

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

Digital data is an important form of research product for which citation, and the generation of credit or recognition for authors, is still not well understood. The notion of *data credit* has therefore recently emerged as a new metric, defined and based on data citation theory.

Data credit is a real value that represents the importance of data cited by a paper or by another research entity. Credit can be used to annotate data contained in a curated scientific database, and used as a measure for the importance and impact of that data in the research world. As such, it is a new method that, together with traditional citations, helps recognize the value of data and its creators.

In this paper we explore the problem of Data Credit Distribution, the process by which credit is distributed 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 scientific relational database. We define two new distribution strategies based on two forms 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 be used 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. We also show how different distribution strategies, based on different types of data provenance, can vary in their sensitivity to the role of an input tuple in the generation of the output data, and thus reward input tuples differently.

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 created and understood. They form a basis on which to give credit to authors, papers, and venues [54, 19, 20]. Citations are used, among other things, to decide on tenure, promotion, hiring, and funding of grants for researchers [40, 21, 32, 37].

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 [38, 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].

Many initiatives are carried out at different levels to make data citation a reality. Scientific publishers, such as Elsevier, Springer, and Nature have been defining data policies and author guidelines to include data citations in the reference lists of published papers [19]. The European Commission has introduced the Open Research Data Pilot (ODP), whose aim is to improve and maximize the access and re-use of research data, together with an increase to the credit given to data creators and curators [48]. Initiatives such as FORCE11 and ESIP (Earth Science Information partners) have collaborated on software citation principles and guidelines [26]. Other examples are the National Science Foundation (NSF), and the National Institute of Health (NIH) in the US [48]

In the last years we also saw activity to promote and specify guidelines for data citations. Among these activities, we can cite the example of the Research Data Alliance (RDA), that produced a recommendation on citing specific subsets of dynamic data [47]. This approach is used to cite specific subsets of very dynamic data by assigning and maintaining a PID (Persistent Identifier) for a specific, time-stamped query of a data set. The repository continues to resolve the query PID and maintains or migrates the technology necessary to resolve the query within the data set. This approach is getting broader adoption. However, while this approach provides reference

and access to a precise subset of data, it does necessarily address specific credit concerns for that subset, such as when different authors contribute to a larger collection [43].

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 [28, 9]. However, even when correctly applied, data citations and the bibliometric computed using them do not always correctly or completely 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) [27, 36, 53] 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 DCD 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.

In this paper, we consider data credit as a measure of value for data in a (curated) scientific database. Credit is a real value that 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 DCD 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. Traditionally, when a paper p_1 cites another paper p_2 , a +1 “credit” is given to p_2 , and to all its authors. It does not matter why or how

paper p_1 cites paper p_2 ¹, the result is always +1 to the citation count of p_2 and of its authors. A different credit distribution strategy can assign a quantity of credit to p_2 , and its authors, that is *proportional* to the role played by p_2 in p_1 . Hence, we can weigh the importance of the cited entities and assign credit according to their role.

2. Traditional citations are *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 . Differently, with data credit, we are using 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. Citations, traditionally, are not.

We study the DCD problem in the context of relational databases (RDBs) since they are widely used² 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. 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, (iii) tuples, and (iv) attributes. The clear distinction between the levels of granularity in a relational database allows us to clearly define the DCD problem at one specific level of granularity. In this paper, we focus on DCD at the tuple level.

The DCD process is summarized in Figure 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

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

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

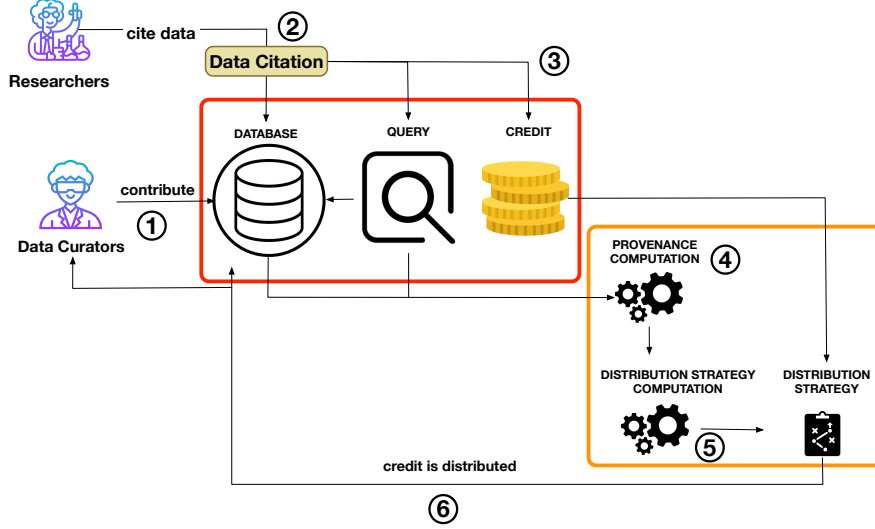


Figure 1: Overview of the credit distribution pipeline.

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.

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

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

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

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

153 For evaluation, we use a well-known curated database, the IUPHAR/BPS³

³International Union of Basic and Clinical Pharmacology/British Pharmacology Soci-
 ety

154 Guide to Pharmacology [31], also known as GtoPdb⁴, which contains ex-
155 pertly curated information about diseases, drugs, cellular drug targets, and
156 their mechanisms of action. We chose GtoPdb for two main reasons: (i) it
157 is a widely-used and valuable curated relational database, (ii) many papers
158 in the literature use, and cite its data (i.e., families, ligands, and receptors).
159 Real queries used in papers can therefore be seen as data citations which, in
160 turn, can be used to assign data credit.

161 We perform four sets of experiments. In the first one, real queries are ex-
162 tracted from papers published in the British Journal of Pharmacology (BJP),
163 that represent data citations to GtoPdb, and are used to distribute credit in
164 the database using the three different provenance-based DSs. In the second
165 and third experiment we analyse the behaviour of the different DS when com-
166 plex citation queries are employed. In the fourth set of experiments we use
167 both real and synthetic queries to assess the differences between traditional
168 citations and credit in rewarding data curators.

169 **Contributions.** Contributions of this work include:

- 170 • The definition of new distribution strategies for the problem of Data
171 Credit Distribution, based on why-provenance and how-provenance;
- 172 • An in-depth analysis of the effects of credit distribution on real-world
173 curated data and of the differences between the three proposed Distri-
174 bution Strategies.
- 175 • A comparison between the behavior of traditional citations and data
176 credit in rewarding data curators.

177 **Outline.** The rest of the paper is organized as follows: Section 2 presents
178 the background and related work. Section 3 describes the GtoPdb use case
179 we adopted. Section 4 briefly presents the forms of provenance used in the
180 paper. Section 5 describes the credit distribution problem and the proposed
181 distribution strategies. In Section 6 we present the experimental evaluation.
182 Finally, Section 7 draws some conclusions and outlines future work.

⁴<https://www.guidetopharmacology.org/>

183 2. Background

184 *Data in Research.* The world of research is rapidly transitioning towards the
185 *fourth paradigm of science* [33], that is, data-intensive scientific discovery,
186 where data are important for scientific advances as well as for traditional
187 publications [6].

188 The scientific community is promoting an *open research culture* [42],
189 founded on methods and tools to share, discover, and access experimental
190 data. The community has identified the FAIR principles (Findable, Acces-
191 sible, Interoperable, and Reusable) [51], that should be enforced by every
192 database. In particular, data should be accessible from the articles, journals,
193 and papers that cite or use them [19]. Aspects such as the need for the *repro-*
194 *ducibility* of experiments through the used data; the *availability* of scientific
195 data; the *connections* between data and the scientific results are all needed
196 aspects for the fourth paradigm, and are all relevant to the domain of *data*
197 *citation* [34].

198 *Data Citation: Principles and Motivations.* Data Citation principles were
199 proposed in [18], and later summarized and endorsed by the Joint Declaration
200 of Data Citation Principles (JDDCP) [39]. The principles are divided into
201 two groups [48]. The first one contains principles concerning the role of
202 data citation in scholarly and research activities such as the (i) *importance*
203 of data (why data citation is important and why data should be considered
204 as first-class citizens); (ii) *credit* and *attribution* to the creators and curators
205 of the data; (iii) *evidence*; (iv) *verifiability*; and *interoperability*, with these
206 last three requiring data citation methods to be flexible enough to operate
207 through different communities. The second group defines the main guidelines
208 to establish a data citation systems, and contains principles such as the (i)
209 *unique identification* of the data being cited; (ii) *(open) access* to data; (iii)
210 guarantee of *persistence* and *availability* of citations even after the lifespan
211 of the cited entity; the (iv) *specificity* of a citation, i.e. it must lead to the
212 data set originally cited.

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

- 214 • *Data attribution:* identify the individuals that should be credited for
215 data with variable granularity.
- 216 • *Data connection:* connect papers to the data being used.

- 217 • *Data Discovery*: citations helps to find data records and subsets that
218 would be otherwise not findable via search engines.
- 219 • *Data Sharing*: share data obtained by researchers within the whole
220 community.
- 221 • *Data Impact*: highlight the results obtained in writing papers using
222 specific data, the frequency and modality data were used.
- 223 • *Reproducibility*: data citation greatly impacts the reproducibility of
224 science [5]. Many authoritative journals ask to share data and provide
225 valid methodologies to reproduce experiments.

226 2.1. Data Citation in Relational Databases

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

234 In one of its most recent sessions [47], the Working Group (WG) on
235 Data Citation reported that there are various implementations of its guide-
236 lines for Data Citation on MySQL/Postgres relational databases. Some of
237 these databases are: DEXHELPP⁶ (Social Security Records); NERC (ARGO
238 Global Array); EODC (Earth Observation Data Centre) [29]; LNEC (River
239 dam monitoring); MDS (Million Song Database) [8]; CBMI⁷ (Center for
240 Biomedical Informatics); VMC (Vermont Monitoring Cooperative); CCA⁸
241 (Climate Change Center Austria); VAMDC (Virtual Atomic and Molecular
242 Data Center) [25, 55].

243 More examples of work on data citation in relational databases are [12,
244 52, 2, 23]. The website <https://fairsharing.org/> keeps a long updated

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>

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, [52] 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 papers” 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 [55]. Every single Webpage is unequivocally identifiable and can be individually cited.

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.

Katz [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

282 products as software and data. In the paper two considerations are made:
 283 (i) research objects such as data and software are currently not formally
 284 rewarded or recognized by the community; (ii) even in traditional papers,
 285 the contribution of each author to the work is hard to understand, unless
 286 explicitly specified in the paper. This is even more true for data, where
 287 different groups of people work on the same database.

288 In [36] credit is defined as a “quantity” that describes the importance of a
 289 research entity, such as papers, software, or data, mentioned in a citation. It
 290 also proposed the idea of a *distribution* of credit from research entities, such
 291 as papers or data, to other research entities through citations. This can be
 292 done by exploiting the structure of the *citation graph*, a directed graph whose
 293 nodes are publications and edges are citations. This graph is the model at
 294 the core of systems such as Google Scholar and the Web of Science. We
 295 add to this that the concept of credit can be built on top of the existing
 296 infrastructure handling traditional and data citations.

297 Katz [36] further explores the idea of a *distribution* of credit from research
 298 entities (i.e., papers and data) to other research entities through citations
 299 that connect them. Thanks to traditional citations and now also to data
 300 citations, this distribution is finally possible, at least between papers and
 301 data. Some problems related to traditional citations can thus be solved by
 302 citations:

- 303 1. Credit rewards research entities that to date are not (formally) recog-
 304 nized (a goal shared with data citation).
- 305 2. Credit can reward authors *proportionally* to their role in generating the
 306 entity. The more an author contributes to a paper, the more credit is
 307 given to him. Zou and Peterson [54] work on something similar with
 308 their zp-index, which includes in its formulation the position (and thus
 309 the role) of a publication author to represent its impact in the work
 310 itself.
- 311 3. Credit can be *transitively* channeled through a chain of papers citing
 312 each other, thus enabling the rewarding of older papers that are no
 313 more cited, since other papers summarize or report their content but
 314 are nevertheless crucial in a research area for the influence of their
 315 content.

316 Fang [27] presents a framework to distribute the credit generated by a
 317 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.

Zeng et al. [53] 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 Zeng et al. [53], 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].

Cheney et al. [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 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

355 tuples that are used in some way by the query to produce the output tuple,
356 the ones that are somehow *relevant* to its generation.

357 The definition of why-provenance is based on the notion of *witness set*.
358 A witness is a set of relevant tuples that guarantees the existence of t in
359 $Q(D)$. The lineage is therefore an example of a witness. The why-provenance
360 of a tuple t is a peculiar set of witnesses – described in [13] – that are
361 computed from the query, called *witness basis*. A witness basis may be
362 composed of more than one witness. Therefore, the why-provenance contains
363 more information than the lineage, since it describes *alternative* ways in
364 which the same output may be generated.

365 The how-provenance takes the form of a polynomial, called *provenance*
366 *polynomial*, where the variables are taken from the set of identifiers of the
367 tuples (provided that each tuple in I has an identifier) and the coefficients are
368 drew from \mathbb{N} . This provenance also contains information on *how* the input
369 tuples are used. For example, when two tuples are combined by a join, they
370 are also combined in the polynomial by the \cdot operator. When two or more
371 tuples become equivalent due to a union or a projection, the corresponding
372 monomials are combined by the $+$ operator.

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

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

378 3. Use Case: GtoPdb

379 As use case we refer to the IUPHAR/BPS Guide to Pharmacology [31]
380 or GtoPdb⁹. GtoPdb is a well-known and well structured scientific relational
381 database that contains expertly curated information about diseases, drugs
382 in clinical use, their cellular targets, and the mechanisms of action on the
383 human body. It is curated and maintained by the GtoPdb Committee, and
384 by 96 subcommittees, comprising 512 scientists collaborating with in-house
385 curators who draw the information contained in the database from high-
386 quality pharmacological and medicinal chemistry literature. Roughly 1000
387 researchers from all over the world have contributed to the database, and the

⁹<https://www.guidetopharmacology.org/>

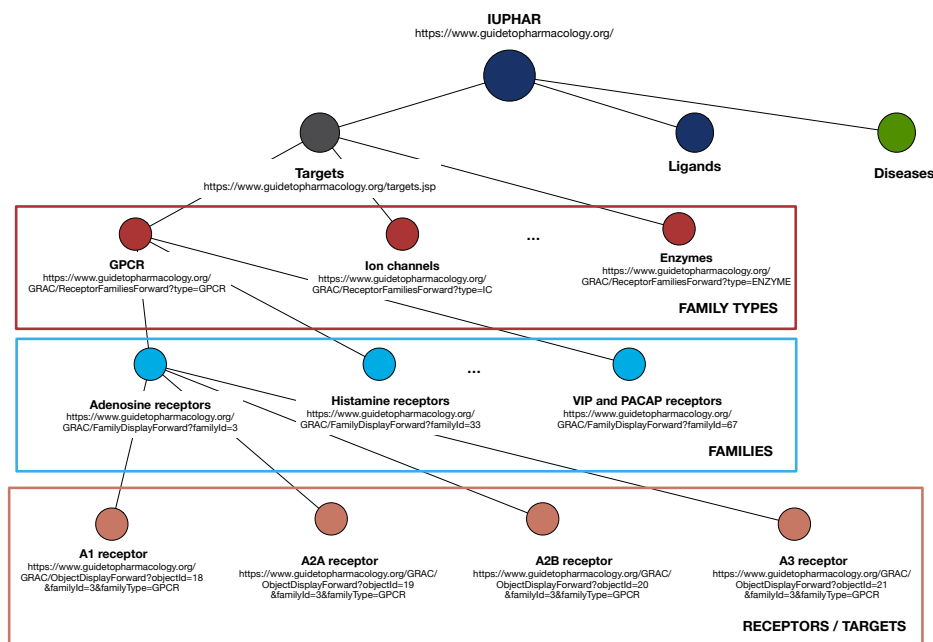


Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

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

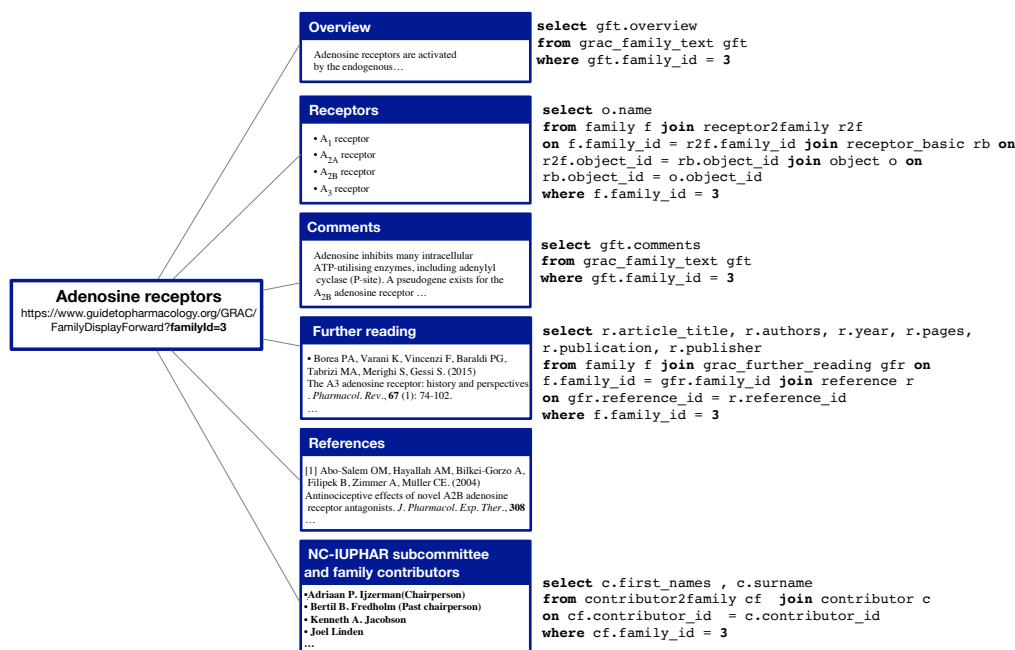


Figure 3: Basic web-page structure of “Adenosine receptors” family (ID 3), with queries used to retrieve the information contained in every section, except references.

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

family			contributor2family		
id	name	type	id	family_id	contributor_id
f_1	Dopamine Receptors	gpcr	$c2f_1$	f_1	c_1
f_2	Bile Acid Receptor	gpcr	$c2f_2$	f_1	c_2
f_3	FAK Family	enzyme	$c2f_3$	f_2	c_3
f_4	YANK Family	enzyme	$c2f_4$	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.

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” (c_1) contributed to “Dopamine Receptors” (f_1) as well as to the “YANK Family” (f_4). We use this example throughout the rest of the paper. In particular, we are using 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.

¹⁰<https://bpspubs.onlinelibrary.wiley.com/journal/13652125>

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 contributed to its “production” [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 set, 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.

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.

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 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 union), 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).

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, $(\text{family}, \langle f_1, \text{Dopamine Receptors}, \text{gpcr} \rangle)$ 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 $f_{I,Q} : \text{TupleLoc} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$ such that $f_{I,Q}(t, k) = h$ where $0 \leq h \leq k$ and $\sum_{t \in \text{TupleLoc}} f_{I,Q}(t, k) = k$. The function $f_{I,Q}$ 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 .

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

564 5.2. A Lineage-based Distribution Strategy

565 In the lineage-based distribution strategy, each tuple in the output of
 566 a query distributes credit equally to each input tuple that appears in its
 567 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:

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

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

572 As an example, consider the output tuples of Table 2, and assume that
 573 each output tuple has credit $k_o = 1$. The lineage of the first tuple, o_1 , is
 574 the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$. Therefore, each tuple in this set receives credit
 575 $1/5$. The other tuples of the database receive zero credit. The lineage of the
 576 second output tuple is $\{f_4, c2f_4, c_1\}$, therefore each of these tuples receives
 577 credit $1/3$.

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

584 Not all of the tuples in the lineage of an output tuple are necessary to be
 585 present at the same time for the output tuple to appear in the query results.
 586 For example, if the database only had the set of tuples $\{f_1, c2f_1, c_1\}$ or the set

587 $\{f_1, c2f_2, c_2\}$, the existence of o_1 would still be guaranteed. In other words,
 588 while f_1 is always needed for o_1 to appear in the output, only one of the sets
 589 of tuples $\{c2f_1, c_1\}$ and $\{c2f_2, c_2\}$ is required. One could therefore argue that
 590 it would be more fair for f_1 to receive more credit than the other four tuples,
 591 given its role in producing o_1 .

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

597 5.3. A Why-Provenance-Based Distribution Strategy

598 The distribution strategy based on why-provenance first equally distributes
 599 the credit k_o among the witnesses of the witness basis for o , and then equally
 600 divides the credit of a witness among the tuples in the witness. Since a tuple
 601 may appear in more than one witness, it will receive more than one portion
 602 of credit from the same distribution. More formally:

603 **Definition 5.3.** *Why-Provenance-based Distribution Strategy*

604 *Let I be a database instance, Q a query over I , $o \in Q(I)$ an output tuple*
 605 *and k_o the total credit associated to o . Let $\mathcal{W} = \text{Why}(Q, I, o)$ be the witness*
 606 *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:

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

607 Figure 4 shows the distribution of credit with why-provenance-based DS
 608 for tuple o_1 . The credit is first equally divided between the two witnesses, so
 609 that both receive credit $1/2$. The credit is then further divided among the
 610 tuples in each witness. Since each witness has three tuples, each tuple in a
 611 witness receives $1/6$ of credit. At the end of the distribution, f_1 receives a
 612 total credit of $1/3$, and the other tuples receive $1/6$ each. This distribution
 613 better reflects the role of f_1 in the generation of o_1 since, as discussed earlier,

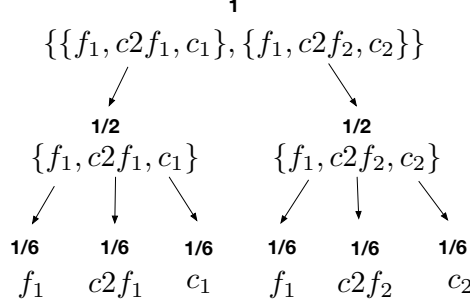


Figure 4: Distribution of credit using why-provenance-based DS for tuple o_1 .

614 it is the only mandatory tuple for o_1 to appear in the output; only one of the
615 two other pairs of tuples are necessary for o_1 to appear in the result.

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

619 5.4. A How-Provenance Based Distribution Strategy

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

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

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

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

$$\begin{aligned}
\mathcal{H} &= \underbrace{3f_1 \cdot c2f_1 \cdot c_1}_{M_1} + \underbrace{f_1 \cdot c2f_2^3 \cdot c_2^3}_{M_2} \\
c(\mathcal{H}) &= 4 & c(M_2) &= 7 \\
mc(M_1) &= 3 & mc(M_2) &= 1 \\
e(c_2, M_2) &= 3 & \gamma(c_1, \mathcal{H}) &= \{M_1\} \\
\gamma(f_1, \mathcal{H}) &= \{M_1, M_2\}
\end{aligned}$$

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

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

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

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

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

```

658 Q2: SELECT DISTINCT F.name
659 FROM family as F JOIN
660 (SELECT DISTINCT f.name AS name
661 FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
662 JOIN contributor AS c ON c2f.contributor_id = c.id

```


id	name
oxs_1	Dopamine Receptors

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

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.

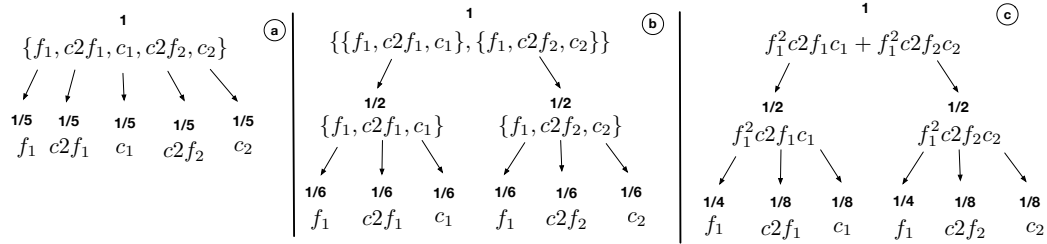


Figure 6: Comparison of different distributions strategies for tuple o_1 produced by query Q2.

```

663 WHERE c.country = "UK") AS R ON F.name = R.name
664 WHERE F.type = "gpcr"

```

665 The result of Q2 is shown in Table 5, and consists of one tuple, anno-
666 tated with each of the three provenances. As can be seen, lineage and why-
667 provenance are identical to those of the tuple o_1 in the previous example.
668 The how-provenance, however, is different since tuple f_1 is used twice: first
669 in the join of the inner query, and second in the join of the outer query. This
670 information is lost in the first two forms of provenances since they are sets,
671 but it is captured in how-provenance through the use of the operator ‘.’.

672 Figure 6 shows the differences between the three DS for the tuple o_1 of
673 Table 5. Subfigure 5.a uses lineage, sub-figure 5.b uses why-provenance, and
674 sub-figure 5.c uses how-provenance. The DS based on the provenance poly-
675 nomial gives credit 1/2 to f_1 , and 1/8 to the other tuples. This is reasonable
676 since Q2 relies on f_1 even more than Q1 does. The distribution based on
677 how-provenance can reward f_1 more, showing that how-provenance is even
678 more sensitive to the tuples’ role in a query than why-provenance. This is
679 a direct consequence of the fact that, as proven in [30], how-provenance is
680 more general than why-provenance and lineage, in the sense that it contains
681 more information.

682 6. Experimental Evaluation

683 To understand the trade-offs between these Distribution Strategies (DSs),
684 we perform four sets of experiments using queries over target families pre-
685 sented on the GtoPdb website. The first set of experiments use real queries
686 extracted from citations to GtoPdb published in the British Journal of Phar-
687 macology. The second set uses synthetically produced provenance polyno-
688 mials, corresponding to more complex queries, in order to better highlight
689 the differences between the DSs. The third set of experiments considers
690 the accrual of credit over time by the three strategies, again using synthetic
691 queries. The fourth set of experiments shows how the DSs compare to tradi-
692 tional citations in giving credit to data curators using both real and synthetic
693 queries.

694 All experiments were carried out on a MacBook Pro with a 2.4 GHz
695 processor Intel Core i5 quad-core and 8 GB of memory at 2133 MHz. Code
696 was written in Java, supported by a PostgreSQL database.¹¹

697 6.1. Real-world queries

698 Examples of real queries are drawn from papers published in the British
699 Journal of Pharmacology (BJP).¹² Each time a paper in this journal cites a
700 webpage from GtoPdb, it reports the URL of the page. From this URL, the
701 query used to obtain the webpage data can be determined. We considered all
702 889 papers in BJCP citing the IUPHAR/BPS Guide to pharmacology [31]
703 as of October 2020, and extracted all webpage URLs to GtoPdb contained
704 within the paper.¹³

705 The queries that we inferred are those used to build target family web-
706 pages within GtoPdb. An example was given in Figure 3, where we show
707 how the structure of the “Adenosine receptors” family can be mapped into
708 queries over the underlying database. In GtoPdb, all target family pages
709 share a similar structure; the only difference is that individual sections, such
710 as “contributors” or “further readings”, may be absent. Therefore, the same

¹¹For purposes of reproducibility, the code we used for our experiments and all queries are available here: https://bitbucket.org/dennis_dosso/credit_distribution_project.

¹²<https://bpspubs.onlinelibrary.wiley.com>

¹³The IUPHAR/BPS Guide is a journal that describes the structure and evolution of GtoPdb. At the time of writing, it had received more than 1200 citations on Google Scholar.



Figure 7: Comparison of three DS on the same table **family** using the distribution given by the queries retrieved from papers. Each cell is a tuple.

711 queries can be used to build all of the target family pages by changing the
 712 family id used in the query (for example, in Figure 3, it is 3). Note that
 713 the queries are fairly simple SQL queries, and fall into a class called “select-
 714 project-join” or “SPJ” queries. A total of more than 12K different queries
 715 were built in this way. Without loss of generality, we give each tuple in the
 716 output of a query a credit of 1.

717 *Results.* Figure 7 shows the heat-maps obtained by the distribution of credit
 718 according to the three different DS on one of the tables in the underlying
 719 database, **family**, which is often joined with other tables in the database to
 720 build the webpages. Each cell in a heat-map represents a tuple of the **family**
 721 table and the color indicates the amount of credit attributed to such tuple.
 722 It can be seen that the result of credit distribution over **family** is the same

for all three strategies. The same result is also obtained with the other tables of the database used by the queries shown in Figure 3.

The reason why credit distribution is the same for all three strategies is that the queries are all simple SPJ queries, which use each table only once and do joins on key attributes. Under these conditions, 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 with only one witness; and (iii) a lineage that coincides with the witness in the basis. Hence, for these queries, the three DSs behave in the same way: credit is uniformly distributed among the tuples present in each provenance.

To illustrate this, consider one of the queries in Figure 3 which is used to build the output webpage:

```

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 returned 10 tuples from the version of GtoPdb used. The first tuple, <Bertil B., Fredholm>, has $c_{939} \cdot c_{2f_{496}}$ as its provenance polynomial. c_{939} represents the provenance token of a tuple in `contributor`, and $c_{2f_{496}}$ the provenance token of a tuple in table `contributor2family`. The why-provenance of this tuple is $\{\{c_{939}, c_{2f_{496}}\}\}$ and its lineage is $\{c_{939}, c_{2f_{496}}\}$. Therefore, the credit assigned to these tuples is 1/2 using all three DS. This happens for all the tuples in the output of each query of GtoPdb, thus making the distributions equivalent over all outputs.

However, this is not the case with more complex queries. As we showed in the previous section, when two or more tuples are merged as a result of a projection or union, the credit distributions will differ between the three strategies.

6.2. Synthetic queries

To simulate synthetic queries, we randomly generated provenance polynomials in which the coefficients and exponents could be greater than 1. The queries involve three GtoPdb tables: `family`, `contributor2family`, and `contributor`. An example can be found in Figure 8, which shows a sample synthetic provenance polynomial (the how-provenance) and the corresponding why-provenance and lineage expressions. The resulting credit distribution for each DS is shown after the provenance expression.

How-provenance: $3f_1^3c_2f_1^2c_1^2 + 2f_1c_2f_2^3c_2^3 + 4f_5c_2f_{17}^4c_{18}^3$

Credit distribution:

$$f_1 = \frac{59}{315}, f_5 = \frac{1}{18}, c_2f_1 = \frac{2}{21}, c_2f_2 = \frac{2}{15}, c_2f_{17} = \frac{2}{9}, c_1 = \frac{2}{21}, c_2 = \frac{2}{15}, c_{18} = \frac{1}{6}$$

Why-provenance: $\{\{f_1, c_2f_1, c_1\}, \{f_1, c_2f_2, c_2\}, \{f_5, c_2f_{17}, c_{18}\}\}$

Credit distribution:

$$f_1 = \frac{2}{9}, f_5 = \frac{1}{9}, c_2f_1 = \frac{1}{9}, c_2f_2 = \frac{1}{9}, c_2f_{17} = \frac{1}{9}, c_1 = \frac{1}{9}, c_2 = \frac{1}{9}, c_{18} = \frac{1}{9}$$

Lineage: $\{f_1, f_5, c_2f_1, c_1, c_2f_1, c_2f_2, c_2f_{17}, c_1, c_2, c_{18}\}$

Credit distribution:

$$f_1 = \frac{1}{8}, f_5 = \frac{1}{8}, c_2f_1 = \frac{1}{8}, c_2f_2 = \frac{1}{8}, c_2f_{17} = \frac{1}{8}, c_1 = \frac{1}{8}, c_2 = \frac{1}{8}, c_{18} = \frac{1}{8}$$

Figure 8: Sample synthetic provenance polynomial (how-provenance) and corresponding why-provenance and lineage expressions with credit distributions.

As an example of how the distribution strategies behave with these synthetic queries, consider tuple f_5 in Figure 8. This tuple receives the highest quantity of credit using lineage-based distribution, and less credit using why- and how-provenance because more information is available about the role of the tuple in the overall computation. Generally speaking, the more complex the distribution (the most complex being how-provenance), the more credit is given to tuples which are more frequently used, and thus have a higher impact in producing the output tuple.

Despite being synthetic, these provenance polynomials represent realistic queries. The polynomials can be obtained by any nested query with join and union operations that use the same tuple multiple times (in which case the exponents are bigger than 1), and the same combination of operations more than once (in which case the coefficients of monomials are bigger than 1).

Results. The results of credit distribution on the `family` table using 10K randomly generated synthetic provenance polynomials are shown in Figure 9. We set the maximum value in the heat maps to the highest value reached by a tuple in all three distributions (i.e., 9.4).

As can be seen, the three strategies generate significantly different credit distributions indicated by the varying hues. However, there is a certain amount of consistency between them in that tuples which are highly rewarded by one strategy are also highly rewarded by the others. This shows that the three DSs consistently reward certain tuples more than others.

Note that lineage-based DS gives the least credit to tuples in the `family`

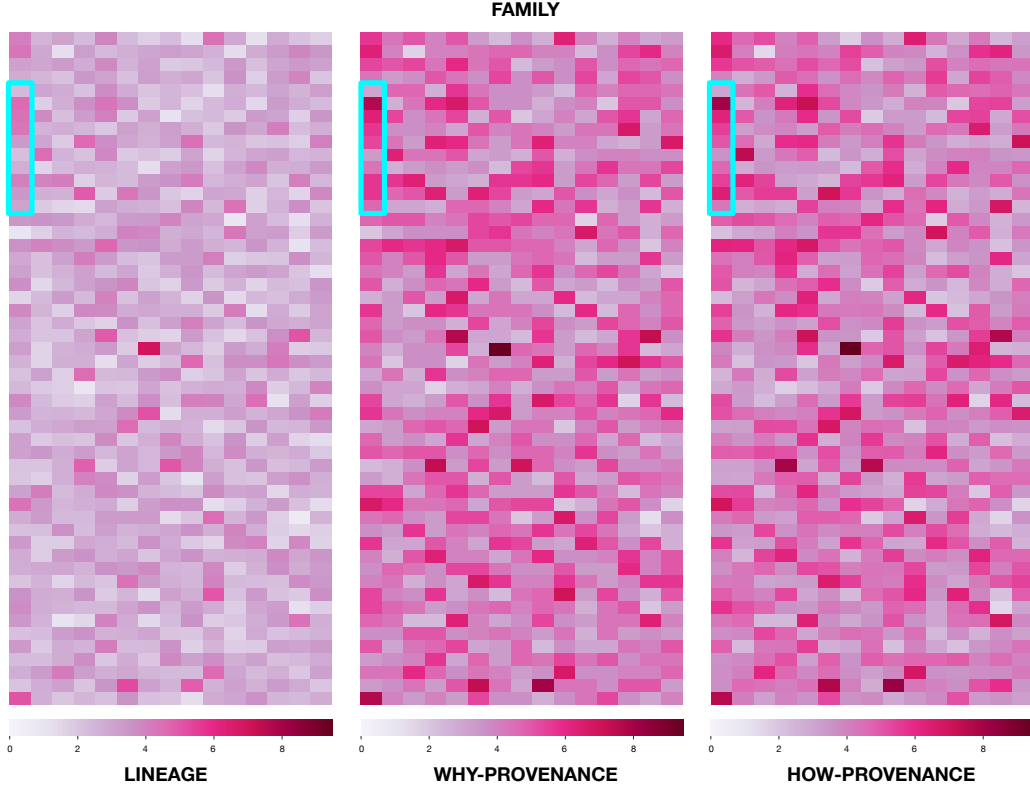


Figure 9: Comparison of three DS on the same table **family** after the distribution computed using 10K synthetic and randomly generated provenance polynomials. The tuples in the blue rectangles are used as example in the discussion connected to Figure 10.

782 table, indicated by an overall lighter hue. This is because the DS distributes
 783 credit equally to all tuples appearing in the lineage. Since these queries also
 784 use two other tables, credit is distributed to tuples in those tables.

785 Moving to why-provenance-based DS, we see that more credit is given to
 786 tuples in the **family** table than with the previous strategy. This is because
 787 the DS considers the different ways that a tuple is used, e.g. in joins with
 788 other tuples. If the same tuple is present in more than one witness, it will
 789 draw more credit and take it from other tuples in the witness basis. In this
 790 case, tuples in **family** drew more credit, taking it from tuples in the other
 791 two tables, due to the role that **family** tuples played in the queries that were
 792 executed. We note that the lineage-based DS gives an average credit of 2.79
 793 to each tuple in the table, while the DS based on why-provenance assigns

4.18. Moreover, lineage distributed a total of circa 2200 units of credit to the table, while the other DS assigned more than 3300 units. That is, the DS based on why-provenance assigns on average 50% more credit to the **family** table than the strategy based on lineage.

Finally, consider the how-provenance-based DS heat-map. As with why-provenance, more credit is typically given to tuples in **family** compared to lineage-based DS since it recognizes the role of these tuples in the queries, and the overall hue is deeper. The two distributions appear similar, although on closer inspection, slight differences between the two distributions can be seen. This is because how-provenance also considers the frequency with which tuples are used, not only the ways in which they are used. Therefore, although the overall distribution is similar, there are small differences due to the presence of exponents and coefficients in the provenance polynomials, influencing the distribution of credit.

To better understand this difference, in the next subsection we consider the accrual of credit over time. In doing so, we will focus on the ten tuples shown within the large light blue rectangles in Figure 10. Each small rectangle within a large blue rectangle is a tuple, and we number them from 1 (top) to ten (bottom).

6.3. Credit accrual over time

Since credit accrues over time, we simulate the passage of time by varying the number of queries executed, and look at the “snapshots” of credit for each of the strategies using synthetic queries. The results are shown in Figure 10.

In this figure, four groups of heat-maps are shown. Each group represents a “snapshot” taken after 1K, 2K, 5K and 10K provenance polynomials have been considered for credit distribution. The ten tuples in each heat-map are those highlighted in the light blue boxes of Figure 9 from the **family** table.

The queries used are the same as the experiment of the previous section. The range of credit in each map goes from 0 (no credit) to 8 (the maximum quantity of credit reached on one of the tuples of the considered window at the “snapshot” with 10K queries). The color hue of the legend, as can be seen, still ranges from 0 to 9.5.

By the end of 1K queries, credit differentials between tuples as well as between strategies can be seen. For example, tuple 4 is usually the one rewarded the most by all three strategies. However, it receives the highest quantity of credit from the why-provenance-based strategy. Tuple 3 is the one that receives the highest value of credit overall with how-provenance. This



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

831 trend continues to the end of 2k queries. By the end of 5k queries, tuple
832 2 emerges with the highest value of credit for why- and how-provenance, a
833 position which is strengthened by the end of 10k queries. This is because
834 tuple 2 is used several times within queries being executed, which is rewarded
835 strongly by why- and how-provenance but not taken into account in lineage.

836 While the relative value of credit “positions” of tuples within a DS strat-
837 egy depends on what queries are being executed, the important thing to
838 notice is the difference between the DSs over time: Overall, lineage gives far
839 less credit to tuples in the **family** table than the other two strategies since
840 credit is shared with tuples in other tables. However, the why- and how-
841 provenance-based strategies recognize the more important role being played

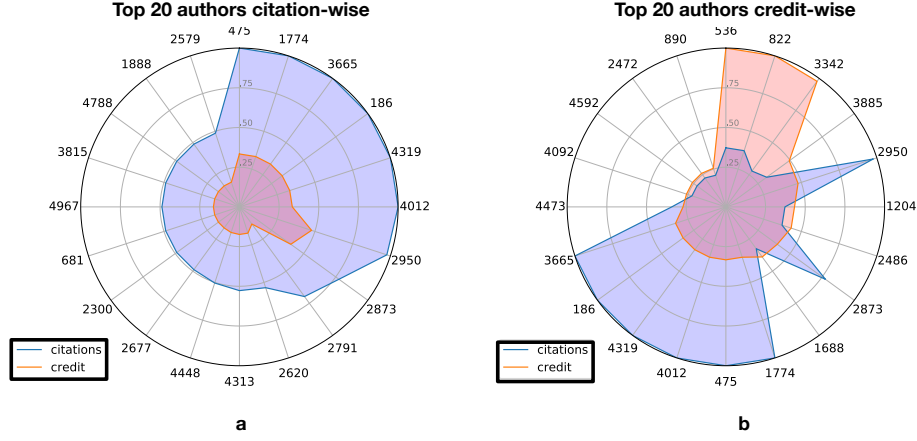


Figure 11: Radars presenting the top 20 authors citation-wise and credit wise, together with their (normalized between 0 and 1) values of citations and credit.

by the Family tuples than those in the other tables. The differences between the why- and how-provenance-based DSs are also relatively minor (about plus or minus 0.2 out of 9.5) in most cases. However, there are certain situations in which the role of a tuple is particularly critical in a query, and in this case the difference in the value of credit assigned is notably higher for how-provenance. An example of this can be seen in tuple 9 of the 10k group of Figure 10.

To sum up, the DS based on lineage is sufficient to highlight which tuples in the database are used by a query, and distributes credit equally to these tuples. The resulting distribution rewards tuples that are used by more queries, but does not reward how many times tuples are used in the same query. However, a DS based on why- or how-provenance may be better if the queries are complex, since they reward more tuples that have a critical role in generating the output. In particular, these two DSs may be useful for finding “hotspots” in the database based on the role of tuples, with the how-provenance-based DS being preferable if a higher sensitivity to the role of a tuple in queries is required.

6.4. Credit vs Citations

In the last set of experiments, we compare traditional citations to the proposed credit distribution strategies to see the difference in reward for data authors and curators. By using both real-world and synthetic queries,

we distribute credit to the authors responsible for the data being cited using the different strategies. This shows that credit is able to reward also authors of data that is cited fewer times, but with a higher impact on the submitted queries.

To do so, we need to identify a set of authors and queries that cite data curated by them. Considering GtoPdb, each target family page has a list of curators, representing the people who are co-creators and curators of the data comprising the page. This list can be obtained using the last query shown in Figure 3. Each time a target family page is cited, we assign one *citation* to each author associated with the page. The authors also receive *credit* in the amount assigned to the data used by the query to construct the webpage, equally divided between the authors of the webpage.

Results: Real-world queries. As described in Section 6.1, we consider real-world queries taken from papers published in the BJP which reference webpages in GtoPdb. Since for these queries there is no difference in the distribution of credit between the three DS, only one value for credit is used.

The results are shown in the radar plots of Figure 11, in which each number on the outer circle (e.g. 475, 1774 and 3665) represents an author (id) and the blue (red) line represents the normalized value of credit generated by citations (credit), respectively. The first radar plot, Figure 11.a, shows the top 20 authors in terms of *citations*, ordered in a clockwise direction, whereas Figure 11.b orders the authors based on *credit*. Comparing the author ids used in the outer circles of these two plots, it can immediately be seen that the “top authors” are very different using these two metrics, although there is some overlap (for example, authors 1774, 475, and 4012).

Diving a bit deeper to focus on the red and blue areas in each of the plots reveals that there is a significance difference between citations and credit: The top 20 authors in terms of citations do not have the highest values of credit (Figure 11.a). Conversely, the authors with the highest values of credit do not necessarily have a large number of citations (Figure 11.b). For example, author 536 has the highest value of credit, but is not even in the top 20 authors in terms of citations. This means that authors like 536, 822, and 3342 in Figure 11.b receive much more credit from their relatively few citations than authors like 475, who receives the largest number of citations. That is, the data underlying certain webpages is more “valuable” in terms of credit than a citation to the webpage.

The reason for the difference between citation and credit is partly due to

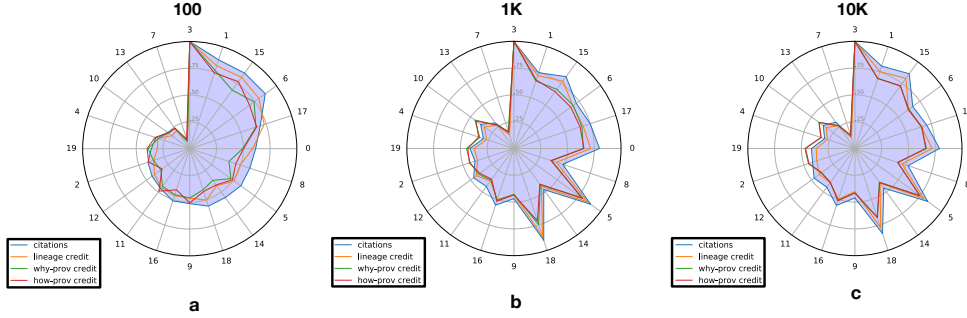


Figure 12: Radars presenting the 20 synthetic authors with corresponding citation and quantities of credit distributed through the 3 DS (all values normalized between 0 and 1) through different numbers of polynomials (respectively, 100, 1K and 10K). The order is the one defined by figure 1, i.e. descending order of citations obtained from 100 polynomials.

the experimental setup: Each output tuple carries a credit of 1, and there can be many tuples used to generate a webpage. Thus a webpage that is created from more tuples will have a higher credit value than one created from fewer tuples. Furthermore, authors who collaborated with fewer people will receive a biggest share of the equally divided credit. However, all authors will receive a citation of one.

Credit distribution therefore rewards authors differently than traditional citations: An author who has curated larger quantities of cited data and collaborated with fewer co-authors, will receive larger quantities of credit. Thus, credit rewards them for their larger contribution to the database.

Results: Synthetic queries. We produced 100, 1K, and 10K batches of synthetic polynomials, as described in Section 6.2, and distributed credit through them to data. Since these polynomials are created by randomly selecting tuples from three tables, they usually correspond to a large set of authors who in reality did not collaborate. To make the size of the author set more realistic, we therefore created 20 synthetic authors, and randomly assigned one author to blocks of consecutive tuples in the database, with the size of each block varying between 10 and 40, to simulate different quantities of work performed by an author. Every time an author appears as curator of one or more tuples used in a polynomial, we assign them one citation. They also receive three kinds of credit, each one using a different DS.

Figure 12 shows three radar plots, one for each batch of synthetic polynomials. Each plot shows the top 20 authors in terms of citations (hence

the authors and clockwise ordering is the same in each of the plots), and additionally shows the the normalized values of citation (blue line), lineage-based credit (yellow line), why-provenance-based credit (green line) and how-provenance-based (red line). As can be seen, given the synthetic nature of the queries, the correlation between the number of citations and the quantity of credit assigned to the authors appears to be a much stronger than with the real-world queries of Figure 11. In fact, for Figure 12.a the linear correlation between the citation number and all three types of credit is always above 0.95 with p values in the order of $1e-11$. The credit distributed via lineage is closest to the number of citations (a linear correlation of 0.98, p value of $6.15e-16$ in Figure 12.a), while the other two types of credit behave slightly differently (a linear correlation of around 0.95 in both cases in Figure 12.a). Similar observations can be made for Figure 12.b and 12.c.

What these figures show is that, in certain cases, authors who do not have a large number of citations receive more credit than others, as for example author 11 in Figure 12.a or author 19 in Figures 12.b and 12.c, especially when credit is distributed using how-provenance. This again shows how credit gives a different perspective on the role of data and authors by going beyond the limitations of traditional citations.

It is worth noting that, when scaling up to $1K$ and $10K$ polynomials, the credit distributions via why-provenance and how-provenance become almost identical (the linear correlation for the values of Figure 12.c is more than 0.99 with a p-value of $1.32e-32$). This is consistent with what we observed in Figure 9.

7. Conclusions and Future Work

This paper defines two new distribution strategies based on why- and how-provenance, and compares them against the lineage-based distribution strategy defined in [24]. The first, why-provenance-based DS, uses the concept of a witness, and gives more credit to tuples that appear in more than one witness. In this way, tuples that are more important to the query and are used in different ways are rewarded more. The second, how-provenance-based DS, considers the frequency with which a tuple or combination of tuples is used in the query through the information contained in a provenance polynomial. In this case, the how-provenance-based DS is more sensitive than the why-provenance-based DS to the role and importance of tuples.

958 To show the differences between the three DSs, we performed extensive
959 experiments based on GtoPdb, a curated scientific relational database, using
960 both real and synthetic queries. In the first set of experiments, we used select-
961 project-join (SPJ) queries extracted from citations to webpages in GtoPdb
962 found in papers published in the British Journal of Pharmacology. Using
963 these “real” queries, we distributed credit to tuples in different tables of the
964 database, highlighting tuples that were more frequently used. We showed
965 that, with these queries, the three strategies produce the same distribution.
966 This is because the SPJ queries were fairly simple, and did not use self-joins.
967 Therefore the formulas underlying the different DSs had the same output.

968 In the second set of experiments, we synthetically produced more com-
969 plex provenance polynomials, corresponding to more complex queries, that
970 resulted in exponents and coefficients in the provenance polynomials that
971 were greater than (or equal to) 1. These experiments highlighted the differ-
972 ences between the three DSs. While the DS based on lineage rewards all the
973 tuples used by a query equally, the strategy based on why-provenance gives
974 more credit to tuples that are more critical to the query. In particular, why-
975 provenance consider the different ways in which a tuple is used in a query.
976 How-provenance is even more sensitive to the tuple’s role: it also considers
977 the frequency with which a tuple or a set of tuples is used.

978 In the third set of experiments, we showed how the differences between
979 the DS are compounded over time, i.e. when more and more queries are
980 processed by the system.

981 In the fourth set of experiments we compared traditional citations to
982 authors to the credit accrued to them via the DSs. We showed how, in
983 both real-world and synthetic scenarios, credit rewards authors who con-
984 tribute/curate data that has the highest impact, and therefore receives the
985 biggest quantity of credit, and not necessarily the data with the highest ci-
986 tation count. In this sense, credit appears to be an useful new measure to
987 discover data and their corresponding curators that have a high impact in
988 the research world, even when they are cited few times or do not appear at
989 all in the data that are cited (i.e. the case of data used to build the output
990 of a query but that is not visualized in the output itself).

991 In future work, we plan to explore different strategies to generate and
992 distribute credit. In this paper we assumed that each output tuple carries
993 credit 1. In more sophisticated scenarios we can employ different strategies
994 to compute credit, that reflect the importance of cited data. Also, other,
995 and more sophisticated strategies could also be used to decide how credit is

distributed between the authors, beyond the uniform distribution used here,
in a way to reflect the work performed by them on the cited data.

We will also explore new applications for credit over relational databases.
One example is *data pricing*, which gives a price to a query submitted by a
user who wants to buy the produced information. Currently, a commonly
strategy used for data pricing is based on query rewriting: A database stores a
set of views with their price. When a new query arrives, the system rewrites
it using the stored views to obtain a query price, a process that can be
computationally expensive. We plan to distribute credit through carefully
planned and representative queries, and use credit information to define a
new, faster, and potentially more flexible pricing function.

Another application is *data reduction* [41], which addresses the problem
of reducing the vast – and rapidly expanding – amount of data that is being
produced.

Data credit can also address this problem, by helping find “hotspots”
and “coldspots” of data. A hotspot is data in a database (e.g. a tuple) with
a high quantity of credit, which is therefore valuable for the set of queries
that execute frequently over the data and distribute the credit. On the other
hand, a coldspot is data with a low quantity of credit, which is therefore
considered less important and could be deleted or moved to cheaper and/or
less efficient memory.

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.

- 1031 [4] Altman, M., Borgman, C. L., Crosas, M., and Martone, M. (2015). An
1032 Introduction to the Joint Principles for Data Citation. *Bulletin of the*
1033 *Association for Information Science and Technology*, 41(3):43–45.
- 1034 [5] Baggerly, K. (2010). Disclose all data in publications. *Nature*,
1035 467(7314):401–401.
- 1036 [6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D.,
1037 Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I.,
1038 Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and
1039 Goble, C. A. (2013). Why linked data is not enough for scientists. *Future*
1040 *Gener. Comput. Syst.*, 29(2):599–611.
- 1041 [7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation
1042 Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.
- 1043 [8] Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The
1044 million song dataset. In *Proceedings of the 12th International Conference*
1045 *on Music Information Retrieval (ISMIR 2011)*, pages 591–596.
- 1046 [9] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In
1047 Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Commu-*
1048 *nication*, pages 93–116. De Gruyter Mouton.
- 1049 [10] Buneman, P. (2006). How to cite curated databases and how to make
1050 them citable. In *18th International Conference on Scientific and Statistical*
1051 *Database Management, SSDBM*, pages 195–203. IEEE Computer Society.
- 1052 [11] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D.,
1053 Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn’t
1054 working, and what to do about it. *Database J. Biol. Databases Curation*,
1055 2020.
- 1056 [12] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation
1057 is a computational problem. *Commun. ACM*, 59(9):50–57.
- 1058 [13] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A
1059 characterization of data provenance. In *Database Theory - ICDT 2001,*
1060 *8th International Conference*, pages 316–330.

- 1061 [14] Buneman, P. and Silvello, G. (2010). A rule-based citation system for
1062 structured and evolving datasets. *IEEE Data Eng. Bull.*, 33(3):33–41.
- 1063 [15] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N.,
1064 Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry,
1065 R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a.,
1066 and Wright, D. (2012). Making Data a First Class Scientific Output:
1067 Data Citation and Publication by NERC’s Environmental Data Centres.
1068 *International Journal of Digital Curation*, 7(1):107–113.
- 1069 [16] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Jour-
1070 nals: A Survey. *Journal of the Association for Information Science and*
1071 *Technology*, 66(9):1747–1762.
- 1072 [17] Cheney, J., Chiticariu, L., and Tan, W. (2009). Provenance in databases:
1073 Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–
1074 474.
- 1075 [18] CODATA-ICSTI Task Group on Data Citation Standards and Practices
1076 (2013). *Out of Cite, Out of Mind: The Current State of Practice, Policy,*
1077 *and Technology for the Citation of Data*, volume 12.
- 1078 [19] Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., and Simons, N.
1079 (2019). Bringing citations and usage metrics together to make data count.
1080 *Data Science Journal*, 18(1).
- 1081 [20] Cronin, B. (1984). *The citation process. The role and significance of*
1082 *citations in scientific communication*. London: Taylor Graham.
- 1083 [21] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evi-
1084 dence of a structural shift in scholarly communication practices? *JASIST*,
1085 52(7):558–569.
- 1086 [22] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of
1087 view data in a warehousing environment. *ACM Trans. Database Syst.*,
1088 25(2):179–227.
- 1089 [23] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model
1090 for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on*
1091 *Innovative Data Systems Research*. www.cidrdb.org.

- 1092 [24] Dosso, D. and Silvello, G. (2020). Data credit distribution: A
1093 new method to estimate databases impact. *Journal of Informetrics*,
1094 14(4):101080.
- 1095 [25] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The vir-
1096 tual atomic and molecular data centre (VAMDC) consortium. *Journal of*
1097 *Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.
- 1098 [26] ESIP Data Preservation and Stewardship Committee (EDPSC) (2019).
1099 Data citation guidelines for earth science data, version 2. Version 2, Earth
1100 Science Information Partners.
- 1101 [27] Fang, H. (2018). A discussion of citations from the perspective of the
1102 contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–
1103 1520.
- 1104 [28] Garfield, E. (1999). Journal impact factor: a brief review. *Can. Med.*
1105 *Assoc.*, 979-980.
- 1106 [29] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data
1107 identification and process monitoring for reproducible earth observation
1108 research. In *2019 15th International Conference on eScience (eScience)*,
1109 pages 28–38. IEEE.
- 1110 [30] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance
1111 semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-*
1112 *SIGART symposium on Principles of database systems*, pages 31–40. ACM.
- 1113 [31] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson,
1114 A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton,
1115 S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F.,
1116 Spedding, M., Davies, J. A., and Nc-Iuphar (2018). The IUPHAR/BPS
1117 guide to PHARMACOLOGY in 2018: updates and expansion to encom-
1118 pass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Re-*
1119 *search*, 46(Database-Issue):D1091–D1106.
- 1120 [32] Hartley, J. (2017). Authors and their citations: a point of view. *Scien-*
1121 *tometrics*, 110(2):1081–1084.
- 1122 [33] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a
1123 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] Kosten, J. (2016). A classification of the use of research indicators. *Scientometrics*, 108(1):457–464.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] Milo, T. (2019). Getting rid of data. *Journal of Data and Information Quality (JDIQ)*, 12(1):1–7.
- [42] 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.

- 1157 [43] Parsons, M. A., Duerr, R. E., and Jones, M. B. (2019). The history and
1158 future of data citation in practice. *Data Science Journal*, 18(1).
- 1159 [44] Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J.
1160 (2016). Research data explored: An extended analysis of citations and
1161 altmetrics. *Scientometrics*, 107(2):723–744.
- 1162 [45] Pröll, S. and Rauber, A. (2013). Scalable data citation in dynamic, large
1163 databases: Model and reference implementation. In *Proceedings of the*
1164 *2013 IEEE International Conference on Big Data*, pages 307–312. IEEE.
- 1165 [46] Rauber, A., Ari, A., van Uytvanck, D., and Pröll, S. (2016). Identifi-
1166 cation of Reproducible Subsets for Data Citation, Sharing and Re-Use.
1167 *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue*
1168 *on Data Citation*, 12(1):6–15.
- 1169 [47] Rauber, A., Asmi, A., van Uytvanck, D., and Proell, S. (2015). Data
1170 citation of evolving data: Recommendations of the working group on data
1171 citation (wgdc). *Result of the RDA Data Citation WG*, 20.
- 1172 [48] Silvello, G. (2018). Theory and practice of data citation. *J. Assoc. Inf.*
1173 *Sci. Technol.*, 69(1):6–20.
- 1174 [49] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data
1175 provenance in e-science. *SIGMOD Record*, 34(3):31–36.
- 1176 [50] Spengler, S. (2012). Data Citation and Attribution: A Funder’s Per-
1177 spective. In of Sciences’ Board on Research Data, N. A. and Information,
1178 editors, *Report from Developing Data Attribution and Citation Practices*
1179 *and Standards: An International Symposium and Workshop*, pages 177–
1180 178. National Academies Press: Washington DC.
- 1181 [51] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G.,
1182 Axton, M., Baak, A., Blomberg, N., Boiten, J., da Silva Santos, L. B.,
1183 Bourne, P. E., et al. (2016). The fair guiding principles for scientific data
1184 management and stewardship. *Scientific data*, 3.
- 1185 [52] Wu, Y., Alawini, A., Davidson, S. B., and Silvello, G. (2018). Data
1186 citation: Giving credit where credit is due. In *Proceedings of the 2018*
1187 *International Conference on Management of Data, SIGMOD*, pages 99–
1188 114.

- 1189 [53] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to
1190 scientific datasets using article citation networks. *Journal of Informetrics*,
1191 14(2).
- 1192 [54] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of
1193 new researchers using the zp-index. *Scientometrics*, 106(3):901–916.
- 1194 [55] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for
1195 Datasets Citation and Extraction Reproducibility in VADMC. *Journal of*
1196 *Molecular Spectroscopy*, 327:122–137.