

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, USA*

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 database parts 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 widely-used 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 frequently used database areas and how it be used as a new bibliometric measure for data and their corresponding curators. In particular, credit rewards data and authors based on their research impact, 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 an input tuple in the generation of the output data and reward input tuples differently.

Keywords: Data Citation, Data Credit

1. Introduction

Citations are an essential component of scientific research, enabling research products to be found as well as the relationships between research products to be created and understood. They form a basis on which to give credit to authors, papers, and venues [19, 20, 54]. Citations are used, among other things, to decide on tenure, promotion, hiring, and funding of grants for researchers [21, 32, 37, 40].

Science and research are increasingly digital, and 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 [15, 38], 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, at different levels, have been promoted 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 data and 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].

Moreover, there are activities to promote and specify guidelines for data citations. A significant activity getting a broad adoption, is the Research Data Alliance (RDA), that produced a recommendation on citing specific subsets of dynamic data [47]. While this approach provides reference and access to a precise subset of data, it does not address specific credit concerns for that subset, such as when different authors contribute to a larger collection [43].

A central problem in the data citation process 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 [9, 28]. However, even when correctly applied, data citations and the bibliometrics 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. When a paper p_1 cites another paper p_2 , a +1 citation “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

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

- 70 role played by p_2 in p_1 . Hence, we can weight the importance of the
 71 cited entities and assign credit according to their role.
- 72 2. Traditional citations are *atomic*: a citation from p_1 to p_2 can never
 73 be broken into pieces and assigned in part to p_2 and in part to other
 74 papers or data that contributed to p_2 . In contrast, with data credit,
 75 we use a *non-atomic* real value, which can be divided and distributed
 76 to multiple components of a database.
- 77 3. Credit can be *transitive*, that is, it can be propagated through one
 78 cited entity to other entities cited by it that contributed to its content.
 79 Citations, traditionally, are not.

80 We study the DCD problem in the context of relational databases (RDBs)
 81 since they are widely used ² and are the main focus of current work in data
 82 citation methods [12, 14, 45]. RDBs are also frequently a test-bed for new
 83 methods that can be adapted to other databases, e.g., graphs or document
 84 databases. The “portions” of data in an RDB that can be credited can be
 85 defined at different levels of granularity, in particular: (i) the whole database,
 86 (ii) tables, (iii) tuples, and (iv) attributes. The ability to specify different
 87 levels of granularity in a relational database allows us to define the DCD
 88 problem at a particular level of granularity. In this paper, we focus on DCD
 89 at the tuple level.

90 The DCD process is summarized in Figure 1:

91 **Step 1** Scientists and experts contribute the curated information contained
 92 in a scientific database. These are called the “Data Curators”.

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

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

98 **Step 4** Given the database instance I and the query Q , it is possible to
 99 compute the *data provenance* of $Q(I)$. The provenance of $Q(I)$ is a

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

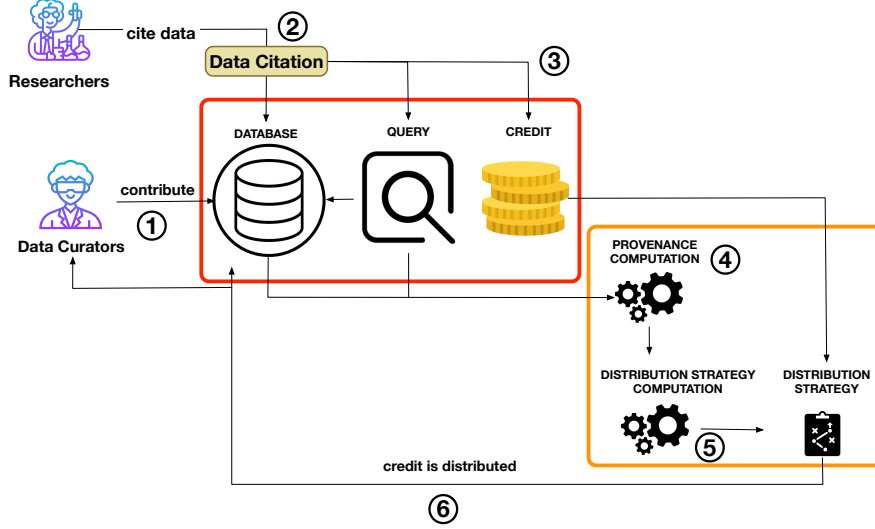


Figure 1: Overview of the credit distribution pipeline.

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 [13, 22, 30], describing different kinds of relationships between data in the input and the output of a query. As reported in [17], these provenances have been used in several applications beyond giving information on how queries work, for example, annotation propagation and the view update problem. In this paper, we consider three types of provenance: lineage, why-provenance, and how-provenance.

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

Step 6 Once the CDS is computed, it is used to distribute the given credit k to the parts of the database that are responsible for the generation

119 of $Q(I)$. Transitively, this credit is also divided and given to the corre-
120 sponding authors of those data.

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

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

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

148 For evaluation, we use a well-known curated database, the IUPHAR/BPS³
149 Guide to Pharmacology [31], also known as GtoPdb⁴, which contains ex-
150 pertly curated information about diseases, drugs, cellular drug targets, and

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

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

151 their mechanisms of action. We chose GtoPdb for two main reasons: (i) it
 152 is a widely-used and valuable curated relational database, (ii) many papers
 153 in the literature use, and cite its data (i.e., families, ligands, and receptors).
 154 Real queries used in papers can therefore be seen as data citations which, in
 155 turn, can be used to assign data credit.

156 We perform four sets of experiments. In the first one, real queries are ex-
 157 tracted from papers published in the British Journal of Pharmacology (BJP),
 158 that represent data citations to GtoPdb, and are used to distribute credit
 159 in the database using the three different provenance-based DSs. In the sec-
 160 ond and third experiment we analyse the behaviour of the different DS when
 161 complex citation queries are employed. In the fourth set of experiments we
 162 use both real and synthetic queries to assess the difference between tradi-
 163 tional citation and the notion of credit distribution in terms of rewarding
 164 those responsible for the data, e.g. data curators.

165 **Contributions** of this work include:

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

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

179 2. Background

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

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

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

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

- 210 • *Data attribution*: identify the individuals that should be credited for
 211 data with variable granularity.
- 212 • *Data connection*: connect papers to the data being used.
- 213 • *Data Discovery*: citations helps to find data records and subsets that
 214 would be otherwise not findable via search engines.
- 215 • *Data Sharing*: share data obtained by researchers within the whole
 216 community.

- *Data Impact*: highlight the results obtained in writing papers using specific data, the frequency and modality data were used.
- *Reproducibility*: data citation greatly impacts the reproducibility of science [5]. Many authoritative journals ask to share data and provide valid methodologies to reproduce experiments.

2.1. Data Citation in Relational Databases

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

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

More examples of work on data citation in relational databases are [2, 12, 23, 52]. The website <https://fairsharing.org/> keeps a long updated list of curated and scientific databases (many of which are relational or graph-based) following FAIR guidelines. These databases are citable since they are compliant with the most recent guidelines, and they are in the vast majority of cases accessible via dynamically created Webpages. In all these databases is, therefore, possible to implement DCD on top of the existing infrastructures for citing data.

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>

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 [2, 10, 12] 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 products as software and data. In the paper two considerations are made: (i) research objects such as data and software are currently not formally rewarded or recognized by the community; (ii) even in traditional papers, the contribution of each author to the work is hard to understand, unless explicitly specified in the paper. This is even more true for data, where different groups of people work on the same database.

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

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

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

312 Fang [27] presents a framework to distribute the credit generated by a
313 paper to its authors and to the papers in its reference list in a transitive way.
314 Let us consider the *citation graph* as the graph where the nodes are papers
315 and the links are the citations among them. In this graph, every paper is
316 a source of credit, which is then transferred to the neighboring nodes. The
317 quantity of credit received by each cited paper depends on its impact/role
318 in the citing paper. So far, this theoretical framework is limited to papers,
319 but it can be easily extended to a citation graph including both papers and
320 data.

321 Zeng et al. [53] proposes the first method to compute credit within a net-
 322 work of papers citing data. Adopting a network flow algorithm, they simulate
 323 a random walker to estimate a score for each dataset, leveraging real-world
 324 usage data to compute the credit. This is the first step towards an automatic
 325 credit computation procedure. This proposal is, however, limited to assign-
 326 ing credit to whole datasets, and it does not deal with the granularity of data.
 327 It does not work to assign credit to a single research entity within a dataset.
 328 Differently from Zeng et al. [53], we do not treat the credit computation
 329 process, but we focus on the distribution process.

330 2.3. Data Provenance

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

341 Cheney et al. [17] presents four main types of data citation for DBMS: *lin-*
 342 *age* [22], *why-provenance* [13], *how-provenance* [30] and *where-provenance* [13].

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

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

353 The definition of why-provenance is based on the notion of *witness set*.
 354 A witness is a set of relevant tuples that guarantees the existence of t in
 355 $Q(D)$. The lineage is therefore an example of a witness. The why-provenance
 356 of a tuple t is a peculiar set of witnesses – described in [13] – that are
 357 computed from the query, called *witness basis*. A witness basis may be

358 composed of more than one witness. Therefore, the why-provenance contains
359 more information than the lineage, since it describes *alternative* ways in
360 which the same output may be generated.

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

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

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

374 3. Use Case: GtoPdb

375 As use case we refer to the IUPHAR/BPS Guide to Pharmacology [31]
376 or GtoPdb⁹. GtoPdb is a well-known and well structured scientific relational
377 database that contains expertly curated information about diseases, drugs
378 in clinical use, their cellular targets, and the mechanisms of action on the
379 human body. It is curated and maintained by the GtoPdb Committee, and
380 by 96 subcommittees, comprising 512 scientists collaborating with in-house
381 curators who draw the information contained in the database from high-
382 quality pharmacological and medicinal chemistry literature. Roughly 1000
383 researchers from all over the world have contributed to the database, and the
384 curators wanted to give recognition to these contributors. This led to some
385 early work on data citation [10].

386 GtoPdb is relational, but its logical structure is hierarchical as shown
387 in Figure 2. The information contained in the database is also organized
388 into webpages focused on specific diseases, targets or ligands, and families
389 for easier access by users. As depicted in Figure 2, the database can be
390 thought of as a tree where the root is the database; the first level consists

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

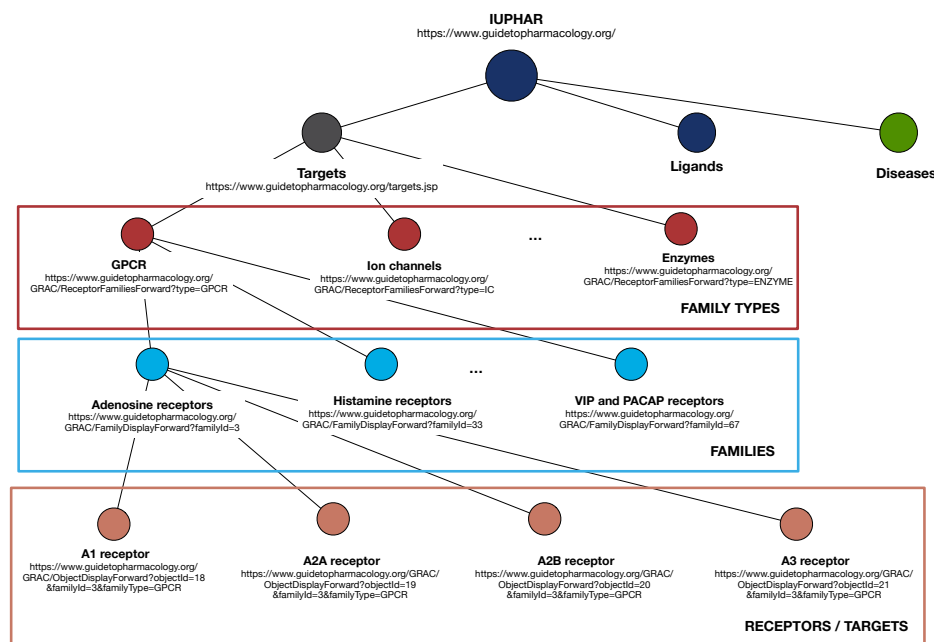


Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

of all targets, ligands, and diseases; and the lower levels consists of specific targets, ligands and diseases. In this paper, we focus on targets; thus at the third level in the figure we show examples of family types, at the fourth level we show specific families of targets (a finer level of granularity), and finally, at the last level, the single targets (also known as receptors).

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

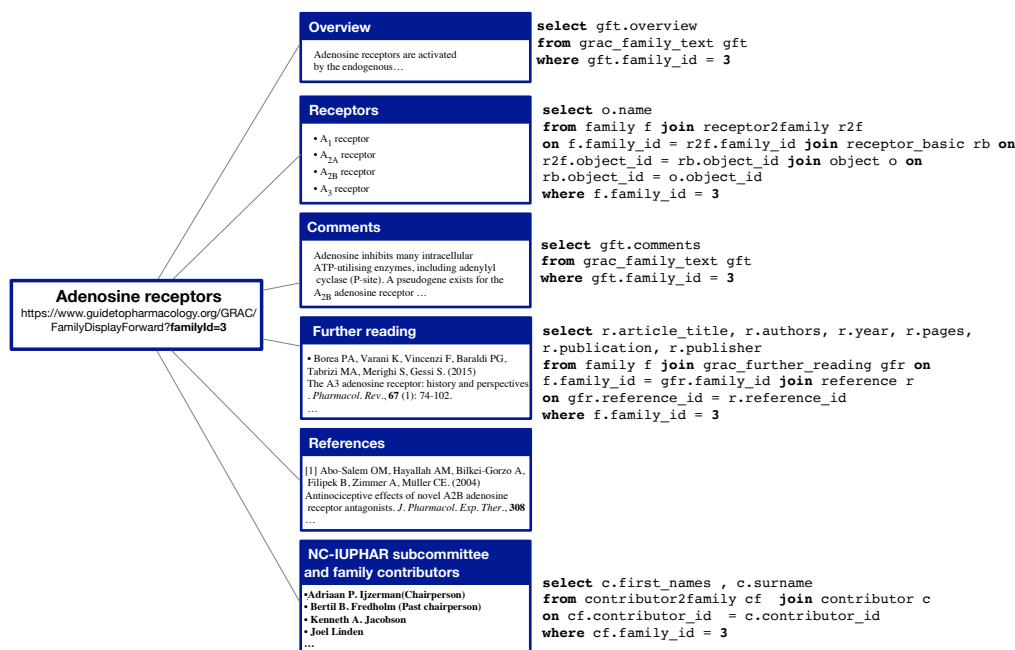


Figure 3: Basic web-page structure of “Adenosine receptors” family (ID 3), with queries used to retrieve the information contained in every section, except references.

family page can be considered a full-fledged traditional publication, consisting of title, authors, abstract (the overview), content, and references.

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

In certain “lucky” cases, as with papers available in PDF and published in the British Journal of Clinical Pharmacology¹⁰ (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

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

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.

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.

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

441 the input that contributed to its “production” [17]. As an example, consider
 442 the following SQL query Q1, applied to the database described in Table 1,
 443 that asks for the names of families curated by researchers based in the United
 444 Kingdom (UK):

```

445     Q1: SELECT DISTINCT f.name
446     FROM family AS f JOIN contributor2family AS c2f
447     ON f.id = c2f.family_id
448     JOIN contributor AS c ON c2f.contributor_id = c.id
449     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.

450 Table 2 shows the query result set, which consists of two tuples. We add
 451 an extra attribute *id* so that we can easily refer to each result tuple. The
 452 lineage for tuple o_1 is the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$, since the tuple f_1 was
 453 joined with $c2f_1$ and then with c_1 , and was also joined with $c2f_2$ and c_2 . No
 454 other tuple is used in the database to produce o_1 . For tuple o_2 the lineage is
 455 $\{f_4, c2f_4, c_1\}$. Lineage is defined for each tuple of the output, and can differ
 456 between tuples.

457 4.2. Why-Provenance

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

462 In particular, while lineage aims to find all and only the tuples in the
 463 input relevant to the production of an output tuple, why-provenance aims to
 464 find sub-instances of the input that “witness” a part of the output. Given a
 465 tuple t in the query’s output, a *witness* is any sub-instance of the database
 466 that produces t . In particular, the whole database and the lineage of t are
 467 both witnesses of t . Since the definition of witness allows for the presence

of “irrelevant” tuples, the set of all witnesses is finite (since the database instance I is finite), but it is potentially exponentially large [17].

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

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

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

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

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

4.3. How-Provenance

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

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.

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

502 Table 4 shows a simple example in which the two output tuples of our
 503 running example are annotated with their respective how-provenances. Tuple
 504 o_2 was produced through the join among the input tuples $f_4, c2f_4$, and c_1 .
 505 The three provenance tokens are, therefore “multiplied” together. The case of
 506 o_1 is slightly more complex. This tuple, as already discussed, can be obtained
 507 through two different joins. The two monomials composing the polynomial
 508 represent these two alternatives. They correspond, in a way, to the witnesses
 509 of the why-provenance of o_1 . The $+$ operator represents the fact that the two
 510 monomials describe alternative derivations. The output tuple is the result
 511 of a merge of two distinct tuples after the projection on the attribute **name**.
 512 This merge is due to the fact that the result of a relational algebra expression
 513 is always a *set* of tuples, which corresponds to the presence of the **DISTINCT**
 514 operator in an SQL query. This simple example gives the basic idea behind
 515 how-provenance and how it allows us to track the operations that produced
 516 an output tuple.

517 Provenance polynomials may also have monomials whose exponents and/or
 518 coefficients are greater than one, for example, $3f_1 \cdot c2f_1 \cdot c_1 + f_1 \cdot c2f_2^3 \cdot c_2^3$.
 519 This is a polynomial of a tuple produced by a query where the result of the
 520 join between the tuples $f_1, c2f_1$, and c_1 is produced three times and then
 521 merged (e.g. as the result of a union), and the tuples $c2f_2$ and c_2 are used
 522 three times in the operation described by the second monomial (e.g., with
 523 nested queries).

524 5. Credit Distribution and Distribution Strategies

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

530 5.1. Data Credit and Data Credit Distribution

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

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

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

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

546 **Definition 5.1. Tuple Level Data Credit Distribution (DCD) [24]**

547 *Given a query Q over I and $k \in \mathbb{R}_{>0}$, DCD is defined by the function $f_{I,Q} :$
 548 $\text{TupleLoc} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$ such that $f_{I,Q}(t, k) = h$ where $0 \leq h \leq k$ and
 549 $\sum_{t \in \text{TupleLoc}} f_{I,Q}(t, k) = k$. The function $f_{I,Q}$ is the distribution strategy (DS).*

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

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

560 *5.2. A Lineage-based Distribution Strategy*

561 In the lineage-based distribution strategy, each tuple in the output of
 562 a query distributes credit equally to each input tuple that appears in its
 563 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:

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

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

568 As an example, consider the output tuples of Table 2, and assume that
 569 each output tuple has credit $k_o = 1$. The lineage of the first tuple, o_1 , is
 570 the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$. Therefore, each tuple in this set receives credit
 571 $1/5$. The other tuples of the database receive zero credit. The lineage of the
 572 second output tuple is $\{f_4, c2f_4, c_1\}$, therefore each of these tuples receives
 573 credit $1/3$.

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

580 Not all of the tuples in the lineage of an output tuple are necessary to be
 581 present at the same time for the output tuple to appear in the query results.
 582 For example, if the database only had the set of tuples $\{f_1, c2f_1, c_1\}$ or the set
 583 $\{f_1, c2f_2, c_2\}$, the existence of o_1 would still be guaranteed. In other words,
 584 while f_1 is always needed for o_1 to appear in the output, only one of the sets
 585 of tuples $\{c2f_1, c_1\}$ and $\{c2f_2, c_2\}$ is required. One could therefore argue that
 586 it would be more fair for f_1 to receive more credit than the other four tuples,
 587 given its role in producing o_1 .

588 This highlights one limitation of the lineage-based DS: while able to find
 589 all and only the relevant tuples of the output, it does not distinguish the

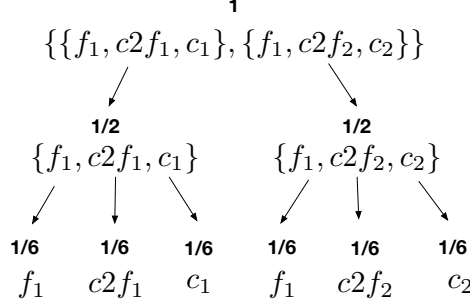


Figure 4: Distribution of credit using why-provenance-based DS for tuple o_1 .

590 *importance* of tuples in the query computations. We therefore present two
 591 other, more sophisticated, forms of distribution strategies based on why- and
 592 how-provenance.

593 5.3. A Why-Provenance-Based Distribution Strategy

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

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

600 *Let I be a database instance, Q a query over I , $o \in Q(I)$ an output tuple
 601 and k_o the total credit associated to o . Let $\mathcal{W} = \text{Why}(Q, I, o)$ be the witness
 602 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:

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

603 Figure 4 shows the distribution of credit with why-provenance-based DS
 604 for tuple o_1 . The credit is first equally divided between the two witnesses, so
 605 that both receive credit $1/2$. The credit is then further divided among the
 606 tuples in each witness. Since each witness has three tuples, each tuple in a

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

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

607 witness receives 1/6 of credit. At the end of the distribution, f_1 receives a
608 total credit of 1/3, and the other tuples receive 1/6 each. This distribution
609 better reflects the role of f_1 in the generation of o_1 since, as discussed earlier,
610 it is the only mandatory tuple for o_1 to appear in the output; only one of the
611 two other pairs of tuples are necessary for o_1 to appear in the result.

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

615 5.4. A How-Provenance Based Distribution Strategy

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

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

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

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

returns a set containing the monomials containing that tuple, if present in the polynomial; thus $\gamma(f_1, \mathcal{H}) = \{M_1, M_2\}$.

Definition 5.4. *How-Provenance-Based Distribution Strategy*

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

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

Going back to the example of Table 4, consider o_1 with provenance polynomial $f_1 c_2 f_1 c_1 + f_1 c_2 f_2 c_2$. The how-provenance-based DS firstly divides the credit between the two monomials. Since the coefficients of each monomial are 1, the credit is split in half. If they were, for example, 1 and 2 respectively, 1/3 of the credit would go to the first monomial, and 2/3 to the second. Since in our example each variable has exponent 1, the credit is further divided equally among the three variables. Thus, at the end of the computation, f_1 receives 1/3, and the other tuples receive 1/6. If, for example, the first monomial was $f_1^2 c_2 f_1 c_1$, then the portion of credit of this monomial would be divided in this way: 1/2 to f_1 and 1/4 to each of the other two tuples.

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

```
Q2: SELECT DISTINCT F.name
FROM family as F JOIN
(SELECT DISTINCT f.name AS name
FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
JOIN contributor AS c ON c2f.contributor_id = c.id
WHERE c.country = "UK") AS R ON F.name = R.name
WHERE F.type = "gpcr"
```

The result of Q2 is shown in Table 5, and consists of one tuple, annotated with each of the three provenances. As can be seen, lineage and why-provenance are identical to those of the tuple o_1 in the previous example. The how-provenance, however, is different since tuple f_1 is used twice: first in the join of the inner query, and second in the join of the outer query. This

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.

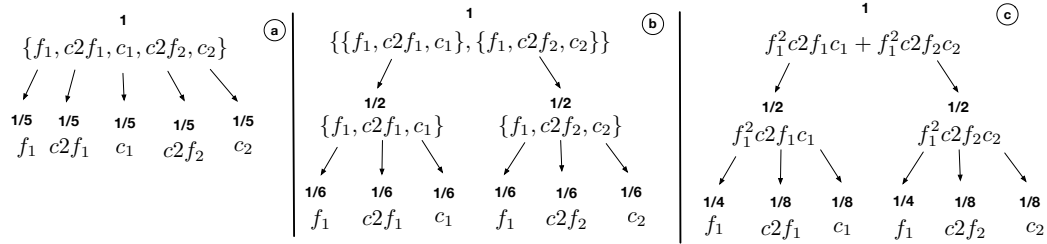


Figure 6: Comparison of different distributions strategies for tuple o_1 produced by query Q2.

information is lost in the first two forms of provenances since they are sets, but it is captured in how-provenance through the use of the operator ‘.’.

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

6. Experimental Evaluation

To understand the trade-offs between these Distribution Strategies (DSs), we perform four sets of experiments using queries over target families presented on the GtoPdb website. The first set of experiments use real queries extracted from citations to GtoPdb published in the British Journal of Pharmacology. The second set uses synthetically produced provenance polyno-

684 mials, corresponding to more complex queries, in order to better highlight
685 the differences between the DSs. The third set of experiments considers
686 the accrual of credit over time by the three strategies, again using synthetic
687 queries. The fourth set of experiments shows how the DSs compare to tradi-
688 tional citations in giving credit to data curators using both real and synthetic
689 queries.

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

693 6.1. *Real-world queries*

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

701 The queries that we inferred are those used to build target family web-
702 pages within GtoPdb. An example was given in Figure 3, where we show
703 how the structure of the “Adenosine receptors” family can be mapped into
704 queries over the underlying database. In GtoPdb, all target family pages
705 share a similar structure; the only difference is that individual sections, such
706 as “contributors” or “further readings”, may be absent. Therefore, the same
707 queries can be used to build all of the target family pages by changing the
708 family id used in the query (for example, in Figure 3, it is 3). Note that
709 the queries are fairly simple SQL queries, and fall into a class called “select-
710 project-join” or “SPJ” queries. A total of more than 12K different queries
711 were built in this way. Without loss of generality, we give each tuple in the
712 output of a query a credit of 1.

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.

713 *Results.* Figure 7 shows the heat-maps obtained by the distribution of credit
 714 according to the three different DS on one of the tables in the underlying
 715 database, **family**, which is often joined with other tables in the database to
 716 build the webpages. Each cell in a heat-map represents a tuple of the **family**
 717 table and the color indicates the amount of credit attributed to such tuple.
 718 It can be seen that the result of credit distribution over **family** is the same
 719 for all three strategies. The same result is also obtained with the other tables
 720 of the database used by the queries shown in Figure 3.

721 The reason why credit distribution is the same for all three strategies
 722 is that the queries are all simple SPJ queries, which use each table only
 723 once and do joins on key attributes. Under these conditions, each tuple of
 724 the output presents: (i) a how-provenance that is a single monomial with
 725 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.

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.

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.

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

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

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

777 Note that lineage-based DS gives the least credit to tuples in the `family`
 778 table, indicated by an overall lighter hue. This is because the DS distributes
 779 credit equally to all tuples appearing in the lineage. Since these queries also
 780 use two other tables, credit is distributed to tuples in those tables.

781 Moving to why-provenance-based DS, we see that more credit is given to
 782 tuples in the `family` table than with the previous strategy. This is because
 783 the DS considers the different ways that a tuple is used, e.g. in joins with
 784 other tuples. If the same tuple is present in more than one witness, it will
 785 draw more credit and take it from other tuples in the witness basis. In this

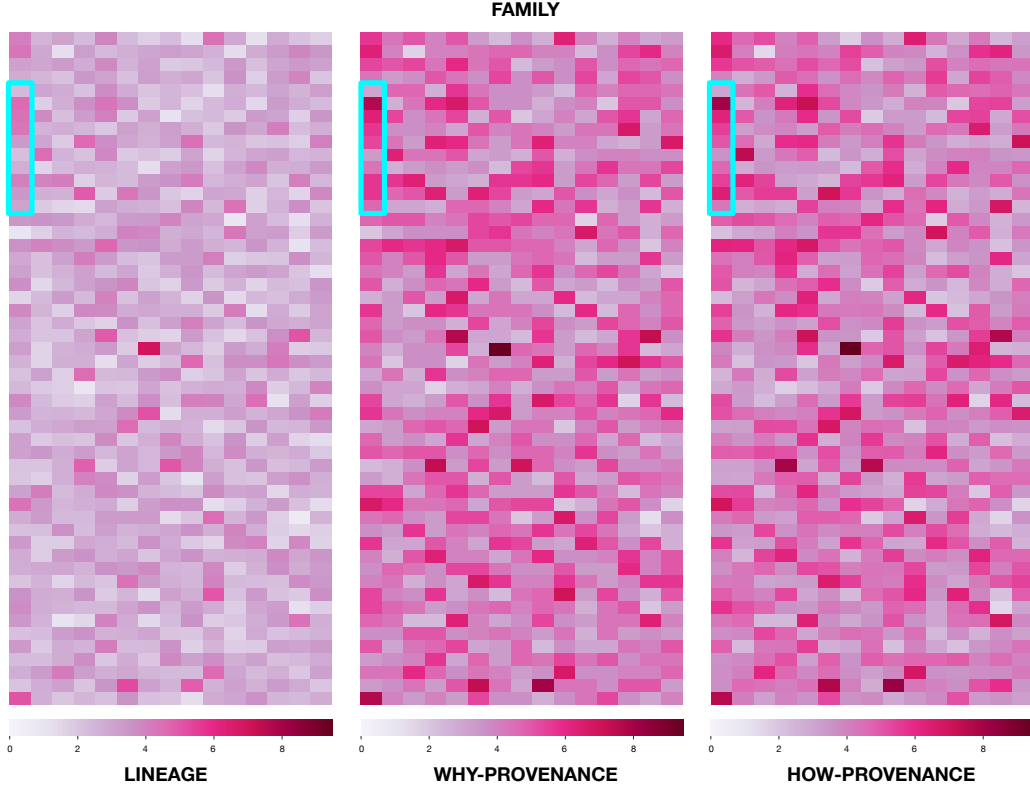


Figure 9: Comparison of three DS on the same table **family** after the distribution computed using 10K synthetic and randomly generated provenance polynomials. The tuples in the blue rectangles are used as example in the discussion connected to Figure 10.

case, tuples in **family** drew more credit, taking it from tuples in the other two tables, due to the role that **family** tuples played in the queries that were executed. We note that the lineage-based DS gives an average credit of 2.79 to each tuple in the table, while the DS based on why-provenance assigns 4.18. Moreover, lineage distributed a total of about 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

798 closer inspection, slight differences between the two distributions can be seen.
799 This is because how-provenance also considers the frequency with which tu-
800 ples are used, not only the ways in which they are used. Therefore, although
801 the overall distribution is similar, there are small differences due to the pres-
802 ence of exponents and coefficients in the provenance polynomials, influencing
803 the distribution of credit.

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

809 6.3. Credit accrual over time

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

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

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

822 By the end of 1K queries, credit differentials between tuples as well as
823 between strategies can be seen. For example, tuple 4 is usually rewarded the
824 most credit by all three strategies. However, it receives the highest quantity of
825 credit from the why-provenance-based strategy. Tuple 3 receives the highest
826 quantity of credit overall with how-provenance. This trend continues to the
827 end of 2k queries. By the end of 5k queries, tuple 2 emerges with the highest
828 value of credit for why- and how-provenance, a position which is strengthened
829 by the end of 10k queries. This is because tuple 2 is used several times
830 within queries being executed, which is rewarded strongly by why- and how-
831 provenance but not taken into account in lineage.

832 While the relative value of credit “positions” of tuples within a DS strat-
833 egy depends on what queries are being executed, the important thing to
834 notice is the difference between the DSs over time: Overall, lineage gives far



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.

less credit to tuples in the **family** table than the other two strategies since credit is shared with tuples in other tables. However, the why- and how-provenance-based strategies recognize the more important role being played 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

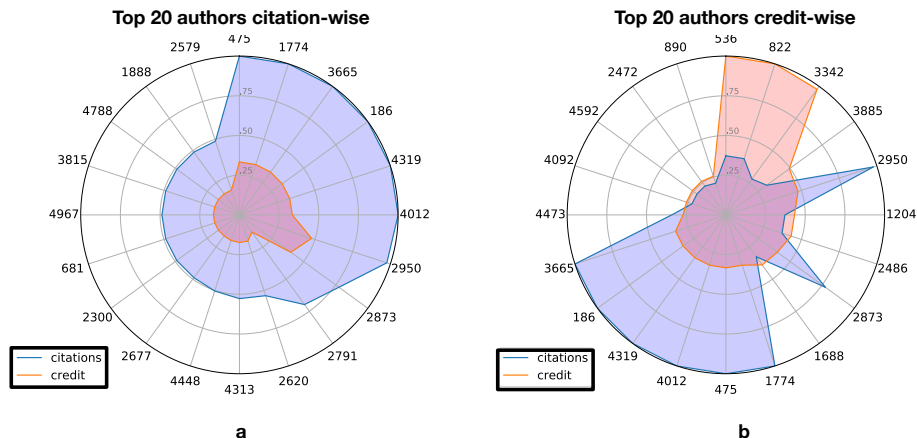


Figure 11: Radars presenting the top 20 authors citation-wise and credit wise, together with their (normalized between 0 and 1) values of citations and credit.

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. Using both real-world and synthetic queries, we distribute credit to the authors responsible for the data under the different strategies. Our results show that credit rewards authors of data that is cited fewer times, but that has a higher impact on the query results.

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

867 *citation* to each author associated with the page. The authors also receive
868 *credit* in the amount assigned to the data used by the query to construct the
869 webpage, equally divided between the authors of the webpage.

870 *Results: Real-world queries.* As described in Section 6.1, we consider real-
871 world queries taken from papers published in the BJP which reference web-
872 pages in GtoPdb. Since for these queries there is no difference in the distri-
873 bution of credit between the three DS, only one value for credit is used.

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

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

894 The reason for the difference between citation and credit is partly due to
895 the experimental setup: Each output tuple carries a credit of 1, and there
896 can be many tuples used to generate a webpage. Thus a webpage that is
897 created from more tuples will have a higher credit value than one created
898 from fewer tuples. Furthermore, authors who collaborated with fewer people
899 will receive a biggest share of the equally divided credit. However, all authors
900 will receive a citation of one.

901 Credit distribution therefore rewards authors differently than traditional
902 citations: An author who has curated larger quantities of cited data and
903 collaborated with fewer co-authors, will receive larger quantities of credit.

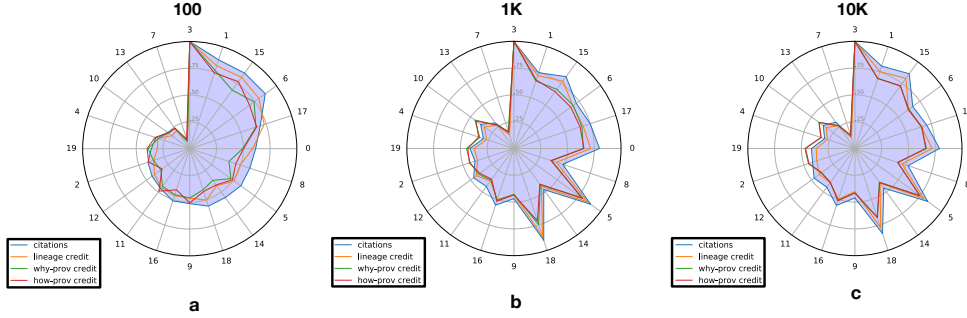


Figure 12: Radars presenting the 20 synthetic authors with corresponding citation and quantities of credit distributed through the 3 DS (all values normalized between 0 and 1) through different numbers of polynomials (respectively, 100, 1K and 10K). The order is the one defined by figure 1, i.e. descending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

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

927 is closest to the number of citations (a linear correlation of 0.98, p value of
 928 $6.15e-16$ in Figure 12.a), while the other two types of credit behave slightly
 929 differently (a linear correlation of around 0.95 in both cases in Figure 12.a).
 930 Similar observations can be made for Figure 12.b and 12.c.

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

937 It is worth noting that, when scaling up to $1K$ and $10K$ polynomials, the
 938 credit distributions via why-provenance and how-provenance become almost
 939 identical (the linear correlation for the values of Figure 12.c is more than
 940 0.99 with a p-value of $1.32e-32$). This is consistent with what we observed in
 941 Figure 9.

942 7. Conclusions and Future Work

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

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

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

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

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

986 In future work, we plan to explore different strategies to generate and
987 distribute credit. In this paper we assumed that each output tuple carries
988 credit 1. In more sophisticated scenarios we can employ different strategies
989 to compute credit, that reflect the importance of cited data. Also, other,
990 and more sophisticated strategies could also be used to decide how credit is
991 distributed between the authors, beyond the uniform distribution used here,
992 in a way to reflect the work performed by them on the cited data.

993 We will also explore new applications for credit over relational databases.
994 One example is *data pricing*, which gives a price to a query submitted by a
995 user who wants to buy the produced information. Currently, a commonly
996 strategy used for data pricing is based on query rewriting: A database stores a
997 set of views with their price. When a new query arrives, the system rewrites
998 it using the stored views to obtain a query price, a process that can be
999 computationally expensive. We plan to distribute credit through carefully
1000 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.

Acknowledgement

The work was partially supported by the ExaMode project, as part of the European Union H2020 program under Grant Agreement no. 825292.

References

- [1] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bernstein, P., Boncz, P., Chaudhuri, S., Cheung, A., Doan, A., Dong, L., Franklin, M. J., Freire, J., Halevy, A., Hellerstein, J. M., Idreos, S., Kossmann, D., Kraska, T., Krishnamurthy, S., Markl, V., Melnik, S., Milo, T., Mohan, C., Neumann, T., Chin Ooi, B., Ozcan, F., Patel, J., Pavlo, A., Popa, R., Ramakrishnan, R., Ré, C., Stonebraker, M., and Suciu, D. (2020). The seattle report on database research. *SIGMOD Rec.*, 48(4):44–53.
- [2] Alawini, A., Davidson, S. B., Hu, W., and Wu, Y. (2017). Automating data citation in citedb. *PVLDB*, 10(12):1881–1884.
- [3] Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., and Wu, Y. (2018). Data citation: A new provenance challenge. *IEEE Data Eng. Bull.*, 41(1):27–38.
- [4] Altman, M., Borgman, C. L., Crosas, M., and Martone, M. (2015). An Introduction to the Joint Principles for Data Citation. *Bulletin of the Association for Information Science and Technology*, 41(3):43–45.

- 1032 [5] Baggerly, K. (2010). Disclose all data in publications. *Nature*,
1033 467(7314):401–401.
- 1034 [6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D.,
1035 Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I.,
1036 Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and
1037 Goble, C. A. (2013). Why linked data is not enough for scientists. *Future*
1038 *Gener. Comput. Syst.*, 29(2):599–611.
- 1039 [7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation
1040 Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.
- 1041 [8] Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The
1042 million song dataset. In *Proceedings of the 12th International Conference*
1043 *on Music Information Retrieval (ISMIR 2011)*, pages 591–596.
- 1044 [9] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In
1045 Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Commu-*
1046 *nication*, pages 93–116. De Gruyter Mouton.
- 1047 [10] Buneman, P. (2006). How to cite curated databases and how to make
1048 them citable. In *18th International Conference on Scientific and Statistical*
1049 *Database Management, SSDBM*, pages 195–203. IEEE Computer Society.
- 1050 [11] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D.,
1051 Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn’t
1052 working, and what to do about it. *Database J. Biol. Databases Curation*,
1053 2020.
- 1054 [12] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation
1055 is a computational problem. *Commun. ACM*, 59(9):50–57.
- 1056 [13] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A
1057 characterization of data provenance. In *Database Theory - ICDT 2001*,
1058 *8th International Conference*, pages 316–330.
- 1059 [14] Buneman, P. and Silvello, G. (2010). A rule-based citation system for
1060 structured and evolving datasets. *IEEE Data Eng. Bull.*, 33(3):33–41.
- 1061 [15] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N.,
1062 Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry,

- 1063 R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a.,
 1064 and Wright, D. (2012). Making Data a First Class Scientific Output:
 1065 Data Citation and Publication by NERC’s Environmental Data Centres.
 1066 *International Journal of Digital Curation*, 7(1):107–113.
- 1067 [16] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Jour-
 1068 nals: A Survey. *Journal of the Association for Information Science and*
 1069 *Technology*, 66(9):1747–1762.
- 1070 [17] Cheney, J., Chiticariu, L., and Tan, W. (2009). Provenance in databases:
 1071 Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–
 1072 474.
- 1073 [18] CODATA-ICSTI Task Group on Data Citation Standards and Practices
 1074 (2013). *Out of Cite, Out of Mind: The Current State of Practice, Policy,*
 1075 *and Technology for the Citation of Data*, volume 12.
- 1076 [19] Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., and Simons, N.
 1077 (2019). Bringing citations and usage metrics together to make data count.
 1078 *Data Science Journal*, 18(1).
- 1079 [20] Cronin, B. (1984). *The citation process. The role and significance of*
 1080 *citations in scientific communication*. London: Taylor Graham.
- 1081 [21] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evi-
 1082 dence of a structural shift in scholarly communication practices? *JASIST*,
 1083 52(7):558–569.
- 1084 [22] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of
 1085 view data in a warehousing environment. *ACM Trans. Database Syst.*,
 1086 25(2):179–227.
- 1087 [23] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model
 1088 for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on*
 1089 *Innovative Data Systems Research*. www.cidrdb.org.
- 1090 [24] Dosso, D. and Silvello, G. (2020). Data credit distribution: A
 1091 new method to estimate databases impact. *Journal of Informetrics*,
 1092 14(4):101080.

- 1093 [25] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The vir-
1094 tual atomic and molecular data centre (VAMDC) consortium. *Journal of*
1095 *Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.
- 1096 [26] ESIP Data Preservation and Stewardship Committee (EDPSC) (2019).
1097 Data citation guidelines for earth science data, version 2. Version 2, Earth
1098 Science Information Partners.
- 1099 [27] Fang, H. (2018). A discussion of citations from the perspective of the
1100 contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–
1101 1520.
- 1102 [28] Garfield, E. (1999). Journal impact factor: a brief review. *Can. Med.*
1103 *Assoc.*, 979-980.
- 1104 [29] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data
1105 identification and process monitoring for reproducible earth observation
1106 research. In *2019 15th International Conference on eScience (eScience)*,
1107 pages 28–38. IEEE.
- 1108 [30] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance
1109 semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-*
1110 *SIGART symposium on Principles of database systems*, pages 31–40. ACM.
- 1111 [31] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson,
1112 A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton,
1113 S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F.,
1114 Spedding, M., Davies, J. A., and Nc-Iuphar (2018). The IUPHAR/BPS
1115 guide to PHARMACOLOGY in 2018: updates and expansion to encom-
1116 pass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Re-*
1117 *search*, 46(Database-Issue):D1091–D1106.
- 1118 [32] Hartley, J. (2017). Authors and their citations: a point of view. *Scien-*
1119 *tometrics*, 110(2):1081–1084.
- 1120 [33] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a
1121 transformed scientific method.
- 1122 [34] Honor, L. B., Haselgrove, C., Frazier, J. A., and Kennedy, D. N. (2016).
1123 Data citation in neuroimaging: proposed best practices for data identifi-
1124 cation and attribution. *Frontiers in neuroinformatics*, 10:34.

- 1125 [35] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E.,
1126 de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis,
1127 S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of bio-
1128 logical pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432.
- 1129 [36] Katz, D. (2014). Transitive credit as a means to address social and
1130 technological concerns stemming from citation and attribution of digital
1131 products. *Journal of Open Research Software*, 2(1).
- 1132 [37] Kosten, J. (2016). A classification of the use of research indicators.
1133 *Scientometrics*, 108(1):457–464.
- 1134 [38] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S.
1135 (2011). Citation and Peer Review of Data: Moving Towards Formal Data
1136 Publication. *International Journal of Digital Curation*, 6(2):4–37.
- 1137 [39] Martone, M. (2014). Joint declaration of data citation principles.
1138 *FORCE11. San Diego CA. Data Citation Synthesis Group*. [https://www.](https://www.force11.org/datacitationprinciples)
1139 [force11.org/datacitationprinciples](https://www.force11.org/datacitationprinciples), online September 2020.
- 1140 [40] Meho, L. I. and Yang, K. (2007). Impact of data sources on citation
1141 counts and rankings of LIS faculty: Web of science versus scopus and
1142 google scholar. *Journal of the american society for information science*
1143 *and technology*, 58(13):2105–2125.
- 1144 [41] Milo, T. (2019). Getting rid of data. *Journal of Data and Information*
1145 *Quality (JDIQ)*, 12(1):1–7.
- 1146 [42] Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D.,
1147 Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G.,
1148 Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff,
1149 D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D.,
1150 Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson,
1151 E., McNutt, M., Miguel, M., Paluck, E. L., Simonsohn, U., Soderberg, C.,
1152 Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers,
1153 E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research
1154 culture. *Science*, 348(6242):1422–1425.
- 1155 [43] Parsons, M. A., Duerr, R. E., and Jones, M. B. (2019). The history and
1156 future of data citation in practice. *Data Science Journal*, 18(1).

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

- 1187 [53] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to
1188 scientific datasets using article citation networks. *Journal of Informetrics*,
1189 14(2).
- 1190 [54] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of
1191 new researchers using the zp-index. *Scientometrics*, 106(3):901–916.
- 1192 [55] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for
1193 Datasets Citation and Extraction Reproducibility in VADMC. *Journal of*
1194 *Molecular Spectroscopy*, 327:122–137.