

Credit Distribution through Data Provenance in Relational Scientific Databases

Dennis Dosso^a, Susan B. Davidson^b, Gianmaria Silvello^a

^a*Department of Information Engineering, University of Padua, Italy*

^b*Department of Computer and Information Science, University of Pennsylvania, United States*

Abstract

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

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

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

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

Using different distribution strategies, we show how credit can highlight areas of a database that are frequently used, and how it can work as a new bibliometric measure for data and their corresponding curators. Credit in particular rewards data and authors based on their research impact, and not

merely on the number of citations. Also, we show how different distribution strategies, based on different types of data provenance, can be more sensible to the role of an input tuple in the generation of the output, and thus rewarding it differently.

Keywords: Data Citation, Data Credit

1 Introduction

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

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

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

Recently, the concepts of *data credit* and *Data Credit Distribution* (DCD) [26, 36, 54] have emerged, built on top of methodologies for data citation. Data

credit is a value that is computed based on the importance of the data being cited in a paper, and represents the impact of the data on the citing paper. The Data Credit Distribution problem consists of distributing this credit to elements in the databases in the citation graph that are responsible for the generation of the data being cited. The goal of DCD is to improve and expand the reach of data citation, rather than being an alternative to it. This means that to employ DCD techniques, we need data citations in some form.

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

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

The DCD problem differs from the traditional citation setting since:

1. In a traditional setting, when a paper cites another paper, a +1 “credit” is given to the cited paper (and to its authors). It does not matter why or how paper p_1 cites paper p_2 ¹, the result is always +1 from p_1 to p_2 and thus a +1 to the citation count of the authors of p_2 . With a different credit distribution strategy, the “value” given to the cited entity can be *proportional* to the role played in the citing entity. Hence, we can weigh the importance of the cited entities and assign credit according to their role.

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



Figure 1: Overview of the credit distribution pipeline.

2. Traditional citations are considered to be *atomic*. A citation from p_1 to p_2 can never be broken into pieces and assigned in part to p_2 and in part to other papers or data that contributed to p_2 . This is due to the intrinsic difficulty in grasping the role and “weight” of the other papers and data, and in automating the credit assignment process. In contrast, we consider data credit to be a *non-atomic* real value, which can be divided and distributed to multiple components of a database.
3. Credit can be *transitive*, that is, it can be propagated through one cited entity to other entities cited by it that contributed to its content.

We study the DCD problem in the context of relational databases (RDBs) since they are widely used² and are the main focus of current work in data citation methods [14, 12, 45]. RDBs are also frequently a test-bed for new methods that can be adapted to other databases, e.g., graphs or document databases. Furthermore, the “portions” of data in an RDB that can be credited can be defined at different levels of granularity, in particular: (i) the whole database, (ii) tables, and (iii) tuples.

The DCD process is summarized in Figure 1:

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

- 82 **Step 1** Scientists and experts contribute the curated information contained
83 in a scientific database. These are called the “Data Curators”.
- 84 **Step 2** Other researchers use the data in their research, and when possible,
85 cite them.
- 86 **Step 3** The citation to the data generates credit, that can be used as a
87 proxy for the impact of the data on the citing paper. This credit is
88 represented as a real value $k \in \mathbb{R}_{>0}$.
- 89 **Step 4** Given the database instance I and the query Q , it is possible to
90 compute the *data provenance* of $Q(I)$. The provenance of $Q(I)$ is a
91 form of metadata that describes the generation process undertaken by
92 Q , and the data used in I to generate the output [17]. Many different
93 notions of provenance have been proposed in the literature for data in
94 database management systems [22, 13, 30], describing different kinds
95 of relationships between data in the input and the output of a query.
96 As reported in [17], these provenances have been used in several appli-
97 cations beyond giving information on how queries work, for example,
98 annotation propagation and the view update problem. In this paper,
99 we consider three types of provenance: lineage, why-provenance, and
100 how-provenance.
- 101 **Step 5** Provenance is input to the CDC problem, whose aim is to compute
102 the *Credit Distribution Strategy* (CDS, also referred only as Distribu-
103 tion Strategy, DS). The CDS is a function that distributes k to the data
104 in the input database I , and is defined on the basis of citation policies
105 decided at the database administration level or at the domain commu-
106 nity level. In this paper, since we base CDS on data provenance, we
107 describe three CDS, each one based on a different form of provenance.
- 108 **Step 6** Once the CDS is computed, it is used to distribute the given credit
109 k to the parts of the database that are responsible for the generation
110 of $Q(I)$. Transitively, this credit is also divided and given to the corre-
111 sponding authors of those data.

112 This paper expands our recent work in [24], which addressed the problem
113 of how to reward data and data curators who are typically overlooked in
114 current citation systems. In that work, we first defined the problem of DCD

115 in relational databases, and proposed a viable Distribution Strategy (DS)
 116 based on *lineage*, which is the simplest form of *data provenance*. The lineage
 117 of a tuple t in the output $Q(I)$ is defined as the set of all and only the tuples
 118 in the database instance I that are “relevant” to the production of t , that
 119 is the tuple that are used by Q in the production of t . The lineage-based
 120 strategy equally redistributes the credit k to the tuples in the lineage set,
 121 thus each tuple receives credit $k/|L_t|$, where L_t is the lineage set of t .

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

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

139 For evaluation, we use a well-known curated database, the IUPHAR/BPS³
 140 Guide to Pharmacology [31], also known as GtoPdb⁴, which contains ex-
 141 pertly curated information about diseases, drugs, cellular drug targets, and
 142 their mechanisms of action. We chose GtoPdb for two main reasons: (i) it
 143 is a widely-used and valuable curated relational database, (ii) many papers
 144 in the literature use, and cite its data (i.e., families, ligands, and receptors).
 145 Real queries used in papers can therefore be seen as data citations which, in
 146 turn, can be used to assign data credit.

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

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

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

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

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

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

159 **Outline.** The rest of the paper is organized as follows: Section 2 presents the
 160 background and related work. Section 3 describes the use case we adopted.
 161 Section 4 briefly presents the forms of provenance used in the paper. Section
 162 5 describes the problem of DCD and the proposed DS. In Section 6 we present
 163 the experimental evaluation. Finally, Section 7 draws some conclusions and
 164 outlines future work.

165 2. Background

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

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

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

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

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

208 2.1. Data Citation in Relational Databases

209 In this paper, we develop our methods and experiments on relational
 210 databases. RDBs have been the main target of data citation methods since
 211 the surge of the data-centric research paradigm. The RDA “Working Group

212 on Data Citation: Making Dynamic Data Citable”⁵ [46] has been working in
213 the last years on large, dynamic, and changing datasets. The working group
214 has finished the development of its guidelines and has now moved on into an
215 adoption phase. The datasets considered by the WG are often relational.

216 In one of its most recent sessions [47], the Working Group (WG) on
217 Data Citation reported that there are various implementations of its guide-
218 lines for Data Citation on MySQL/Postgres relational databases. Some of
219 these databases are: DEXHELPP⁶ (Social Security Records); NERC (ARGO
220 Global Array); EODC (Earth Observation Data Centre) [29]; LNEC (River
221 dam monitoring); MDS (Million Song Database) [8]; CBMI⁷ (Center for
222 Biomedical Informatics); VMC (Vermont Monitoring Cooperative); CCA⁸
223 (Climate Change Center Austria); VAMDC (Virtual Atomic and Molecular
224 Data Center) [25, 56].

225 More examples of work on data citation in relational databases are [12,
226 53, 2, 23]. The website <https://fairsharing.org/> keeps a long updated
227 list of curated and scientific databases (many of which are relational or graph-
228 based) following FAIR guidelines. These databases are citable since they are
229 compliant with the most recent guidelines, and they are in the vast majority
230 of cases accessible via dynamically created Webpages. In all these databases
231 is, therefore, possible to implement DCD on top of the existing infrastructures
232 for citing data.

233 Data citation techniques are primarily applied to relational databases
234 because of their diffusion and also because the portions of data that are to
235 be cited are easily identified: the whole database, a relation, a tuple, or
236 even an attribute. Many papers [10, 12, 2] consider more complex citable
237 units, recognizing that often the *views* of a database are the ones to be cited.
238 Generally, a *view* is a query on the database. To this end, [53] suggested
239 decomposing the database in a set of views, where each view is associated
240 with its citation.

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

- 242 1. A database cited as a whole, even though only parts of the databases
243 are used in the papers or datasets. Alternatively, the so-called “data pa-

⁵<https://www.rd-alliance.org/groups/data-citation-wg.html>

⁶<http://www.dexhelpp.at/>

⁷<https://medicine.missouri.edu/centers-institutes-labs/center-for-biomedical-informatics>

⁸<https://ccca.ac.at/startseite>

- pers” can be cited, being traditional papers that describe a database [16].
 In this case, all the credit from the citations goes to the database administrators or to the authors of the data papers.
2. Subsets of data, obtained by issuing queries to a database, are individually cited. This is the solution adopted by the *Resource Data Alliance* (RDA) working group on Data Citation [46]. In this case, the credit generated from citations can be distributed among the contributors of the portions of data being cited, and/or to the database administrators.
 3. The database is accessible via a series of Webpages that arrange the content of the database by topic or theme. Examples in the life science domain include the Reactome Pathway database [35], the GtoPdb [31], and the VAMDC [56]. Every single Webpage is unequivocally identifiable and can be individually cited.

Despite all the research efforts dedicated to the study and promotion of data citation, none of the largest citation-based systems, such as Elsevier Scopus, Web of Science, Microsoft Academia, or Google Scholar, consider scientific datasets as citable objects in academic work. Clarivate Analytics Data Citation Index (DCI) [27] is an exception, since its infrastructure tracks data usage in scientific domains and provides the technical means to connect datasets and repositories to scientific papers. However, DCI considers only citations to (previously registered and approved) databases as a whole and does not count citations to database portions such as views, tables, or tuples.

2.2. Data Credit

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

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.

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

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

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

Fang [26] presents a framework to distribute the credit generated by a paper to its authors and to the papers in its reference list in a transitive way. Let us consider the *citation graph* as the graph where the nodes are papers and the links are the citations among them. In this graph, every paper is a source of credit, which is then transferred to the neighboring nodes. The quantity of credit received by each cited paper depends on its impact/role in the citing paper. So far, this theoretical framework is limited to papers, but it can be easily extended to a citation graph including both papers and data.

Zeng et al. [54] proposes the first method to compute credit within a network of papers citing data. Adopting a network flow algorithm, they simulate a random walker to estimate a score for each dataset, leveraging real-world usage data to compute the credit. This is the first step towards an automatic credit computation procedure. This proposal is, however, limited to assigning credit to whole datasets, and it does not deal with the granularity of data. It does not work to assign credit to a single research entity within a dataset.

317 Differently from Zeng et al. [54], we do not treat the credit computation
 318 process, but we focus on the distribution process.

319 2.3. Data Provenance

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

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

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

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

342 The definition of why-provenance is based on the notion of *witness set*.
 343 A witness is a set of relevant tuples that guarantees the existence of t in
 344 $Q(D)$. The lineage is therefore an example of a witness. The why-provenance
 345 of a tuple t is a peculiar set of witnesses – described in [13] – that are
 346 computed from the query, called *witness basis*. A witness basis may be
 347 composed of more than one witness. Therefore, the why-provenance contains
 348 more information than the lineage, since it describes *alternative* ways in
 349 which the same output may be generated.

350 The how-provenance takes the form of a polynomial, called *provenance*
 351 *polynomial*, where the variables are taken from the set of identifiers of the
 352 tuples (provided that each tuple in I has an identifier) and the coefficients are
 353 taken from \mathbb{N} . This provenance also contains information on *how* the input

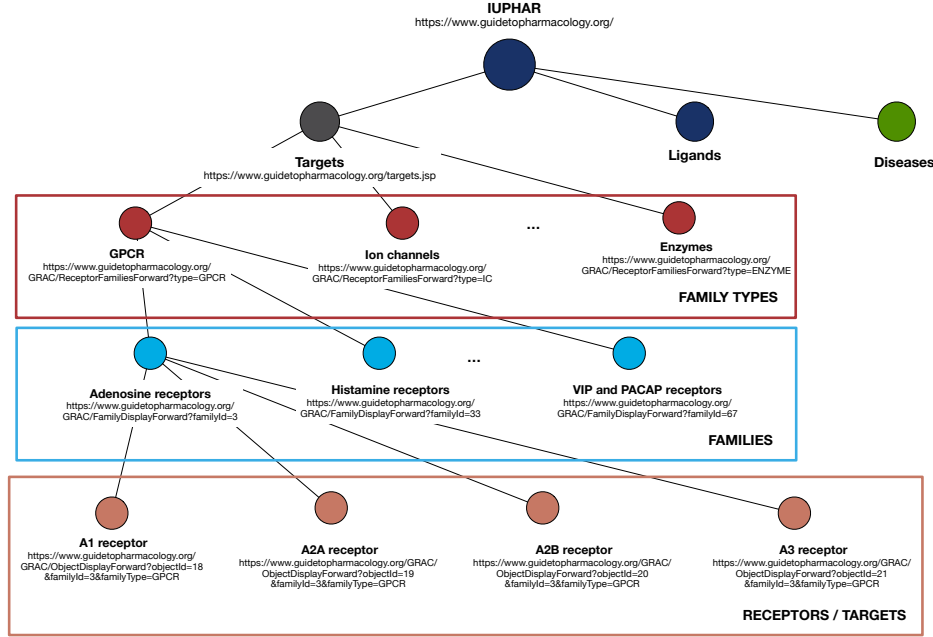


Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

354 tuples are used. For example, when two tuples are combined by a join, they
 355 are also combined in the polynomial by the \cdot operator. When two or more
 356 tuples become equivalent due to a union or a projection, the corresponding
 357 monomials are combined by the $+$ operator.

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

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

363 3. Use Case: GtoPdb

364 As use case we refer to the IUPHAR/BPS Guide to Pharmacology [31]
 365 or GtoPdb⁹. GtoPdb is a well-known and well structured scientific relational
 366 database that contains expertly curated information about diseases, drugs

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

367 in clinical use, their cellular targets, and the mechanisms of action on the
368 human body. It is curated and maintained by the GtoPdb Committee, and
369 by 96 subcommittees, comprising 512 scientists collaborating with in-house
370 curators who draw the information contained in the database from high-
371 quality pharmacological and medicinal chemistry literature. Roughly 1000
372 researchers from all over the world have contributed to the database, and the
373 curators wanted to give recognition to these contributors. This led to some
374 early work on data citation [10].

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

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

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

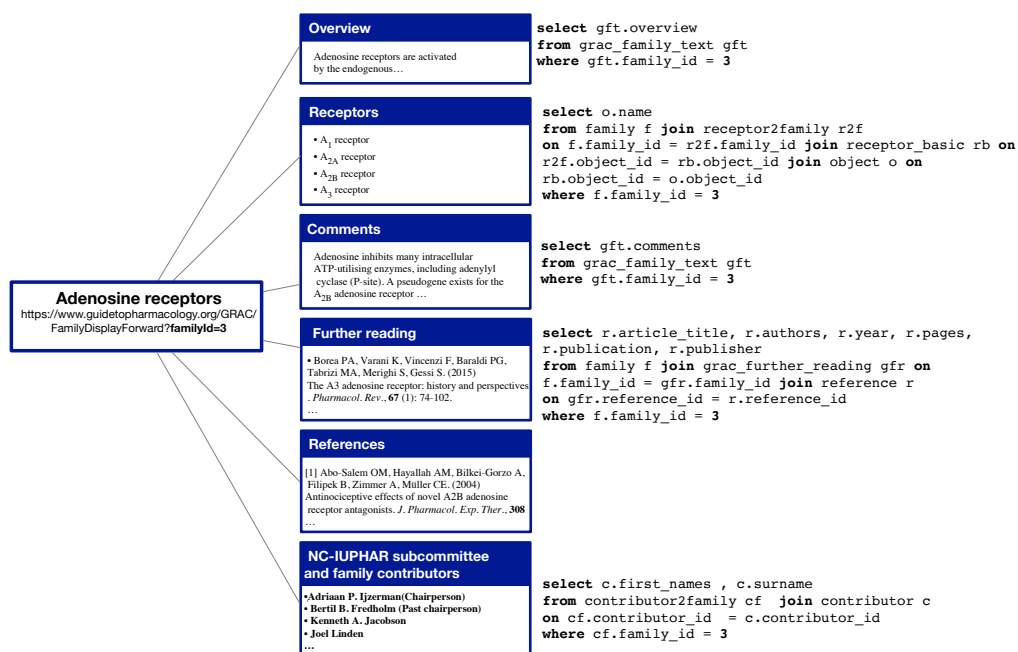


Figure 3: Basic web-page structure of “Adenosine receptors” family (ID 3), with queries used to retrieve the information contained in every section, except references.

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

412 For our running example, consider Table 1. This simplified version of
 413 GtoPdb illustrates three tables: **family**, **contributor** and **contributor2family**.
 414 The first table, **family**, has tuples representing families with three attributes:
 415 the id of the family, its name, and type. Table **contributor** consists of peo-
 416 ple who have helped generate the data of the database. The third table,
 417 **contributor2family**, serves as a link between the families and the people
 418 who contributed to them. For instance, “John Smith” (c_1) contributed to
 419 “Dopamine Receptors” (f_1) as well as to the “YANK Family” (f_4). We use
 420 this example throughout the rest of the paper. In particular, we are using
 421 the **id** attribute of the tables as *provenance token* of its corresponding tu-
 422 ples, that is, as a symbol that serves to identify a tuple when talking about
 423 provenance.

424 4. Data Provenances

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

427 4.1. Lineage

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

```
434 Q1: SELECT DISTINCT f.name
435 FROM family AS f JOIN contributor2family AS c2f
436 ON f.id = c2f.family_id
```

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.

```

437 JOIN contributor AS c ON c2f.contributor_id = c.id
438 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.

439 Table 2 shows the query result, which consists of two tuples. We add
440 an extra attribute **id** so that we can easily refer to each result tuple. The
441 lineage for tuple o_1 is the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$, since the tuple f_1 was
442 joined with $c2f_1$ and then with c_1 , and was also joined with $c2f_2$ and c_2 . No
443 other tuple is used in the database to produce o_1 . For tuple o_2 the lineage is
444 $\{f_4, c2f_4, c_1\}$. Lineage is defined for each tuple of the output, and can differ
445 between tuples.

4.2. Why-Provenance

446 Why-Provenance was first defined in terms of a deterministic semistruc-
447 tured data model and query language [13]. While why-provenance can be
448

defined in many ways, we refer to [17], where it is expressed in terms of the relational model using the relational algebra.

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

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

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

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

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

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

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

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

4.3. How-Provenance

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

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

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

Provenance polynomials may also have monomials whose exponents and/or coefficients are greater than one, for example, $3f_1 \cdot c2f_1 \cdot c_1 + f_1 \cdot c2f_2^3 \cdot c_2^3$. This is a polynomial of a tuple produced by a query where the result of the join between the tuples f_1 , $c2f_1$, and c_1 is produced three times and then merged (e.g. as the result of a union), and the tuples $c2f_2$ and c_2 are used

three times in the operation described by the second monomial (e.g., with nested queries).

5. Credit Distribution and Distribution Strategies

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

5.1. Data Credit and Data Credit Distribution

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

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

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

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

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

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

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

549 5.2. A Lineage-based Distribution Strategy

550 In the lineage-based distribution strategy, each tuple in the output of
 551 a query distributes credit equally to each input tuple that appears in its
 552 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:

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

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

557 As an example, consider the output tuples of Table 2, and assume that
 558 each output tuple has credit $k_o = 1$. The lineage of the first tuple, o_1 , is
 559 the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$. Therefore, each tuple in this set receives credit
 560 $1/5$. The other tuples of the database receive zero credit. The lineage of the
 561 second output tuple is $\{f_4, c2f_4, c_1\}$, therefore each of these tuples receives
 562 credit $1/3$.

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

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

572 $\{f_1, c2f_2, c_2\}$, the existence of o_1 would still be guaranteed. In other words,
 573 while f_1 is always needed for o_1 to appear in the output, only one of the sets
 574 of tuples $\{c2f_1, c_1\}$ and $\{c2f_2, c_2\}$ is required. One could therefore argue that
 575 it would be more fair for f_1 to receive more credit than the other four tuples,
 576 given its role in producing o_1 .

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

582 5.3. A Why-Provenance-Based Distribution Strategy

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

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

589 *Let I be a database instance, Q a query over I , $o \in Q(I)$ an output tuple
 590 and k_o the total credit associated to o . Let $\mathcal{W} = \text{Why}(Q, I, o)$ be the witness
 591 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:

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

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

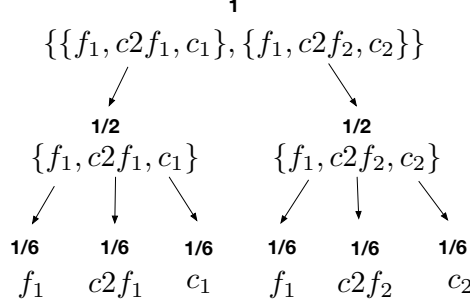


Figure 4: Distribution of credit using why-provenance-based DS for tuple o_1 .

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

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

5.4. A How-Provenance Based Distribution Strategy

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

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

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

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

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

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

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

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

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

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

```

643 Q2: SELECT DISTINCT F.name
644 FROM family as F JOIN
645 (SELECT DISTINCT f.name AS name
646 FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
647 JOIN contributor AS c ON c2f.contributor_id = c.id

```


id	name
oxs_1	Dopamine Receptors

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

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

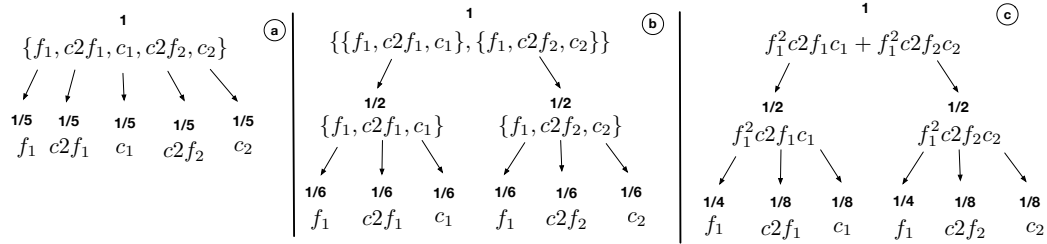


Figure 6: Comparison of different distributions strategies for tuple o_1 produced by query Q2.

```

648 WHERE c.country = "UK") AS R ON F.name = R.name
649 WHERE F.type = "gpcr"

```

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

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

667 6. Experimental Evaluation

668 To understand the trade-offs between these Distribution Strategies (DSs),
669 we perform four sets of experiments using queries over target families pre-
670 sented on the GtoPdb website. The first set of experiments use real queries
671 extracted from citations to GtoPdb published in the British Journal of Phar-
672 macology. The second set uses synthetically produced provenance polynomi-
673 als, corresponding to more complex queries, in order to highlight the differ-
674 ences between the DSs. The third set of experiments considers the accrual of
675 credit over time by the three strategies, again using synthetic queries. The
676 fourth set of experiments shows how the DSs compare to traditional citations
677 in giving credit to data curators using both real and synthetic queries. We
678 close by discussing relative execution times of the three strategies.

679 6.1. Real-world queries

680 Examples of real queries are drawn from papers published in the British
681 Journal of Pharmacology (BJP).¹¹ Each time a paper in this journal cites a
682 webpage from GtoPdb, it reports the URL of the page. From this URL, the
683 query used to obtain the webpage data can be determined. We considered all
684 889 papers in BJCP citing the IUPHAR/BPS Guide to pharmacology [31]
685 as of October 2020, and extracted all webpage URLs to GtoPdb contained
686 within the paper¹². * **Revised, check that it is correct.** *

687 There are eight target family types presented on the GtoPdb website:
688 *GPCR*, *Ion channels*, *NHRs*, *Kinases*, *Catalytic receptors*, *Transporters*, *En-*
689 *zymes* and *Other protein targets*.

690 The queries that we inferred are those used to build target family web-
691 pages. An example was given in Figure 3, where we show how the structure of
692 the “Adenosine receptors” family can be mapped into queries over the under-
693 lying database. In GtoPdb, all target family pages share a similar structure;
694 the only difference is that individual sections, such as “contributors” or “fur-
695 ther readings”, may be absent. Therefore, the same queries can be used to
696 build all of the target family pages by simply changing the family id used in
697 the query (in Figure 3, it is 3). Note that the queries are fairly simple SQL

¹¹<https://bpspubs.onlinelibrary.wiley.com>

¹²The 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.

698 queries, and fall into a class called “select-project-join” or “SPJ” queries. A
 699 total of more than 12K different queries were built in this way.¹³ Without
 700 loss of generality, we give each tuple in the output of a query a credit of 1.

701 *Results.* Figure 7 shows the heat-maps obtained by the distribution of credit
 702 according to the three different DS on one of the tables in the underlying
 703 database, **family**, which is often joined with other tables in the database to
 704 build the webpages. It can be seen that the result of credit distribution over
 705 **family** is the same for all three strategies. The same result is also obtained

¹³For reproducibility purposes, the code we used for our experiments and all queries are available here: https://bitbucket.org/dennis_dosso/credit_distribution_project.

706 with the other tables of the database used by the queries shown in Figure 3.

707 The reason why credit distribution is the same for all three strategies is
 708 that the queries are all simple SPJ queries, which use each table only once and
 709 do joins on key attributes. Under these conditions, each tuple of the output
 710 presents: (i) a how-provenance that is a single monomial with coefficient 1
 711 and exponent 1 in each variable; (ii) a why-provenance with only one witness;
 712 and (iii) a lineage that coincides with the witness in the basis. Hence, for
 713 these queries, the three DSs behave in the same way: Credit is uniformly
 714 distributed among the tuples present in each provenance.

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

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

721 Q3 returned 10 tuples from the version of GtoPdb used. The first tu-
 722 ple, <Bertil B., Fredholm>, has $c_{939} \cdot c_{2f_{496}}$ as its provenance polynomial.
 723 c_{939} represents the provenance token of a tuple in `contributor`, and $c_{2f_{496}}$
 724 the provenance token of a tuple in table `contributor2family`. The why-
 725 provenance of this tuple is $\{\{c_{939}, c_{2f_{496}}\}\}$ and its lineage is $\{c_{939}, c_{2f_{496}}\}$.
 726 Therefore, the credit assigned to these tuples is 1/2 using all three DS. This
 727 happens for all the tuples in the output of each query of GtoPdb, thus making
 728 the distributions equivalent over all outputs.

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

733 6.2. Synthetic queries

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

741 As an example of how the distribution strategies behave with these syn-
 742 thetic queries, consider tuple f_5 in Figure 9. This tuple receives the highest

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:

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

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:

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

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:

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

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

743 quantity of credit using lineage-based distribution, and less credit using why-
 744 and how-provenance because more information is available about the role of
 745 the tuple in the overall computation. Generally speaking, the more complex
 746 the distribution (the most complex being how-provenance), the more credit
 747 is given to tuples which are more frequently used or which have a higher
 748 impact in producing the output tuple.

749 Although synthetic, these provenance polynomials represent realistic queries.
 750 The polynomials can be obtained by any nested query with join and union
 751 operations that use the same tuple multiple times (in which case the expo-
 752 nents are bigger than 1), and the same combination of operations more than
 753 once (in which case the coefficients of monomials are bigger than 1).

754 *Results.* The results of credit distribution on the **family** table using 10K
 755 randomly generated synthetic provenance polynomials are shown in Figure
 756 9. We set the maximum value in the heat maps to the highest value reached
 757 by a tuple in all three distributions (i.e., 8.33). As can be seen, the three
 758 strategies generate significantly different credit distributions indicated by the
 759 varying hues.

760 Lineage-based DS gives the least credit to tuples in the **family** table,
 761 indicated by an overall lighter hue. This is because the DS equally distributes
 762 credit to all tuples appearing in the lineage. Since these queries also use two
 763 other tables, credit is distributed to tuples in those tables.

764 Moving to why-provenance-based DS, we see that more credit is given to
 765 tuples in the **family** table than either of the other two strategies, indicated

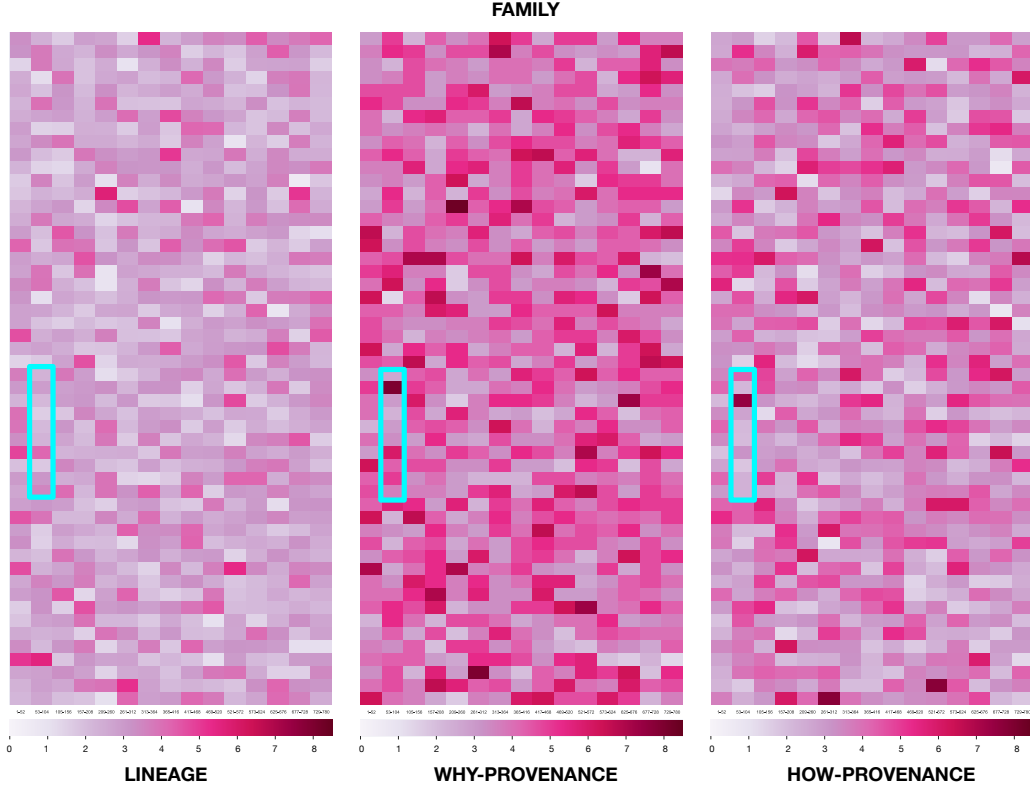


Figure 9: Comparison of three DS on the same table `family` after the distribution computed using 10K synthetic and randomly generated provenance polynomials. The tuples in the blue rectangles are used as example in the discussion connected to Figure 10.

by the overall darker hue. This is because the DS considers the different ways that a tuple is used, e.g. in joins with other tuples. If the same tuple is present in more than one witness, it will draw more credit and take it from other tuples in the witness basis. In this 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.

* **Dennis, rewrote, please check for accuracy.** * 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. However, the distribution does not reward all tuples in the same way, since it uses all information contained in the provenance polynomials.

Referring to the ten tuples within each large blue rectangle (each small rectangle within the large blue rectangle is a tuple), we will number them from 1 (top) to ten (bottom). Note that tuples 2, 7, 8, and 9 are awarded less credit than in either the lineage- or why-provenance strategies, whereas tuples 1 and 3 receives more credit than either of the other two strategies. This shows in more detail that the DS based on how-provenance is more nuanced in terms of credit distribution than the other strategies.

6.3. Credit accrual over time

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

*** Dennis: What is the “rank” of tuple about, and why do we need to bring it up? I’ve taken it out, see if this is still correct. ***

In this figure, four groups of heat-maps are shown. Each group represents a “snapshot” taken after 1K, 2K, 5K and 10K queries have been executed. The ten tuples in each heat-map are from the `family` table, and are the ones highlighted previously in the blue boxes of Figure 9. The queries used are the same as the experiment reported in the previous section. The range of credit in each map goes from 0 (no credit) to 6 (the maximum quantity of credit reached on a tuple at the “snapshot” with 10K queries).

Focusing on the 1K and 2K groups, we see that credit distribution by the three DS are very similar. Still, there are small differences, in particular in tuple 5. *** Dennis: need to say more about tuple 5. Why not also 7 and 10? The interesting one to me is tuple 9, whose credit appears to decrease. ***

The first interesting differences come after 5K queries. In particular, note how tuple 7 receives very little credit in the lineage-based DS, more in the why-provenance-based DS, and the most credit in the how-provenance-based DS. This is because tuple 7 appears in a relatively few lineages, but its role is critical to these queries; thus, why- and how-provenance-based strategies reward it more. On the other hand, tuple 5 is highly rewarded by the lineage-based and why-provenance-based DSs, and less by how-provenance. Although tuple 5 appears in many queries and is used in different combinations, its exponents in the provenance polynomials is low, therefore giving it less credit with how-provenance than with why-provenance. *** Dennis: What does “tuple 7 appears in a relatively few lineages” mean? Do**



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.

814 you mean it appears in relatively few results, but plays a crucial
815 role? *

816 * Dennis: I don't understand this paragraph at all. I am stop-
817 ping here in this subsection until you clean this up a bit, it's very
818 confusing. * It is also interesting to note how other tuples like tuple
819 2 now surpass certain tuples, like tuple 1 that up to 2K queries presented
820 the highest values of credit. This shows how credit can keep track of the
821 "hotspots" in a database over time. The presence of new queries and new
822 credit distributions can change the hotspots in a table, showing how the
823 research community's interests may change during time.

824 Finally, the largest differences are shown in the 10K group. In this case,

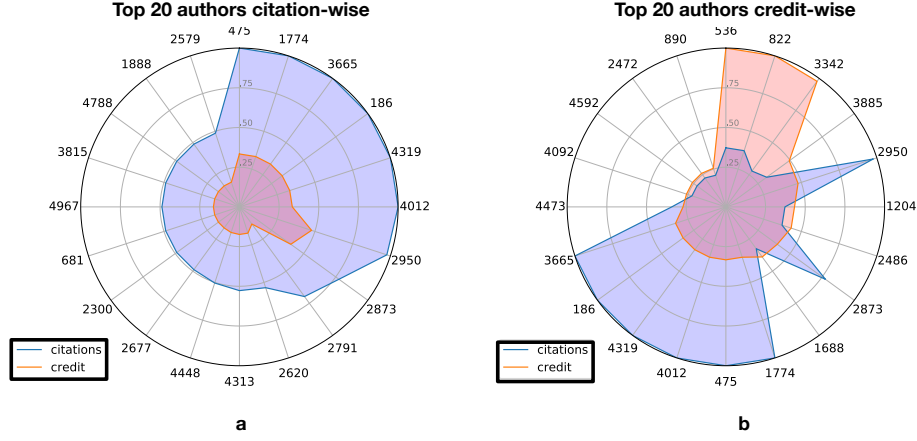


Figure 11: Radars presenting the top 20 authors citation-wise and credit wise, together with their (normalized between 0 and 1) values of citations and credit.

we see a situation similar to the one with 5K queries. Like 8 or 10, specific tuples receive more credit with why-provenance and how-provenance, rather than with lineage. This is still due to the critical role of the tuple in the queries where it appears.

From this progression, given the particular synthetic provenance polynomials that were presented, we can see the differences between the three distributions. These differences become more evident with time, i.e., the more credit is distributed to the tuples.

The DS based on lineage is sufficient when a user only wants to highlight the tuples of the database used by a query (and not only visualized in the output). However, it equally distributes the credit to the tuples of the lineage, therefore not considering the information on the tuples' role in the production of the output.

For this reason, a user may want (depending on the nature of the queries) to use DS based on why-provenance and how-provenance. Using the why-provenance and how-provenance DS, it is possible to change the distribution of credit to the tuple, rewarding more the tuples that have a more critical role in generating the output. Therefore, these two DS can be preferred when the user aims to find “hotspots” in the database based on the tuples' role.

6.4. Credit vs Citations

*** What is the set of authors, just ones on papers in BPC or are**

846 **they the data curators? Check if what I say below is correct! ***

847 In the last set of experiments, we compare credit generated by traditional
848 citations and the proposed credit distribution strategies to see the difference
849 in reward for authors, including data curators. The set of authors used
850 in the experiments are * ?? * those appearing on the paper traditionally
851 cited for GtoPdb, as well as data creators and curators in the `contributor`
852 table in GtoPdb responsible for producing the data used in the queries. An
853 author receives credit via a traditional citation when the traditional GtoPdb
854 paper is cited; they receive credit via a DS when data which they created or
855 curated receives credit. We perform the comparison using both real-world
856 and synthetic queries.

857 *Results: Real-world queries.* As described in Section 6.1, the real-world queries
858 are taken from papers published in the BJCP which reference webpages in
859 GtoPdb. Since for these queries there is no difference in the distribution of
860 credit between the three DS, only one value for credit is given.

861 The results are shown in the radar plots of Figure 11, in which each
862 number on the outer circle (e.g. 475, 1774 and 3665) represents an author (id)
863 and the blue (pink) line represents the normalized value of credit generated
864 by citations (credit), respectively. The first radar plot, Figure 11.a, orders
865 the top 20 authors based on citations in a clockwise direction, whereas Figure
866 11.b, orders the authors with the highest value of credit who do not also have
867 the highest number of citations. *** what does that mean? and it seems**
868 **that there is not a large overlap in the authors in fig. a and b ***

869 The second radar plot, Figure 11.b, orders authors based on the quantity
870 of credit. *** Shoudl discuss that this is not necessarily the same set**
871 **of authors? *** As can be seen, the quantity of credit and the number of
872 citations differ significantly: An author with a high number of citations may
873 have a low credit value (e.g. 475). Similarly, an author with a high credit
874 value may have a low number of citations. *** I didn't see any to point to in**
875 **the figure... and I didn't understand the next few sentences, please**
876 **reword. *** This means that there are citations that are more “valuable” for
877 an author regarding credit. This is because the quantity of credit assigned
878 by these citations is very high, i.e., the impact of those cited data is high.
879 Authors that are cited less than others can have, nonetheless, a high impact
880 on the research community and thus receive a higher quantity of credit.

881 *Results: Synthetic queries.* We produced 100, 1K, and 10K synthetic poly-
882 nomials and distributed credit through them to data. *** I don't understand**

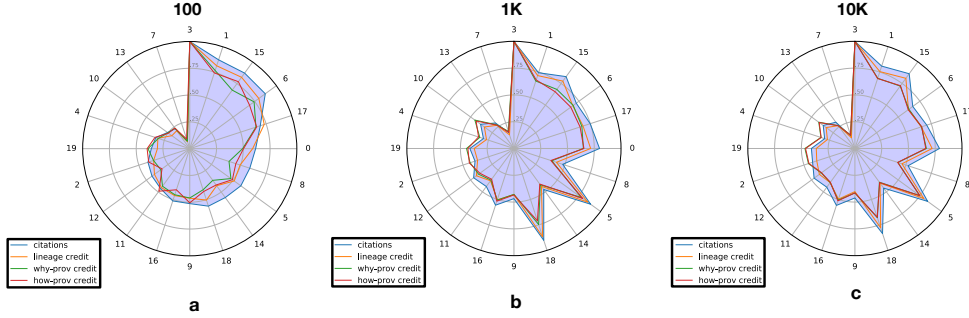


Figure 12: Radars presenting the 20 synthetic authors with corresponding citation and quantities of credit distributed through the 3 DS (all values normalized between 0 and 1) through different numbers of polynomials (respectively, 100, 1K and 10K). The order is the descending one of the citations of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

the rest of this paragraph. ARe you saying you can't figure out the creators/curators of the data? But you were able to do that in earlier experiments, so why not now? I'm missing something in the setup. * Since these polynomials correspond to queries whose authors are not easily identifiable, we created 20 "synthetic" authors, and we randomly assigned one author to each tuple in the database. The authors receive "blocks" of consecutive tuples, with each block of the size varying between 10 and 40 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 query, we assigned one citation to that author. He also receives three kinds of credit, the ones assigned to his tuples through the three different DSs.

Figure 12 reports the three radar plots that are a consequence of the distribution of credit and citations performed and described above with the different quantities of polynomials. Figure 12.a reports the radar plot obtained with 100 polynomials, showing the normalized values of the citation and types of credit assigned to each author. As we see, given the synthetic nature of these queries, the correlation between the number of citations and the quantity of credit assigned to the authors appears to be a much stronger with respect to the case with the real-world queries (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 $1e-11$). Nonetheless, it is still possible to observe how credit does not always exactly follow the citations. The credit distributed via lineage is the one that follows closer the number of citations (a linear correlation of 0.98, p value of $6.15e-16$), while the other types of

907 credit behave slightly differently (a linear correlation of around 0.95 in both
908 cases).

909 Similar observations can be made for Figure 12.b and 12.c, where we kept
910 the order of authors as found in Figure 12.a.

911 What appears from these figures is that, in certain cases, authors that do
912 not have the highest values of citations receive more credit than others, as
913 for example author 11 in Figure 12.a, or author 19 in Figures 12.b and 12.c,
914 with credit distributed with how-provenance-based DS.

915 This once again shows how credit allows us to gain a different perspective
916 on the role of data and authors by going beyond the limitations of traditional
917 citations.

918 It is worth pointing out that, when scaling up to $1K$ and $10K$ polyno-
919 mials, the distributions performed via why-provenance and how-provenance
920 become almost equivalent. We can note that, although not exactly over-
921 lapping, the values of credit assigned to the authors by those DS become
922 quite similar with these higher quantities of polynomials, suggesting a sort
923 of equivalence between the two DSs in this case, at least in the task of re-
924 warding authors (the linear correlation for the values of Figure 12.c is more
925 than 0.99 with a p-value of $1.32e-32$).

926 Since in these experiments we assumed that each output tuple carries
927 credit 1, the queries that return outputs with more tuples also generate more
928 credit. In Figures 11 and 12 the authors that curated bigger bulks of data also
929 receive higher quantities of credit. In more complex and sophisticated scenar-
930 ios, where different strategies may be implemented to decide the generated
931 quantity of credit to be distributed, new factors beyond the only “quantity”
932 of curated data can be factored in in rewarding data curators. The result
933 will be a distribution of credit that represents even better the actual work
934 and worth of data curators.

935 6.5. Execution time

936 *** Where is this table? Table 5 does not show times, and I**
937 **can’t find one that does.** * Table 5 shows the time required to calculate
938 the credit distribution for the three strategies. As can be seen, the execution
939 time grows linearly with the number of polynomials that are submitted to the
940 system. With a high number of polynomials (1M), the time required by the
941 DS based on lineage and why-provenance is lower than the time needed for
942 the DS based on how-provenance. This is due to the more significant number
943 of operations required to calculate the how-provenance DS and distribute the

# of polynomials	lineage	why-prov.	how-prov.
100	226.6 ms	192.0 ms	185.5 ms
200	431.2 ms	392.2 ms	403.2 ms
500	1.013 s	934.2 ms	881.8 ms
1K	2.041 s	1.934 s	1.744 s
2K	3.773 s	3.491 s	3.510 s
5K	8.992 s	8.653 s	8.889 s
10K	17.10 s	16.84 s	16.84 s
20K	34.59 s	35.30 s	39.70 s
100K	3.289 min	3.442 min	3.652 min
1M	35.91 min	34.87 min	37.91 min

Table 6: The times required to perform the three DS for different number of synthetic polynomials.

portions of credit to be assigned to the different tuples. We note that, since we created these polynomials on-the-fly, these values do not include the time required to compute the provenances. Therefore, limited to the time required to distribute credit, the three DS are equivalent in terms of performances. The first differences can be seen only with high number of polynomials, when lineage and why-provenance may be preferred if there are no requirements to assign credit with the strategy implemented by the how-provenance-based DS.

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

7. Conclusions

This paper expanded on our previous work on data credit and data credit distribution in [24] by defining two new distribution strategies, based on why- and how-provenance. The first distribution is based on the concept of witness, and it can give more credit to tuples that appear in more than one witness. In other words, tuples that are more important to the query and are used in different ways are also rewarded more by the strategy. The second DS, based on how-provenance, considers the frequency in which a tuple or a combination of tuples is used in the query through the information contained in the provenance polynomial. In this case, the distribution is even more

965 sensitive than the first one to the role and importance of tuples.

966 To show the differences between the three DS (also considering the one
967 based on lineage, defined in our previous work), we performed different ex-
968 periments on GtoPdb, a curated scientific relational database, with the use
969 of both real and synthetic queries. In the first set of experiments, we used
970 SPJ queries extracted by data citations present in papers published in the
971 British Journal of Pharmacology. Employing these queries, we were able to
972 distribute the credit to the tuples in different tables of the database, high-
973 lighting the tuples used more than others. We showed that with these queries,
974 the three strategies produce the same distribution. These are SPJ queries
975 that do not present self-joins, and therefore the formulas at the base of the
976 DS have the same output.

977 In the second set of experiments, we synthetically produced more complex
978 provenance polynomials, corresponding to more complex synthetic queries,
979 that present exponents and coefficients different than 1. In this way, we
980 showed that, even though all three DS can highlight all the tuples used by
981 the queries in the database, the three have different behaviors. While the DS
982 based on lineage rewards all the tuples used by a query in equal measure, the
983 strategy based on why-provenance tends to reward the tuples more critical
984 to the query. In particular, why-provenance can consider the different ways
985 in which one tuple is used in a query. How-provenance is even more sensitive
986 to the tuples' role: it can also consider the frequency by which a tuple or a
987 set of tuples is used in the case of more complex queries. Depending on the
988 goal of a user, one provenance may be preferred to another.

989 We also showed how the differences between the DS become more and
990 more evident with the passing of time, i.e. when more and more polynomials
991 are processed by the system.

992 In the third set of experiments we compared the citations to the authors
993 to the credit brought to them. We showed how, both in the real-world and
994 synthetic scenarios the credit rewards more the authors that have a higher
995 impact, i.e. the authors connected to the data that produce the highest
996 quantities of credit, and not necessarily the data with the highest citation
997 count. In this sense, credit appears to be an useful new measure to discover
998 data and their corresponding curators that have a high impact in the research
999 world, even when they are cited few times or do not appear at all in the data
1000 that are cited (i.e. the case of data used to build the output of a query but
1001 that is not visualized in the output itself).

1002 In future work, we plan to explore the different potential applications of

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

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

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

References

- [1] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bernstein, P., Boncz, P., Chaudhuri, S., Cheung, A., Doan, A., Dong, L., Franklin, M. J., Freire, J., Halevy, A., Hellerstein, J. M., Idreos, S., Kossmann, D., Kraska, T., Krishnamurthy, S., Markl, V., Melnik, S., Milo, T., Mohan, C., Neumann, T., Chin Ooi, B., Ozcan, F., Patel, J., Pavlo, A., Popa, R., Ramakrishnan, R., Ré, C., Stonebraker, M., and Suciu, D. (2020). The seattle report on database research. *SIGMOD Rec.*, 48(4):44–53.
- [2] Alawini, A., Davidson, S. B., Hu, W., and Wu, Y. (2017). Automating data citation in citedb. *PVLDB*, 10(12):1881–1884.

- 1037 [3] Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., and Wu, Y.
1038 (2018). Data citation: A new provenance challenge. *IEEE Data Eng.*
1039 *Bull.*, 41(1):27–38.
- 1040 [4] Altman, M., Borgman, C. L., Crosas, M., and Martone, M. (2015). An
1041 Introduction to the Joint Principles for Data Citation. *Bulletin of the*
1042 *Association for Information Science and Technology*, 41(3):43–45.
- 1043 [5] Baggerly, K. (2010). Disclose all data in publications. *Nature*,
1044 467(7314):401–401.
- 1045 [6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D.,
1046 Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I.,
1047 Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and
1048 Goble, C. A. (2013). Why linked data is not enough for scientists. *Future*
1049 *Gener. Comput. Syst.*, 29(2):599–611.
- 1050 [7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation
1051 Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.
- 1052 [8] Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The
1053 million song dataset. In *Proceedings of the 12th International Conference*
1054 *on Music Information Retrieval (ISMIR 2011)*, pages 591–596.
- 1055 [9] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In
1056 Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Commu-*
1057 *nication*, pages 93–116. De Gruyter Mouton.
- 1058 [10] Buneman, P. (2006). How to cite curated databases and how to make
1059 them citable. In *18th International Conference on Scientific and Statistical*
1060 *Database Management, SSDBM*, pages 195–203. IEEE Computer Society.
- 1061 [11] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D.,
1062 Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn’t
1063 working, and what to do about it. *Database J. Biol. Databases Curation*,
1064 2020.
- 1065 [12] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation
1066 is a computational problem. *Commun. ACM*, 59(9):50–57.

- 1067 [13] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A
1068 characterization of data provenance. In *Database Theory - ICDT 2001*,
1069 *8th International Conference*, pages 316–330.
- 1070 [14] Buneman, P. and Silvello, G. (2010). A rule-based citation system for
1071 structured and evolving datasets. *IEEE Data Eng. Bull.*, 33(3):33–41.
- 1072 [15] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N.,
1073 Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry,
1074 R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a.,
1075 and Wright, D. (2012). Making Data a First Class Scientific Output:
1076 Data Citation and Publication by NERC’s Environmental Data Centres.
1077 *International Journal of Digital Curation*, 7(1):107–113.
- 1078 [16] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Jour-
1079 nals: A Survey. *Journal of the Association for Information Science and*
1080 *Technology*, 66(9):1747–1762.
- 1081 [17] Cheney, J., Chiticariu, L., and Tan, W. (2009). Provenance in databases:
1082 Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–
1083 474.
- 1084 [18] CODATA-ICSTI Task Group on Data Citation Standards and Practices
1085 (2013). *Out of Cite, Out of Mind: The Current State of Practice, Policy,*
1086 *and Technology for the Citation of Data*, volume 12.
- 1087 [19] Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., and Simons, N.
1088 (2019). Bringing citations and usage metrics together to make data count.
1089 *Data Science Journal*, 18(1).
- 1090 [20] Cronin, B. (1984). *The citation process. The role and significance of*
1091 *citations in scientific communication*. London: Taylor Graham.
- 1092 [21] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evi-
1093 dence of a structural shift in scholarly communication practices? *JASIST*,
1094 52(7):558–569.
- 1095 [22] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of
1096 view data in a warehousing environment. *ACM Trans. Database Syst.*,
1097 25(2):179–227.

- 1098 [23] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model
1099 for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on*
1100 *Innovative Data Systems Research*. www.cidrdb.org.
- 1101 [24] Dosso, D. and Silvello, G. (2020). Data credit distribution: A
1102 new method to estimate databases impact. *Journal of Informetrics*,
1103 14(4):101080.
- 1104 [25] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The vir-
1105 tual atomic and molecular data centre (VAMDC) consortium. *Journal of*
1106 *Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.
- 1107 [26] Fang, H. (2018). A discussion of citations from the perspective of the
1108 contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–
1109 1520.
- 1110 [27] Force, M., Robinson, N., Matthews, M., Auld, D., and Boletta, M.
1111 (2016). Research data in journals and repositories in the web of science:
1112 Developments and recommendations. *Bulletin of IEEE Technical Com-*
1113 *mittee on Digital Libraries, Special Issue on Data Citation*, 12(1):27–30.
- 1114 [28] Garfield, E. (1999). Journal impact factor: a brief review. *Can. Med.*
1115 *Assoc.*, 979-980.
- 1116 [29] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data
1117 identification and process monitoring for reproducible earth observation
1118 research. In *2019 15th International Conference on eScience (eScience)*,
1119 pages 28–38. IEEE.
- 1120 [30] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance
1121 semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-*
1122 *SIGART symposium on Principles of database systems*, pages 31–40. ACM.
- 1123 [31] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson,
1124 A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton,
1125 S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F.,
1126 Spedding, M., Davies, J. A., and Nc-Iuphar (2018). The IUPHAR/BPS
1127 guide to PHARMACOLOGY in 2018: updates and expansion to encom-
1128 pass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Re-*
1129 *search*, 46(Database-Issue):D1091–D1106.

- 1130 [32] Hartley, J. (2017). Authors and their citations: a point of view. *Scien-*
1131 *tometrics*, 110(2):1081–1084.
- 1132 [33] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a
1133 transformed scientific method.
- 1134 [34] Honor, L. B., Haselgrove, C., Frazier, J. A., and Kennedy, D. N. (2016).
1135 Data citation in neuroimaging: proposed best practices for data identifi-
1136 cation and attribution. *Frontiers in neuroinformatics*, 10:34.
- 1137 [35] Joshi-Tope, G., Gillespie, M., Vastrik, I., D’Eustachio, P., Schmidt, E.,
1138 de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis,
1139 S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of bio-
1140 logical pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432.
- 1141 [36] Katz, D. (2014). Transitive credit as a means to address social and
1142 technological concerns stemming from citation and attribution of digital
1143 products. *Journal of Open Research Software*, 2(1).
- 1144 [37] Katz, D. S., Hong, N., Clark, T., Fenner, M., and Martone, M. (2020).
1145 Software and data citation. *Computing in Science & Engineering*, 22 (2):4–
1146 7.
- 1147 [38] Kosten, J. (2016). A classification of the use of research indicators.
1148 *Scientometrics*, 108(1):457–464.
- 1149 [39] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S.
1150 (2011). Citation and Peer Review of Data: Moving Towards Formal Data
1151 Publication. *International Journal of Digital Curation*, 6(2):4–37.
- 1152 [40] Martone, M. (2014). Joint declaration of data citation principles.
1153 *FORCE11. San Diego CA. Data Citation Synthesis Group*. [https://www.](https://www.force11.org/datacitationprinciples)
1154 [force11.org/datacitationprinciples](https://www.force11.org/datacitationprinciples), online September 2020.
- 1155 [41] Meho, L. I. and Yang, K. (2007). Impact of data sources on citation
1156 counts and rankings of LIS faculty: Web of science versus scopus and
1157 google scholar. *Journal of the american society for information science*
1158 *and technology*, 58(13):2105–2125.
- 1159 [42] Milo, T. (2019). Getting rid of data. *Journal of Data and Information*
1160 *Quality (JDIQ)*, 12(1):1–7.

- 1161 [43] Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D.,
1162 Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G.,
1163 Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff,
1164 D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D.,
1165 Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson,
1166 E., McNutt, M., Miguel, M., Paluck, E. L., Simonsohn, U., Soderberg, C.,
1167 Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers,
1168 E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research
1169 culture. *Science*, 348(6242):1422–1425.
- 1170 [44] Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J.
1171 (2016). Research data explored: An extended analysis of citations and
1172 altmetrics. *Scientometrics*, 107(2):723–744.
- 1173 [45] Pröll, S. and Rauber, A. (2013). Scalable data citation in dynamic, large
1174 databases: Model and reference implementation. In *Proceedings of the*
1175 *2013 IEEE International Conference on Big Data*, pages 307–312. IEEE.
- 1176 [46] Rauber, A., Ari, A., van Uytvanck, D., and Pröll, S. (2016). Identifi-
1177 cation of Reproducible Subsets for Data Citation, Sharing and Re-Use.
1178 *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue*
1179 *on Data Citation*, 12(1):6–15.
- 1180 [47] Rauber, A., Asmi, A., van Uytvanck, D., and Proell, S. (2015). Data
1181 citation of evolving data: Recommendations of the working group on data
1182 citation (wgdc). *Result of the RDA Data Citation WG*, 20.
- 1183 [48] Silvello, G. (2018). Theory and practice of data citation. *J. Assoc. Inf.*
1184 *Sci. Technol.*, 69(1):6–20.
- 1185 [49] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data
1186 provenance in e-science. *SIGMOD Record*, 34(3):31–36.
- 1187 [50] Spengler, S. (2012). Data Citation and Attribution: A Funder’s Per-
1188 spective. In of Sciences’ Board on Research Data, N. A. and Information,
1189 editors, *Report from Developing Data Attribution and Citation Practices*
1190 *and Standards: An International Symposium and Workshop*, pages 177–
1191 178. National Academies Press: Washington DC.

- 1192 [51] Ur Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah,
1193 T. V., and Khan, S. U. (2016). Big data reduction methods: a survey.
1194 *Data Science and Engineering*, 1(4):265–284.
- 1195 [52] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G.,
1196 Axton, M., Baak, A., Blomberg, N., Boiten, J., da Silva Santos, L. B.,
1197 Bourne, P. E., et al. (2016). The fair guiding principles for scientific data
1198 management and stewardship. *Scientific data*, 3.
- 1199 [53] Wu, Y., Alawini, A., Davidson, S. B., and Silvello, G. (2018). Data
1200 citation: Giving credit where credit is due. In *Proceedings of the 2018*
1201 *International Conference on Management of Data, SIGMOD*, pages 99–
1202 114.
- 1203 [54] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to
1204 scientific datasets using article citation networks. *Journal of Informetrics*,
1205 14(2).
- 1206 [55] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of
1207 new researchers using the zp-index. *Scientometrics*, 106(3):901–916.
- 1208 [56] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for
1209 Datasets Citation and Extraction Reproducibility in VADMC. *Journal of*
1210 *Molecular Spectroscopy*, 327:122–137.