of

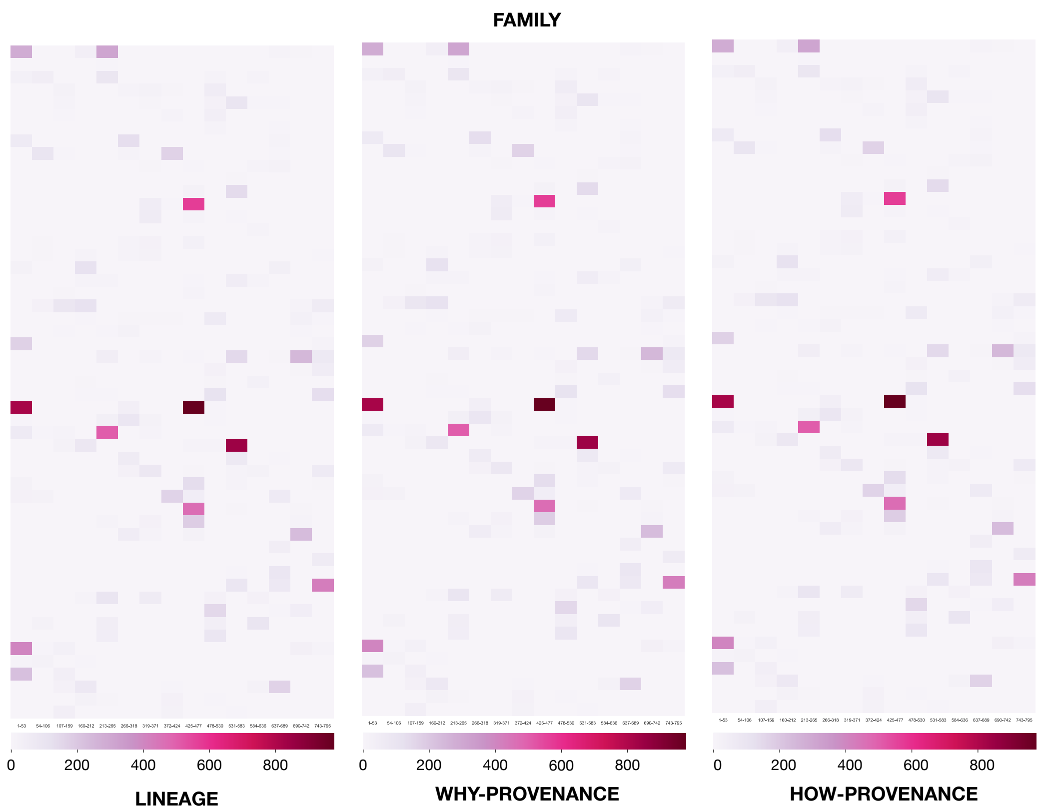

Figure 7: Comparison of three DS on the same table `family` using the distribution given by the queries retrieved from papers.

the receptors lists, further readings, and contributors sections). Therefore, the same queries are used to build all other pages by simply changing the family id (which, in our example, is 3). All these queries are SPJ.

As already stated, many papers that draw information from the GtoPdb website[11] 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 happens, we consider a paper subset citing the 2018 GtoPdb [31] data paper. At the time of writing, this paper received more than 1200 citations.

As explained in Section 3, in the papers published in the British Journal of

[11] `https://www.guidetopharmacology.org`

Clinical Pharmacology, that cite GtoPdb, the name of families are hyperlinks that point to the corresponding webpages. We considered all the 460 papers in BJCP citing [31] as of February 2020. We automatically extracted the URL references to family pages were automatically extracted to guide in building the queries to produce corresponding webpages. A total of $5,945$ different queries were built in this way. [12]

Figure 7 shows the heat-maps obtained by three different DS on the table `contributor`. It is immediately evident that the result of the distribution is the same with the three strategies. The same effect is also obtained in the other tables of the database used by the considered queries. Why is that? It is the case that the conditions in which we produced this experiment are quite peculiar. The queries that we used share similar characteristics. They are all SPJ queries, each of them utilizes each table only once in the join condition (there are no self-joins), and all the joins are made using key attributes. In this particular condition, each tuple of the output presents: (i) a how-provenance that is a single monomial with coefficient 1 and exponent 1 in each variable; (ii) a why-provenance that is composed by only one witness; (iii) a lineage that coincides with the only witness in the witness basis. It is easy to see how, given these queries, the three distributions act in the same way. The credit is always uniformly distributed among the tuples appearing in each provenance.

To better clarify what is happening, let us consider one of the types of queries used to build the output webpage, as shown in Figure 3:

```
Q3: SELECT c.first_names, c.surname
FROM contributor2family AS cf JOIN contributor AS c ON
cf.contributor_id = c.contributor_id
WHERE f.family_id = 3
```

`Q3` returns a series of 10 tuples from the version of GtoPdb we considered. The first tuple produced by this query, `<Bertil B., Fredholm>`, has $c_{939} \cdot c2f_{496}$ as provenance polynomial. c_{939} represents the provenance token of a tuple in `contributor`, the same for $c2f_{496}$ in table `contributor2family`. It is easy to see that the why-provenance of this tuple is $\{\{c_{939}, cf_{496}\}\}$ and its

[12]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`.

lineage is $\{c_{939}, c2f_{496}\}$. Therefore, the credit assigned to these tuples is 1/2 using all three DS. This actually happens for each tuple of the output of each query of GtoPdb, thus making the distributions equivalent.

This is not always the case with general queries and other databases. As we showed in the examples in the previous section, when two or more tuples are merged by the effect of a projection or union, we see sensible differences between the three distribution strategies.

To give an example of how the CDS can differ from one another in their behavior, let us consider a different query:

```
Q4: SELECT f.name AS name
FROM family AS F JOIN
(SELECT DISTINCT f.family_id, f.name
FROM "family" AS f JOIN contributor2family AS cf ON
f.family_id = cf.family_id
JOIN contributor c ON
cf.contributor_id = c.contributor_id
WHERE c.country = 'UK') AS R
ON F.name = R.name
```

Here the innermost query retrieves all the names and ids of the families written by an author from the UK producing a relation called R. This relation is then joined with the table `family` on the attribute `name`.

One output tuple of this query is `<Histamine receptors>`, that has the following provenance polynomial:

$$f_{625}(f_{625}c2f_{656}c_{184} + f_{625}c2f_{113}c_{180} + f_{625}c2f_{283}c_{198}+ \\ +f_{625}c2f_{550}c_{865} + f_{625}c2f_{573}c_{101} + f_{625}c2f_{95}c_{109})$$

As already discussed, the different monomials represent possible *alternatives* of combinations of tuples that produce the considered output tuple. Tuple f_{625} is used each time with different joins, thus it appears in each monomial. The last join, performed in the outmost query, is responsible for the final multiplication of f_{625} with the rest of the polynomial between parenthesis.

From this polynomial we compute the why-provenance as a set of six

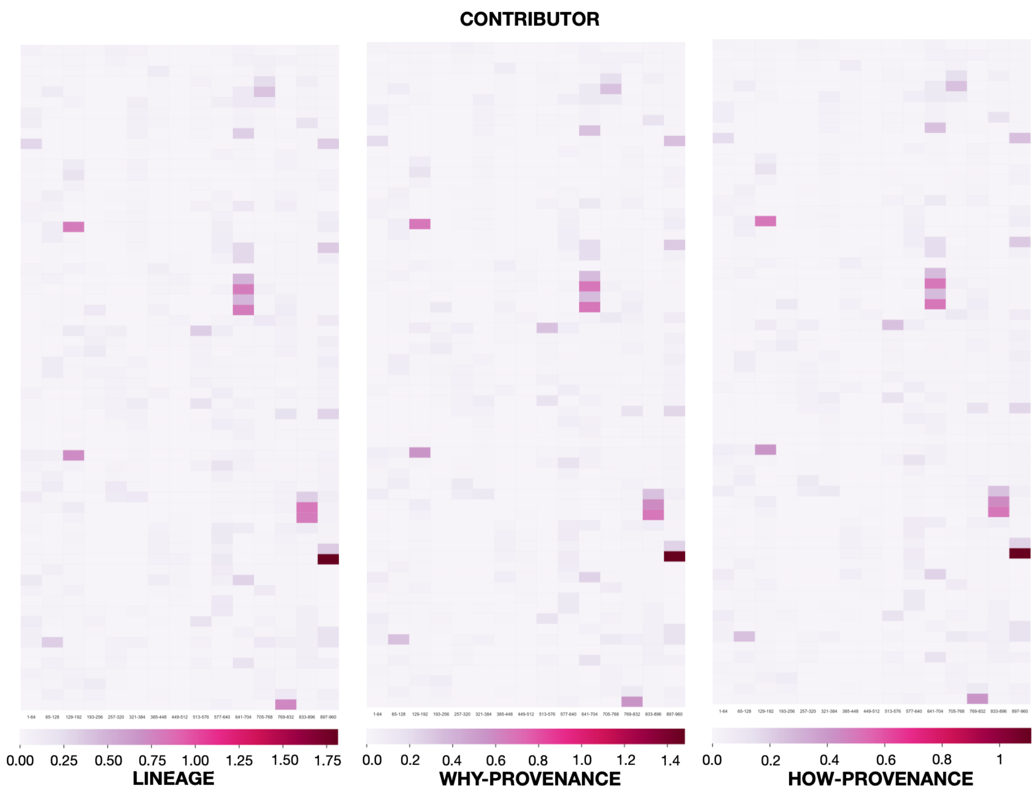

Figure 8: Comparison of three DS on the same table `family` after the distribution of the credit connected to query `Q4`.

different witnesses:

$$\begin{gathered}\{\{f_{625}, c2f_{656}, c_{184}\},\\ \{f_{625}, c2f_{113}, c_{180}\}\\ \{f_{625}, c2f_{283}, c_{198}\},\\ \{f_{625}, c2f_{550}, c_{865}\},\\ \{f_{625}, c2f_{573}, c_{101}\},\\ \{f_{625}, c2f_{95}, c_{109}\}\}\end{gathered}$$

And corresponding lineage:

$$\{f_{625}, c2f_{656}, c_{184}, c2f_{113}, c_{180}, c2f_{283}, c_{198}, c2f_{550}, c_{865}, c2f_{573}, c_{101}, c2f_{95}, c_{109}\}$$

This was only one tuple among the 86 obtained from this query. If we assign credit 1 to all these tuples and distribute it with the different strategies, we obtain the result shown in Figure 8 for the table `contributor`. At first sight, it may appear that the three distributions produce the same result. This is only partially true: the heat maps appear equal, but the absolute values assigned to each tuple are different. This is more evident if we look at the legend of each heat-map, where the maximum quantity of credit is different for each distribution. The one performed through lineage is around 1.8, the why-provenance's one is around 1.4, and the one based on how-provenance is around 1.1.

To understand what is happening with this query in this specific table, consider the output tuple `<Histamine receptors>` and its provenances, as discussed above. Let us focus on its lineage. There are a total of six authors for this family and 13 tuples in total in the lineage. Thus, using the lineage-based DS, each tuple belonging to the `contributor` table (i.e. $c_{184}, c_{180}, c_{198}, c_{865}, c_{101}, c_{109}$) receives credit equal to 1/13. Tuple f_{625} too receives a portion of credit equal to 1/13.

Let us consider now why-provenance. Tuple f_{625} appears six times in six different witnesses composed of 3 elements each. From each witness it receives a portion of credit equal to 1/18, thus its total credit is 1/3. On the other hand, all the authors appear only once in each witness, thus each of them receives credit 1/18. In this case, why-provenance is recognizing more credit to tuple f_{625}, since it appears in each witness. The consequence is that this distribution is equally *subtracting* credit from the other tuples in the witnesses and giving it to f_{625}. In Figure 8 we are only looking at table `contributor`. This same effect is reproduced for each tuple of the output of query `Q4`, thus the *absolute* credit values on the tuples vary depending on the deployed strategy. What happens is that the tuples in table `contributor` receive less credit than the one received using lineage, but in the same proportions. The heat map appears thus equal to the one obtained with lineage. This same effect is also present with the how-provenance-based CDS. In this case, tuple f_{625} is rewarded even more, since it appears with an exponent 2 in each monomial, thus attracting even more credit.

This is also why when we look at the legend for each part of Figure 8, the maximum value reached with the lineage-based DS is higher than the one reached with the why-provenance-based DS, which in turn is higher than the one obtained with the how-provenance. This is because the different strategies reward less and less the tuples of table `contributor` and more the ones in table `family`.

This clearly shows the ability of the different strategies to adapt to situations. All three of them can highlight the relevant tuples in the table.

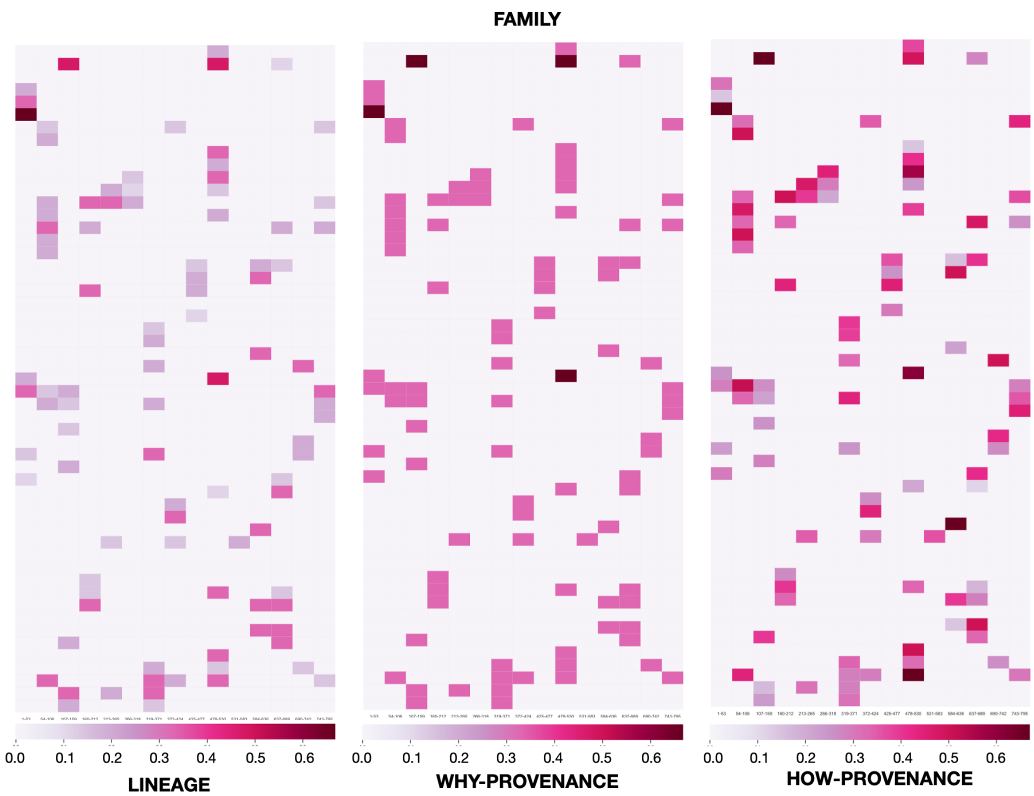

Figure 9: Comparison of three DS on the same table `family` after the distribution computed on provenances randomly generated.

However, they differ in the way they reward the tuples. Depending on the task, one provenance can be preferred to the other. If the only interest is to highlight the relevant tuples, lineage is sufficient. If the interest is also to reward more the tuples that are fundamental to the output, one can also choose why- or how-provenance, knowing that how-provenance rewards even more than why-provenance the relevant tuples that are indispensable for the output.

Let us consider another interesting case we show in Figure 9. The figure reports a distribution of credit performed on `family` through the generation of *synthetic* polynomials. In this last case, we did not produce full-fledged queries. Rather, we randomly generated provenance polynomials that might be the how-provenance of randomly generated synthetic queries. An example of such synthetic polynomial is:

$$3f_1^3 c2f_1^2 c_1^2 + 2f_1 c2f_2^3 c_2^3 + 4f_5 c2f_{17}^4 c_{18}^3$$

As can be seen, we made sure to also include coefficients and exponents that differ from 1. Its corresponding why-provenance is:

$$\{\{f_1, c2f_1, c_1\}, \{f_1, c2f_2, cf_2\}, \{f_5, c2f_{17}, c_{18}\}\}$$

its lineage is:

$$\{f_1, f_5, c2f_1, c_1, c2f_1, c2f_2, c2f_{17}, c_1, c_2, c_{18}\}$$

These types of polynomials are not impossible to obtain. They can be obtained by writing nested queries with join and union operations that use multiple times the same tuples (thus the presence of exponents bigger than 1) and that use the same combination of operations more than once (thus the presence of coefficients for monomials bigger than 1). We randomly generated a set of 100 such polynomials.

Using how-provenance, this is the distribution obtained from the example polynomial we are considering:

$$f_1 = \frac{59}{315}, f_5 = \frac{1}{18}, c2f_1 = \frac{2}{21}, c2f_2 = \frac{2}{15}, c2f_{17} = \frac{2}{9}, c_1 = \frac{2}{21}, c_2 = \frac{2}{15}, c_{17} = \frac{1}{6}$$

Using why-provenance, this is the output:

$$f_1 = \frac{2}{9}, f_5 = \frac{1}{9}, c2f_1 = \frac{1}{9}, c2f_2 = \frac{1}{9}, c2f_{17} = \frac{1}{9}, c_1 = \frac{1}{9}, c_2 = \frac{1}{9}, c_{17} = \frac{1}{9}$$

Finally, with lineage, this is the distribution:

$$f_1 = \frac{1}{8}, f_5 = \frac{1}{8}, c2f_1 = \frac{1}{8}, c2f_2 = \frac{1}{8}, c2f_{17} = \frac{1}{8}, c_1 = \frac{1}{8}, c_2 = \frac{1}{8}, c_{17} = \frac{1}{8}$$

To highlight how the distributions behave differently with these polynomials, consider tuple f_5. f_5 receives the highest quantity of credit when we use the lineage-based distribution. Why-provenance and how-provenance reduce its quantity of credit since more information is available for the computation and the algorithms weigh less and less its role.

Generally speaking, the more complex the distribution, the more polarized the credit is toward the tuples that are used more frequently or with a higher impact in the production of the output tuple. Looking at the heatmaps of Figure 9, it appears that lineage tends to distribute credit more "equally" among the tuples, with only one or two tuples receiving higher quantities of credit, primarily because they are used in many different queries.

Why-provenance produces more tuples that are rewarded with high values of credit. Moreover, it appears that the other tuples that are not on the top of the spectrum are rewarded even more evenly compared to the DS based on lineage. That is, why-provenance, in this case, rewarded many tuples with roughly the same quantity of credit, and few tuples (but more compared to the DS based on lineage) with higher quantities of credit. This is due to the fact that why-provenance not only rewards the presence of a tuple in the computation but also the ways in which it is used.

How-provenance, finally, produces the distribution more sensible to the way a tuple is used in a query. Compared to the previous two DS, it also takes into consideration how many times a tuple is used, and weighs this factor in the distribution. It is interesting to see how certain tuples that received the lowest values of credit with lineage are now rewarded with higher values, showing that their fundamental role in certain queries outshines the fact that other tuples were used more frequently in the set of queries.

For our last set of experiments, consider Figure 10. We still use the 100 polynomials described above and the credit distributed through them. Since these polynomials correspond to queries whose corresponding authors are not easily identifiable, we considered 20 "synthetic" authors, and we randomly assigned one author to each tuple in the database. The authors receive "blocks" of consecutive tuples, with each block of the size varying between 10 and 40. Every time a tuple was used in a provenance polynomial, we assigned one citation to the author corresponding to the tuple. The same author also receives the three different credits assigned to the tuple at the end of the distribution process using the three DS.

Figure 10 presents the radar plot where the 20 authors are sorted based on the normalized number of received citations, together with the corresponding normalized quantities of credits. Credit presents a different behavior from one of the citations, and each form of credit, i.e., the credit obtained from the different DS, behaves differently from the others. For example, it appears that authors T, C, and R that are low in the number of citations are still rewarded more than other more cited authors in terms of credit. Even if the tuples of these authors received fewer citations, they still received more credit than other more cited tuples. This shows how credit can be an effective

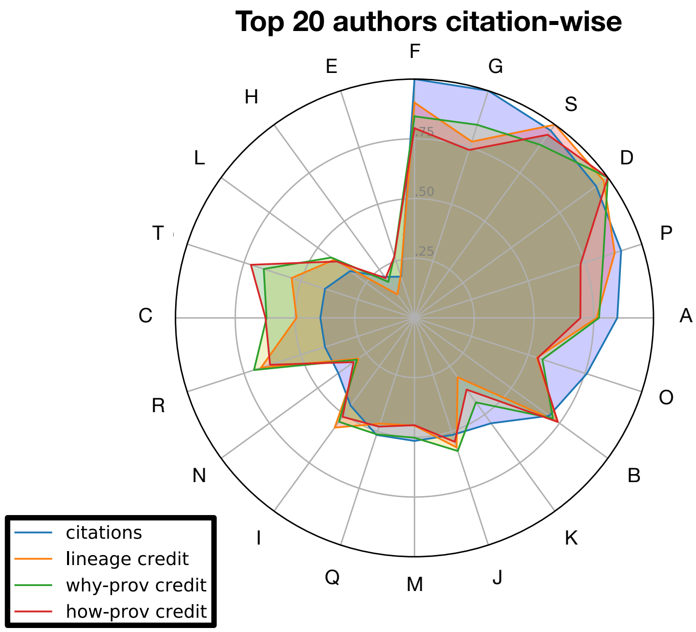

Figure 10: Top 20 authors by number of citations and their credit given through the three different DS.

new method to use together with traditional citations to reward curators, highlighting aspects lost using the traditional bibliometrics.

The three DS are all effective ways to distribute credit, and there is not one distribution that is preferable to the other all the time. It all depends on the needs of the users. Lineage is to be preferred when users only want to see the tuples used in queries and reward more the tuples used in many queries. It only rewards based on the *presence* of the tuples. Why-provenance is more versatile when users also want to consider how many ways a tuple is used; thus, in a way, its *versatility* inside the queries that used it. Finally, how-provenance also counts how many times a tuple is used, its *frequency* in the computation of a query.

6.1. Comparing provenances through time

To show how the DS based on different provenances may actually differ in their behavior, let us consider Figure 11.

In this figure we report four groups of heat-maps. Each group presents three maps obtained by selecting the same ten tuples from the GtoPdb

Figure 11: Comparison of the distribution of credit performed by the three DSs on a subset of 10 tuples taken from table `family` simulating the passing of time. The number on top of each group of heat-maps represent the number of queries computed.

`family` table after an incremental distribution of credit (the tuples of indexes ranging from 653 to 663). In particular, the four groups presents a distribution of credit obtained from the execution of 1K, 2K, 5K and 10K queries. In this way we are simulating the passing of time on a database where credit distribution is performed. Each group of heat-maps can be thought as a snapshot of that set of tuples at a certain moment, after a certain amount of queries are executed. The queries utilized are the same of the experiment of the previous section. The range of credit in each map goes from 0 (no credit) to 6 (maximum quantity of credit reached on a tuple at the "snapshot" reached at 10K queries).

Focusing on the 1K and 2K groups, we see that the three DS do not behave very differently. The tuples highlighted by the three are the same, even when we increment the number of computed queries to 2K. There are differences, in particular in tuple 1 and 10, but are almost negligible.

The first interesting interesting differences appear at 5K queries. In particular, we note how tuple 7 is rewarded poorly by the DS based on lineage, while it is rewarded more by why-provenance-based DS and most of all by the DS based on how-provenance. This is due to the fact that tuples 7 appears in a relative low number of lineages, but its role is critical to these queries, thus the other DS reward it more. On the other hand, a tuple as 5 is rewarded by the DS based on lineage and why-provenance, and less by how-provenance. This means that, although tuple 5 appears in many queries and it is used in different combinations, its exponents in the provenance polynomials where it appears must be low, therefore giving it low credit with how-provenance. It is also interesting to note how certain tuples, like 1, that up until 2K queries presented the highest values of credit, are now surpassed by other tuples like 2. This shows how credit is able, during the passage of time, to keep track of the "hotspots" in a database. The presence of new queries and new credit distribute can change the hotspots in a table, showing how the interests of the research community may change during time.

Finally, the highest differences are shown in the 10K group. In this case, we see a situation similar to the one already seen with the case of 5K queries. Certain tuples, like 8 or 10, receive more credit with why-provenance and how-provenance, rather than with lineage. This is still due to the important role of the tuple in the queries where it appears.

From this progression we see how, given the peculiar synthetic provenance polynomials that we presented, it is actually possible to see the differences between the three distribution. These differences become more and more evident with the passing of time, i.e. the more credit is distributed to the tuples.

7. Conclusions

This paper expanded on our previous work on data credit and data credit distribution by defining two new distribution strategies, based on the why- and how-provenance. The first distribution is based on the concept of witness, and it can give more credit to tuples that appear in more than one witness. In other words, tuples that are more important to the query and are used in different ways by a query are also rewarded more by the distribution. The second distribution, based on how-provenance, considers the frequency in which a tuple or a combination of tuples is used in the query through the provenance polynomial information. In this sense, it is even more sensitive than the first one.

To show the differences between the three DS (also considering the one based on lineage, defined in our previous work), we performed different experiments on GtoPdb, a curated scientific relational database. In the first set of experiments, we used SPJ queries extracted by data citations present in papers published in the British Journal of Pharmacology. Employing these queries, we were able to distribute the credit to the tuples in different tables of the database, highlighting the tuples used more than others. We showed that with these queries, the three strategies produce the same distribution. With the specific type of queries that do not present self-joins, the formulas at the base of the strategies have the same output. In this particular case, the tuples are used in the same way by the queries; thus, the DSs do not register any particular difference in the tuples' role.

In the second and third sets of experiments, we synthetically produced more complex queries, i.e., nested queries whose provenance polynomials presents coefficients and exponents bigger than 1. In this way, we showed that, even though all three DS can highlight all the tuples used by the queries in the database, the three have different behaviors. While the DS based on lineage rewards all the tuples used by a query in equal measure, the strategy based on why-provenance tends to reward the tuples more critical to the query. In particular, why-provenance can consider the different ways in which one tuple is used in a query. How-provenance is even more sensitive to the tuples' role: it can also consider the frequency by which a tuple or a set of tuples is used in the case of more complex queries. Depending on the goal of a user, one provenance may be preferred to another.

In the fourth set of experiments, we showed how, compared with traditional citations, the credit distributed with the three strategies works as a new tool highlighting different aspects of an author's role in the research context identified by queries. Authors with a limited number of citations can still have a high quantity of credit due to the importance of the data to which they contributed to the queries.

In future work, we plan to explore the different potential applications of credit on relational databases. One example is the so-called *data pricing*. Data pricing consists of giving a price to a query submitted by a user who wants to buy the produced information. Currently, a commonly used strategy to face data pricing is based on query rewriting. A database stores a set of views correlated with their price. When a new query arrives, the system tries to rewrite it using the stored views and obtain a query price. This process is computationally expensive. We plan to distribute credit through carefully planned and representative queries and use it as information to define a new, faster, and potentially more flexible pricing function.

Another application is *data reduction* [42], concerned with reducing the vast mole of data that is produced in the evolving world of research and information technology. Data reduction deals with different aspects of dealing with huge amounts of data, such as finding reduced and relevant data streams from the multiple gigabytes of data produced by big data systems every second or dealing with the curse of dimensionality which requires unbounded computational resources to uncover actionable knowledge patters [51].

Data credit can also help to find "hotspots" and "coldspots". A hotspot is data in a database (a tuple or a single attribute, for example) that presents a high quantity of credit and is therefore valuable for the set of queries that distributed that credit. On the other hand, a coldspot is data that present low quantities of credit and can be considered useless or less relevant and can therefore be removed or moved in another cheaper and less efficient memory location.

References

[1] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bernstein, P., Boncz, P., Chaudhuri, S., Cheung, A., Doan, A., Dong, L.,

Franklin, M. J., Freire, J., Halevy, A., Hellerstein, J. M., Idreos, S., Kossmann, D., Kraska, T., Krishnamurthy, S., Markl, V., Melnik, S., Milo, T., Mohan, C., Neumann, T., Chin Ooi, B., Ozcan, F., Patel, J., Pavlo, A., Popa, R., Ramakrishnan, R., Ré, C., Stonebraker, M., and Suciu, D. (2020). The seattle report on database research. *SIGMOD Rec.*, 48(4):44–53.

[2] Alawini, A., Davidson, S. B., Hu, W., and Wu, Y. (2017). Automating data citation in citedb. *PVLDB*, 10(12):1881–1884.

[3] Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., and Wu, Y. (2018). Data citation: A new provenance challenge. *IEEE Data Eng. Bull.*, 41(1):27–38.

[4] Altman, M., Borgman, C. L., Crosas, M., and 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.

[5] Baggerly, K. (2010). Disclose all data in publications. *Nature*, 467(7314):401–401.

[6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D., Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and Goble, C. A. (2013). Why linked data is not enough for scientists. *Future Gener. Comput. Syst.*, 29(2):599–611.

[7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.

[8] Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 591–596.

[9] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Communication*, pages 93–116. De Gruyter Mouton.

[10] Buneman, P. (2006). How to cite curated databases and how to make them citable. In *18th International Conference on Scientific and Statistical Database Management, SSDBM*, pages 195–203. IEEE Computer Society.

[11] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D., Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn't working, and what to do about it. *Database J. Biol. Databases Curation*, 2020.

[12] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation is a computational problem. *Commun. ACM*, 59(9):50–57.

[13] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A characterization of data provenance. In *Database Theory - ICDT 2001, 8th International Conference*, pages 316–330.

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

[15] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N., Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry, R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a., and Wright, D. (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.

[16] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Journals: A Survey. *Journal of the Association for Information Science and Technology*, 66(9):1747–1762.

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

[18] 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*, volume 12.

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

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

[21] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evidence of a structural shift in scholarly communication practices? *JASIST*, 52(7):558–569.

[22] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25(2):179–227.

[23] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research*. www.cidrdb.org.

[24] Dosso, D. and Silvello, G. (2020). Data credit distribution: A new method to estimate databases impact. *Journal of Informetrics*, 14(4):101080.

[25] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The virtual atomic and molecular data centre (VAMDC) consortium. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.

[26] Fang, H. (2018). A discussion of citations from the perspective of the contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–1520.

[27] Force, M., Robinson, N., Matthews, M., Auld, D., and 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, Special Issue on Data Citation*, 12(1):27–30.

[28] Garfield, E. (1999). Journal impact factor: a brief review. *Can. Med. Assoc.*, 979-980.

[29] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data identification and process monitoring for reproducible earth observation research. In *2019 15th International Conference on eScience (eScience)*, pages 28–38. IEEE.

[30] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40. ACM.

[31] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson, A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton, S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F., Spedding, M., Davies, J. A., and Nc-Iuphar (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.

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

[33] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a transformed scientific method.

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

[35] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis, S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432.

[36] 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).

[37] Katz, D. S., Hong, N., Clark, T., Fenner, M., and Martone, M. (2020). Software and data citation. *Computing in Science & Engineering*, 22 (2):4–7.

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

[39] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S. (2011). Citation and Peer Review of Data: Moving Towards Formal Data Publication. *International Journal of Digital Curation*, 6(2):4–37.

[40] Martone, M. (2014). Joint declaration of data citation principles. *FORCE11. San Diego CA. Data Citation Synthesis Group.* `https://www.force11.org/datacitationprinciples`, online September 2020.

[41] Meho, L. I. and 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.

[42] Milo, T. (2019). Getting rid of data. *Journal of Data and Information Quality (JDIQ)*, 12(1):1–7.

[43] Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D., Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson, E., McNutt, M., Miguel, M., Paluck, E. L., Simonsohn, U., Soderberg, C., Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers, E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research culture. *Science*, 348(6242):1422–1425.

[44] Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J. (2016). Research data explored: An extended analysis of citations and altmetrics. *Scientometrics*, 107(2):723–744.

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

[46] Rauber, A., Ari, A., van Uytvanck, D., and Pröll, S. (2016). Identification of Reproducible Subsets for Data Citation, Sharing and Re-Use. *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue on Data Citation*, 12(1):6–15.

[47] Rauber, A., Asmi, A., van Uytvanck, D., and Proell, S. (2015). Data citation of evolving data: Recommendations of the working group on data citation (wgdc). *Result of the RDA Data Citation WG*, 20.

[48] Silvello, G. (2018). Theory and practice of data citation. *J. Assoc. Inf. Sci. Technol.*, 69(1):6–20.

[49] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36.

[50] Spengler, S. (2012). Data Citation and Attribution: A Funder's Perspective. In of Sciences' Board on Research Data, N. A. and Information, editors, *Report from Developing Data Attribution and Citation Practices and Standards: An International Symposium and Workshop*, pages 177–178. National Academies Press: Washington DC.

[51] Ur Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah, T. V., and Khan, S. U. (2016). Big data reduction methods: a survey. *Data Science and Engineering*, 1(4):265–284.

[52] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3.

[53] Wu, Y., Alawini, A., Davidson, S. B., and Silvello, G. (2018). Data citation: Giving credit where credit is due. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD*, pages 99–114.

[54] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to scientific datasets using article citation networks. *Journal of Informetrics*, 14(2).

[55] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of new researchers using the zp-index. *Scientometrics*, 106(3):901–916.

[56] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for Datasets Citation and Extraction Reproducibility in VADMC. *Journal of Molecular Spectroscopy*, 327:122–137.