

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, say, 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 base of systems such as Google Scholar and 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. **Gianmaria: talk about the zp-index in the related**

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

65 **work**

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

75 We study the DCD problem in the context of relational databases (RDBs)
76 for the following reasons:

- 77 • RDBs are pervasive in the scientific world and are the main focus of
78 current data citation methods [12, 45]. Many scientific curated RDBs
79 are accessible via Webpages dynamically generated via queries to the
80 database.
- 81 • RDBs, being well-consolidated technologies, are widely used. The
82 “relational database market alone has revenue upwards of \$50B” [1].
83 Known outside the database community, they are often the test-bed
84 for new methods that can be adapted to other databases, e.g., graphs
85 or document databases.
- 86 • In an RDB, the data portions that can be credited may easily be de-
87 fined. In particular, we consider the following: (i) the whole database,
88 (ii) the tables, and (iii) the tuples.

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

96 We summarize the DCD process in Figure 1:

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

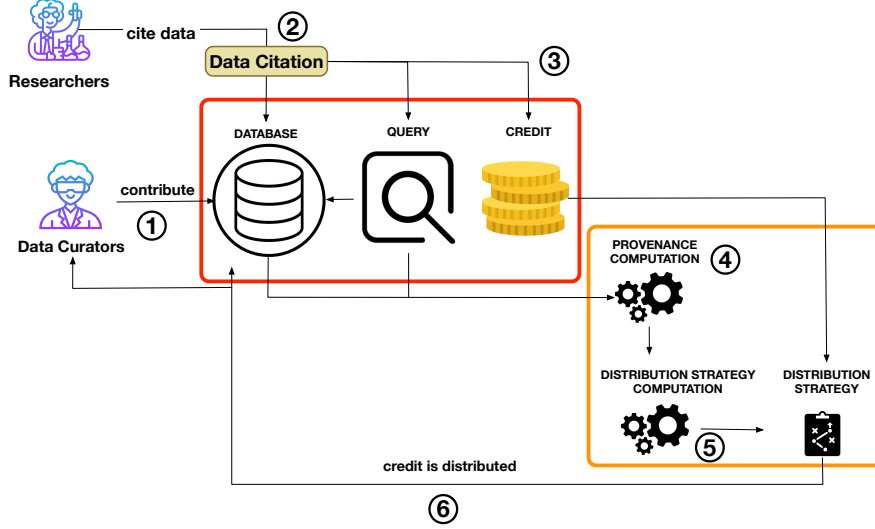


Figure 1: Overview of the credit distribution pipeline.

- 97 **Step 1** Scientists and experts contribute the curated information contained
 98 in a scientific database. These are called the “Data Curators”.
- 99 **Step 2** Other researchers use the data in their research, and when possible,
 100 cite them.
- 101 **Step 3** The citation to the data generates credit, that can be used as a
 102 proxy for the impact of the data in the citing paper. This credit is
 103 represented as a real value $k \in \mathbb{R}_{>0}$.
- 104 **Step 4** Given the database instance I and the query Q , it is possible to
 105 compute the *data provenance* of $Q(I)$. The provenance of $Q(I)$ is a
 106 form of metadata that describes the generation process undertaken by
 107 Q , and the data used in I to generate the output [17]. Many different
 108 notions of provenance have been proposed in the literature for data in
 109 database management systems [22, 13, 30], describing different kinds of
 110 relationships between data in the input and the output of a query. As
 111 reported in [17], these provenances, beyond the intrinsic information
 112 on how queries work, have been used in several applications, as the
 113 study of annotation propagation and view update. In this paper, we
 114 consider three types of provenance: lineage, why-provenance, and how-
 115 provenance.

116 **Step 5** The provenance is the input of the CDC problem, whose aim is the
 117 computation of the *Credit Distribution Strategy* (CDS, also referred
 118 only as Distribution Strategy, DS). The CDS is a function that dis-
 119 tributes k to the data in the input database I , and is defined on the
 120 basis of citation policies decided at the database administration level
 121 or at the domain community level. Certainly, we are in a one-solution-
 122 does-not-fit-all scenario, but CDS can be defined with great variability
 123 and flexibility, thus allowing for ample customization. In this paper,
 124 we describe CDS based on data provenance, and they are therefore
 125 closely related to the kind of provenance computed in step 4. In this
 126 work, we describe three CDS, each based on one of the three considered
 127 provenances.

128 **Step 6** Once the CDS is computed, it is then used to distribute the given
 129 credit k to the parts of the database that are responsible for the gener-
 130 ation of $Q(I)$. Transitively, this credit is also divided and given to the
 131 corresponding authors of those data.

132 In this paper, we expand the recent work presented in [24], where we first
 133 asked how to correctly reward data and data curators that are usually left
 134 without credit from the current citation systems. In that work, we first de-
 135 fined the problem of DCD in relational databases and we proposed a viable
 136 distribution strategy based on *lineage* – i.e., the simplest form of *data prove-*
 137 *nance*. The lineage of a tuple t in the output $Q(I)$ is defined as the set of
 138 all and only the tuples in the database instance I that are “relevant” to the
 139 production of t , that is the tuple that are used by Q in the production of t .
 140 The lineage-based strategy equally redistributes the credit k to the tuples in
 141 the lineage set, thus each tuple receives credit $k/|L_t|$, where L_t is the lineage
 142 set of t .

143 One may argue that this distribution strategy may be too simplistic, since
 144 lineage only tells the relevant tuple used to produce the output and it does
 145 not convey any information about their role or importance in the query.
 146 Therefore, one may instead desire to give more credit to the tuples that are
 147 more relevant or essential to the production of the output, i.e. those tuples
 148 that, if removed, would prevent the output tuple to appear in the final result,
 149 or those tuples used more than once by the query.

150 Therefore, in this paper, we expand the research done in [24] by propos-
 151 ing new Distribution Strategies, based on other forms of data provenance.
 152 Namely, why-provenance [13] and how-provenance [30]. We compare them

also with the lineage-based solution and discuss why one may be preferred to another depending on the application and its goals. In particular, we show that why-provenance and how-provenance are more sensitive to the role of a tuple in a query (depending on how many times the tuple is used or how it is used). The DS based on why-provenance rewards more the tuples that are essential to the production of the result set. Whereas, the DS based on how-provenance also takes into consideration in how many different ways a tuple is used, presenting an higher level of discernment.

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

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

Contributions. of this work include:

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

Outline. The rest of the paper is organized as follows: Section 2 presents the background and related work; Section 3 describes the use case we adopted;

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

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

183 Section 4 briefly presents the provenances used in the paper; Section 5 de-
184 scribes the problem of DCD and the proposed DS; in Section 6 we present
185 the experimental evaluation. Finally, Section 7 draws some conclusions and
186 outlines future work.

187 2. Background

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

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

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

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

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

230 2.1. Data Citation in Relational Databases

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

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

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>

245 (Climate Change Center Austria); VAMDC (Virtual Atomic and Molecular
246 Data Center) [25, 56].

247 More examples of work on data citation in relational databases are [12,
248 53, 2, 23]. The website <https://fairsharing.org/> keeps a long updated
249 list of curated and scientific databases (many of which are relational or graph-
250 based) following FAIR guidelines. These databases are citable since they are
251 compliant with the most recent guidelines, and they are in the vast majority
252 of cases accessible via dynamically created Webpages. In all these databases
253 is, therefore, possible to implement DCD on top of the existing infrastructures
254 for citing data.

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

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

- 264 1. A database cited as a whole, even though only parts of the databases
265 are used in the papers or datasets. Alternatively, the so-called “data pa-
266 pers” can be cited, being traditional papers that describe a database [16].
267 In this case, all the credit from the citations goes to the database ad-
268 ministrators or to the authors of the data papers.
- 269 2. Subsets of data, obtained by issuing queries to a database, are individ-
270 ually cited. This is the solution adopted by the *Resource Data Alliance*
271 (RDA) working group on Data Citation [46]. In this case, the credit
272 generated from citations can be distributed among the contributors of
273 the portions of data being cited, and/or to the database administrators.
- 274 3. The database is accessible via a series of Webpages that arrange the
275 content of the database by topic or theme. Examples in the life science
276 domain include the Reactome Pathway database [35], the GtoPdb [31],
277 and the VAMDC [56]. Every single Webpage is unequivocally identifi-
278 able and can be individually cited.

279 Despite all the research efforts dedicated to the study and promotion of
280 data citation, none of the largest citation-based systems, such as Elsevier
281 Scopus, Web of Science, Microsoft Academia, or Google Scholar, consider

282 scientific datasets as citable objects in academic work. Clarivate Analytics
283 Data Citation Index (DCI) [27] is an exception, since its infrastructure tracks
284 data usage in scientific domains and provides the technical means to connect
285 datasets and repositories to scientific papers. However, DCI considers only
286 citations to (previously registered and approved) databases as a whole and
287 does not count citations to database portions such as views, tables, or tuples.

288 2.2. Data Credit

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

293 [36] suggests the need for a *modified citation system* that includes the
294 idea of *transient* and *fractional credit*, to be used by developers of research
295 products as software and data. In the paper two considerations are made:
296 (i) research objects such as data and software are currently not formally
297 rewarded or recognized by the community; (ii) even in traditional papers,
298 the contribution of each author to the work is hard to understand, unless
299 explicitly specified in the paper. This is even more true for data, where
300 different groups of people work on the same database.

301 In [36] credit is defined as a “quantity” that describes the importance of
302 a research entity, such as papers, software, or data, mentioned in a citation.
303 We add that the concept of credit can be built on top of the existing infras-
304 tructure handling traditional and data citations. [36] further explores the
305 idea of a *distribution* of credit from research entities (i.e., papers and data)
306 to other research entities through citations that connect them. Thanks to
307 traditional citations and now also to data citations, this distribution is fi-
308 nally possible, at least between papers and data. Some problems related to
309 traditional citations can thus be solved by citations:

- 310 1. Credit rewards research entities that to date are not (formally) recog-
311 nized (a goal shared with data citation).
- 312 2. Credit can reward authors *proportionally* to their role in generating
313 the entity. The more an author contributes to a paper, the more credit
314 is given to him. [55] work on something similar with their zp-index,
315 which includes in its formulation the position (and thus the role) of a
316 publication author to represent its impact in the work itself.

317 3. Credit can be *transitively* channeled through a chain of papers citing
 318 each other, thus enabling the rewarding of older papers **that are no**
 319 **more cited, since other papers summarize or report their con-**
 320 **tent. Gianmaria: I do not understand this token, what do you**
 321 **mean with: papers that are no more cited?** but are nevertheless
 322 crucial in a research area for the influence of their content.

323 [26] presents a framework to distribute the credit generated by a paper to
 324 its authors and to the papers in its reference list in a transitive way. Let us
 325 consider the *citation graph* as the graph where the nodes are papers and the
 326 links are the citations among them. In this graph, every paper is a source of
 327 credit, which is then transferred to the neighboring nodes. The quantity of
 328 credit received by each cited paper depends on its impact/role in the citing
 329 paper. So far, this theoretical framework is limited to papers, but it can be
 330 easily extended to a citation graph including both papers and data.

331 [54] proposes the first method to compute credit within a network of
 332 papers citing data. Adopting a network flow algorithm, they simulate a
 333 random walker to estimate a score for each dataset, leveraging real-world
 334 usage data to compute the credit. This is the first step towards an automatic
 335 credit computation procedure. This proposal is, however, limited to assigning
 336 credit to whole datasets, and it does not deal with the granularity of data.
 337 It does not work to assign credit to a single research entity within a dataset.
 338 Differently from [54], we do not treat the credit computation process, but we
 339 focus on the distribution process.

340 2.3. Data Provenance

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

351 [17] presents four main types of data citation for DBMS: *lineage* [22],
 352 *why-provenance* [13], *how-provenance* [30] and *where-provenance* [13].

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

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

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

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

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

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

3. Use Case: GtoPdb

As use case we refer to the IUPHAR/BPS Guide to Pharmacology [31] or GtoPdb⁹. GtoPdb is a well-known and well structured scientific relational

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

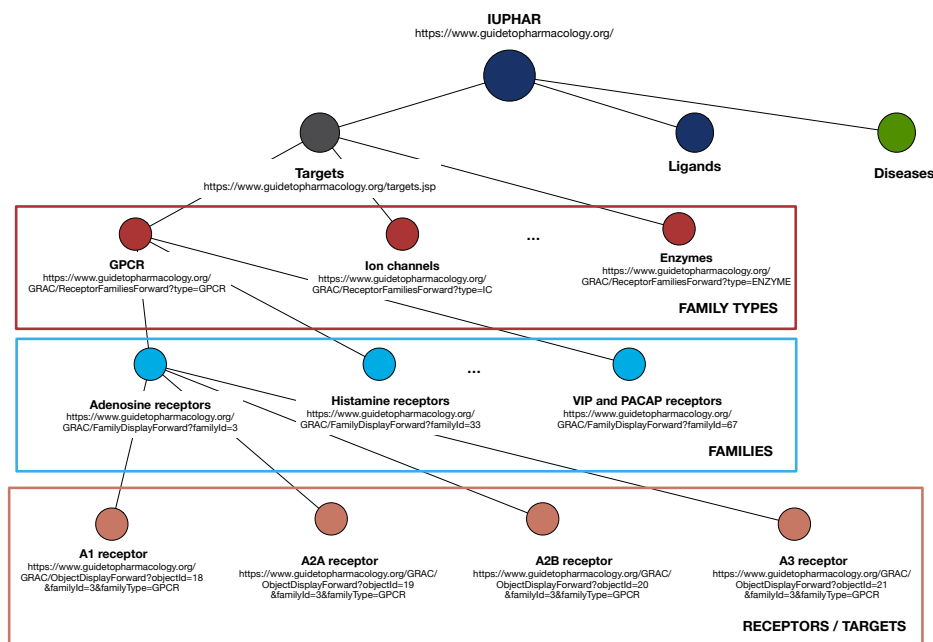


Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

387 database that contains expertly curated information about diseases, drugs in
 388 clinical use, their cellular targets, and the mechanisms of action on the hu-
 389 man body. It is curated and maintained by the GtoPdb Committee, and by
 390 96 subcommittees, comprising 512 scientists collaborating with in-house cu-
 391 rators that draw the information contained in the database from high-quality
 392 pharmacological and medicinal chemistry literature. Circa 1000 researchers
 393 from many parts of the world have contributed to the database. The curators
 394 desire to give recognition to the contributors that led to some early work on
 395 data citation [10].

396 GtoPdb is relational, but its logical structure is hierarchical, as shown
 397 in Figure 2, and the information contained in the database is also organized
 398 into webpages focused on specific diseases, targets or ligands and families
 399 (i.e., groups) of them for easier access by the users. As depicted in Figure
 400 2, the database can be thought of as a tree where the root is the database
 401 itself in its entirety; the first level is composed of the Targets, Ligands, and
 402 Diseases considered in their entirety. In this paper, we focus on target and
 403 target families; thus, in the figure, at the third level, we show some of the

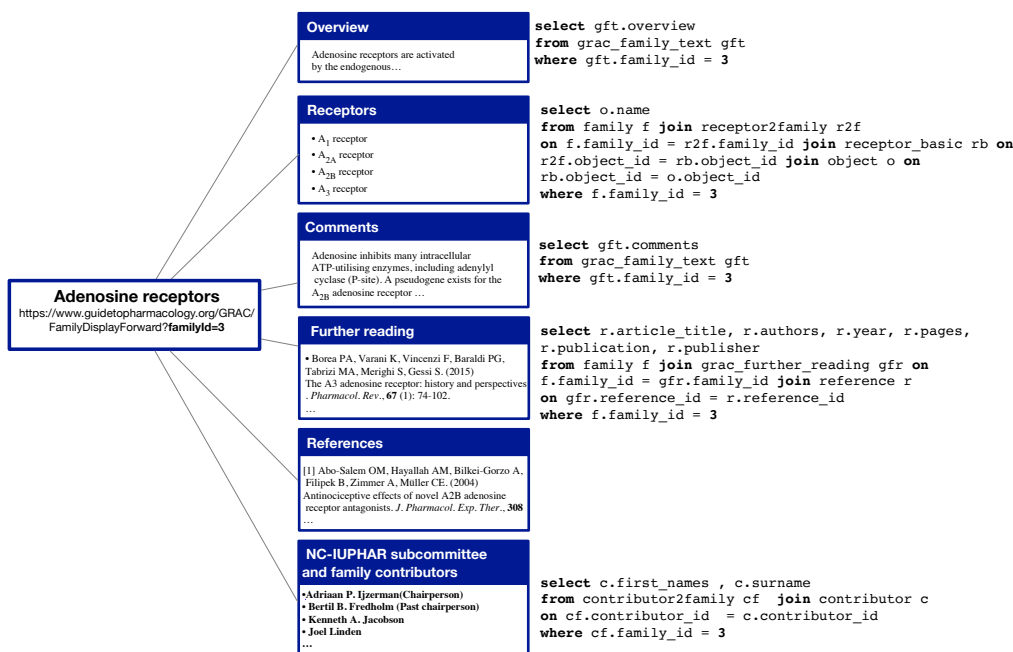


Figure 3: Basic web-page structure of “Adenosine receptors” family (ID 3), with queries used to retrieve the information contained in every section, except references.

family types, that is, groups of targets. At the third level, we show families of targets, a finer level of granularity, and finally, at the last level, the single targets, also known as receptors.

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

421 overview), content, and references.

422 What happens is that many papers in the literature use the GtoPdb’s
423 information without including a reference to the specific page being cited.
424 Instead, they only cite one data paper describing GtoPdb (e.g., the more
425 recent [31]) and refer to targets, ligands, diseases, etc. only by their name.
426 Thus, the citations to the specific families turn out to be *de-facto* “hidden” to
427 the citation systems such as Google Scholar, and useless for the computation
428 of bibliometrics.

429 In specific “lucky” circumstances, as in the case of papers available in PDF
430 and published in the British Journal of Clinical Pharmacology ¹⁰ (BJCP),
431 when a family, a ligand, a receptor name, etc. are used, they also have
432 a hyperlink pointing to the corresponding webpage in GtoPdb. Therefore,
433 the citations to the families can be spotted and counted using the URLs
434 reported in the papers. These citations to GtoPdb webpages, in any case,
435 are not counted as such by the citation systems, so they are not converted
436 into credit for curators and collaborators.

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

439 The first table’s tuples represent four families, composed of three at-
440 tributes: the id of the family, its name, and type. **contributor** contains the
441 people who have helped to generate the data of the database. Finally, table
442 **contributor2family** serves as a link between the families and the people
443 who contributed to them. For instance, “John Smith” (c_1) contributed to
444 “Dopamine Receptors” (f_1) as well as to the “YANK Family” (f_4). We use
445 this example throughout the rest of the paper.

446 4. Data Provenances

447 4.1. Lineage

448 Lineage was first introduced by Cui et al. [22], and given a database
449 instance I a query Q , it associates each tuple $o \in Q(I)$ to a set of tuples
450 in the input. In general, the lineage of one tuple o is a collection of tuples
451 that helped to “produce” it [17]. To give an idea of what can happen with
452 the lineage of tuples, consider the following SQL query **Q1**, applied to the
453 database described in Table 1, that asks for the names of families curated by
454 researchers based in the United Kingdom (UK):

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

455 Q1: SELECT DISTINCT f.name
456 FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
457 JOIN contributor AS c ON c2f.contributor_id = c.id
458 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 a SQL query applied on the database of Table 1, asking the names of the families curated by a researcher based in the UK. The attribute **id** is not part of the output and we added it to easily identify the two tuples. Every tuple is annotated with its lineage, we use the id of the tuples to identify them.

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

461 For tuple o_1 the lineage is the set $\{f_1, c2f_1, c_1, c2f_2, c_2\}$, since the tuple f_1
462 was joined with $c2f_1$ and then with c_1 , but also with $c2f_2$ and c_2 . No other
463 tuple is used in the database to produce o_1 . For tuple o_2 , instead, the lineage
464 is $\{f_4, c2f_4, c_1\}$. Therefore, as we see, the lineage is defined for each tuple of
465 the output, and it can be different for tuples in the same output.

4.2. Why-Provenance

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

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, all the witnesses’ set is finite (if 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 *particular subset* of the set of witnesses, that is, a particular selection of witnesses. This subset is called *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 . Thus, the basis tends to be very small compared to the set of all possible witnesses [17]. Also, the witnesses are 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-provenances 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 property, consider the example in Table 3, where we reported the result of query **Q1** with the tuples annotated with their why-provenance.

Output tuple o_2 presents a why-provenance composed of 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 presents a witness basis made of two witnesses.

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.

These two sets fundamentally represent the 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 , and is represented by the first witness. The second possibility is that f_1 is joined with $c2f_2$ and c_2 . This means that to generate o_1 it is sufficient that only one of the two witnesses is present in the input database.

4.3. How-Provenance

While why-provenance describes the source tuples that witness an output tuple in the result of the query, it leaves out some information. How-provenance was firstly defined in [30], based on the introduction of a *semiring* algebraic structure, and it is a form of provenance that takes the form of a *polynomial*.

The key idea of 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).

For a simple example of how how-provenance works, refer to Table 4, where 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 alternatives. The output tuple is the result of a merge of two distinct tuples after the projection on the attribute **name**. This merge is, in turn, due to the presence of the **DISTINCT** operator in the SQL query. This simple example gives a first basic idea behind how-provenance and how it allows us to track the operations that produced an output tuple.

526 5. Credit Distribution and Distribution Strategies

527 5.1. Data Credit and Data Credit Distribution

528 Given a database instance I , a *recipient of credit* corresponds to a unit
 529 of information within the same database. In the case of relational databases,
 530 recipients may be (i) the whole database itself; (ii) a table; (iii) a tuple; (iv)
 531 an attribute.

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

536 *Data Credit Distributions* (DCD) considers a database instance I , a cer-
 537 tain quantity of credit k (here, without loss of generality, deemed to be
 538 given), and a query Q producing a result set $Q(I)$. DCD consists of defining
 539 a function, i.e., a *distribution strategy* (DS), to split credit into portions to
 540 be assigned to the tuples in I .

541 In the following, we follow the notation given by Cheney et al. [17]: a
 542 *tuple location* is defined as a tuple in one relation of I , tagged with its name.
 543 It is indicated with (R, t) , where R is the relation in the database, and t
 544 is the tuple in R . With reference to the running example, $(\text{family}, \langle f_1,$
 545 $\text{Dopamine Receptors}, \text{gpcr} \rangle)$ is the tuple location of the first tuple in the
 546 **family** relation. The set of all the tuple locations in I is called *TupleLoc*.
 547 The following is the definition of DCD at *tuple level*. We refer to the level of
 548 tuple because the credit is annotated to tuples.

549 **Definition 5.1. Data Credit Distribution at tuple level (DCD) [24]**
 550 *Given a database instance I , a query Q over I and the value $k \in \mathbb{R}_{>0}$, DCD*
 551 *is defined as the computation of the function $f_{I,Q} : \text{TupleLoc} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$*
 552 *such that $f_{I,Q}(t, k) = h$ where $0 \leq h \leq k$ and $\sum_{t \in \text{TupleLoc}} f_{I,Q}(t, k) = k$.*

553 As we see, the DS is a function that annotates each tuple in the database
 554 with a real value, which is a fraction of the given quantity k . The only
 555 constraint is that the sum of the credit fractions annotation the tuples must
 556 be k (i.e. no credit is generated nor destroyed during the distribution). Given
 557 I and Q , many different DS may be defined as long as they sum up to k . We
 558 use the information provided by data provenance to define sensible functions
 559 that take into account the issued query.

560 *5.2. A Lineage-based Distribution Strategy*

561 With the information provided by the lineage, we defined the following
562 DS:

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 the credit associated to o . Let L be the lineage of o and t be a generic tuple in I , then t receives a credit equal to:

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

563 The DS is defined for one tuple of the output. Therefore, to perform credit
564 distribution for a whole set of output tuples, it is necessary to first divide
565 the credit to each tuple in the output and then compute the distribution for
566 each one of them. In this paper, we assume that each output tuple carries
567 credit equal to 1. This lineage-based DS distributes credit only among tuples
568 that have a role, whichever it is, in creating o by the query Q . Each of them
569 receives an equal share of the credit. The more the tuples in a lineage set,
570 the less the credit every single tuple receives.

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

576 At the end of the process, tuples $f_1, c2f_2, c_2$ receive credit $1/5$ each, tuples
577 f_4 and $c2f_4$ receive $1/3$, while tuple c_1 receive $8/15$. Suppose one tuple
578 appears in more than one lineage set. In that case, it will accumulate credit
579 from the distribution associated with each one of these sets, implying its
580 more significant relevance in the context of the considered query, as is the
581 case with c_1 in this example.

582 Not all of the tuples of the lineage of o_1 are necessary at the same time
583 for the generation of the tuple. If the database only had the set of tuples
584 $\{f_1, c2f_1, c_1\}$ or the set $\{f_1, c2f_2, c_2\}$, its existence would still be guaranteed.
585 In other words, only one among the couples of tuples $\langle c2f_1, c_1 \rangle$ and $\langle c2f_2, c_2 \rangle$
586 is required, while f_1 is always needed. One could argue that it would be
587 fairer for f_1 to receive more credit than the other four tuples, given its role
588 in producing o_1 .

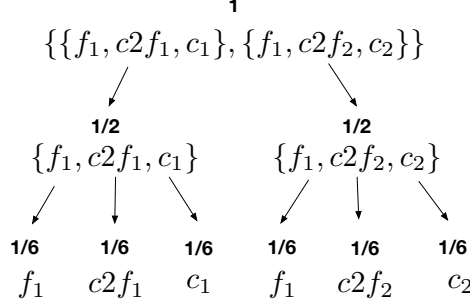


Figure 4: Exemplification of the distribution of credit through the why-provenance based DS for tuple o_1 .

589 This highlights one limitation of the DS based on lineage: while able to
 590 find all and only the relevant tuples of the output, it cannot distinguish the
 591 *importance* of tuples in the query computations. This information could be
 592 incorporated in the definition of a DS to distribute credit based on the actual
 593 role that tuples play in the computation.

594 For this reason, we present in this paper more sophisticated forms of
 595 Distribution Strategies, based on the other two provenances discussed above.

596 5.3. A Why-Provenance-Based Distribution Strategy

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

Let I be a database instance, Q a query over I , $o \in Q(I)$ an output tuple and k the total credit associated to o . Let t be a generic tuple in I . Let us call $\mathcal{W} = \text{Why}(Q, I, o)$ the witness basis of o according to Q and I , where $W \in \mathcal{W}$ is a generic witness. Let us call $\gamma(\mathcal{W}, t) : (\mathcal{W}, t) \mapsto \mathcal{P}(\mathcal{P}(\text{TupleLoc}))$ the function that returns the set $\{W \in \mathcal{W} : t \in W\}$. The tuple t receives a credit equal to:

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

597 This strategy first equally distributes the credit among the witnesses of
 598 the witness basis then, successively, it further equally divides the credit
 599 among the tuples in a witness. Since one tuple may appear in more than
 600 one witness, it will receive more than one portion of credit from the same
 601 distribution.

602 Figure 4 represents the distribution of credit with this DS with the why-
 603 provenance of tuple o_1 . The credit is first divided among the two witnesses,

604 that both receive credit $1/2$. The credit is then further divided among the
 605 tuples in each witness. Each tuple in each witness receives $1/6$ of credit. At
 606 the end of the distribution, f_1 receives a cumulative total credit of $1/3$, the
 607 other tuples receive $1/6$ each. This distribution better reflects the role of f_1
 608 in the generation of o_1 since, as we discussed, it is the only mandatory tuple
 609 for the production of the output, while we need only one of the two other
 610 couples of tuples to get the result.

611 From this example, it is immediately evident how why-provenance can
 612 better reward the tuples depending on their role. Tuples that appear in more
 613 than one witness are rewarded more than others. This means that tuples that
 614 are more important to the generation of the output, since they are used more
 615 by the query, are rewarded more than tuples that are “interchangeable” with
 616 others.

617 5.4. A How-Provenance Based Distribution Strategy

618 The how-provenance conveys more information than the why-provenance
 619 since it does not only capture what tuples are relevant to the output and in
 620 which combination, but also how they are used. To define the Distribution
 621 Strategy based on the how-provenance, we first need some other preliminary
 622 definitions.

623 Consider the provenance polynomial $\mathcal{H} = H(Q, I, o)$ of a tuple o . We
 624 define:

- 625 1. $c(\mathcal{H}) = n$ the function $c : \mathbb{N}[TupleLoc] \mapsto \mathbb{N}$ that, given a polynomial,
 626 returns the sum of its coefficients;
- 627 2. $c(M)$ the function $c : \mathcal{M} \mapsto \mathbb{N}$ that, given a monomial M , returns the
 628 sum of its exponents (with $\mathcal{M} \subset \mathbb{N}[TupleLoc]$ such that \mathcal{M} is made
 629 only by the monomials M in $\mathbb{N}[TupleLoc]$);
- 630 3. $e(t, M)$ the function $e : TupleLoc \times \mathcal{M} \mapsto \mathbb{N}$ that, given in input a
 631 tagged tuple and a monomial, returns the exponent of that tuple inside
 632 the monomial;
- 633 4. $mc(M)$ the function $mc : \mathcal{M} \mapsto \mathbb{N}$ that, given in input one monomial,
 634 returns its coefficient;
- 635 5. $\gamma(t, \mathcal{H})$ the function $\gamma : TupleLoc \times \mathbb{N}[TupleLoc] \mapsto \mathcal{M}$ that, given a
 636 tuple t and a provenance polyomial \mathcal{H} , returns the (possibly empty)
 637 set of monomials M in \mathcal{H} such that t appears in M .

638 Definition 5.4. How-Provenance-Based Distribution Strategy

639 Let I be a database instance, Q a query over I , $o \in Q(I)$ an output tuple

640 and k the total credit associated to o . Let also t be a generic tuple in I . The
 641 credit given to t is:

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

642 The how-provenance-based DS first distributes the credit to the monomi-
 643 als of the polynomial accordingly to the weight represented by their coeffi-
 644 cients, then to the single tuples in every monomial accordingly to the weights
 645 represented by their exponents.

646 Going back to the example of Table 4, consider o_1 , that has provenance
 647 polynomial $f_1 c_2 f_1 c_1 + f_1 c_2 f_2 c_2$. The DS firstly divides the credit between the
 648 two monomials. Since the coefficients are both 1, the credit is split in half.
 649 If they were, for example, 1 and 2 respectively, 1/3 of the credit would go to
 650 the first monomial, 2/3 to the second. Since in our example each variable has
 651 exponent 1, the credit is further divided equally among the three variables.
 652 Thus, at the end of the computation, f_1 receives 1/3, the other tuples receive
 653 1/6. If, for example, the first monomial was $f_1^2 c_2 f_1 c_1$, then the portion of
 654 credit of this monomial would be divided in this way: 1/2 to f_1 and 1/4 to
 655 each of the other two tuples.

656 In this specific example, the how-provenance-based distribution has the
 657 same outcome of the strategy based on why-provenance. Thus, let us consider
 658 this new query Q2, that asks for the families of type **gpcr** that have as
 659 contributor a researcher localized in the UK from GtoPdb:

```
660 Q2: SELECT DISTINCT F.name
661 FROM family as F JOIN
662 (SELECT DISTINCT f.name AS name
663 FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
664 JOIN contributor AS c ON c2f.contributor_id = c.id
665 WHERE c.country = 'UK') AS R ON F.name = R.name
666 WHERE F.type = 'gpcr'
```

667 Table 5 presents the result, composed of one tuple, annotated with the
 668 three provenances. As can be seen, lineage and why-provenance are identical
 669 to those of the tuple o_1 in the previous example. The how-provenance is
 670 different since tuple f_1 is used twice: firstly, in the join of the inner query,
 671 secondly in the join of the outer query. This information is lost in the first two

	id	name
	oxs_1	Dopamine Receptors
lineage	why-provenance	
$\{f_1, c2f_1, c_1, c2f_2, c_2\}$	$\{\{f_1, c2f_1, c_1\}, \{f_1, c2f_2, c_2\}\}$	
	how-provenance	
	$f_1(f_1c2f_1c_1 + f_1c2f_2c_2)$	

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

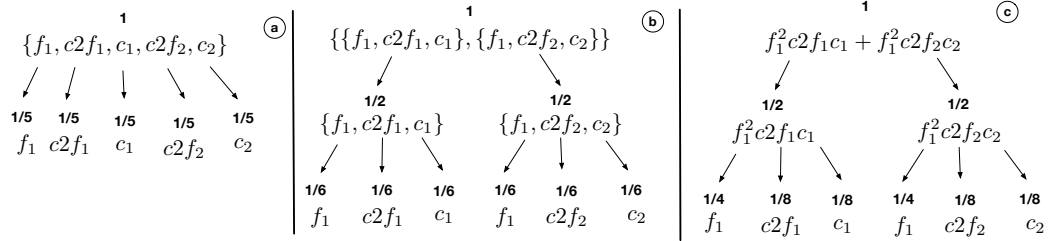


Figure 5: Comparison of different distributions strategies from the credit of tuple o_1 produced by query Q2.

provenances since they are sets, but it is maintained in the how-provenance through the use of the operator ‘.’.

Figure 5 shows the differences between the three DS for the tuple o_1 of Table 5. In subfigure 5.a we used lineage, in sub-figure 5.b we used why-provenance, and in sub-figure 5.c we used how-provenance. The DS based on the provenance polynomial gives credit 1/2 to f_1 , and 1/8 to the other tuples. This is reasonable since Q2 utilizes f_1 even more than Q1. The distribution based on how-provenance can reward f_1 more, proving that how-provenance is even more sensitive to the tuples’ role in a query than why-provenance. In this case, the why-provenance is not sensible to this difference. This is somehow a direct consequence of the fact that, as demonstrated in [30], how-provenance is more general than why-provenance and lineage, in the sense that it contains more information.

6. Experimental Evaluation: comparing provenances

We evaluate the proposed distribution strategies on GtoPdb, and in we focus on target families, all of those are described in webpages. GtoPdb particular identifies eight family types: *GPCR*, *Ion channels*, *NHRs*, *Kinases*, *Catalytic receptors*, *Transporters*, *Enzymes* and *Other protein targets*.

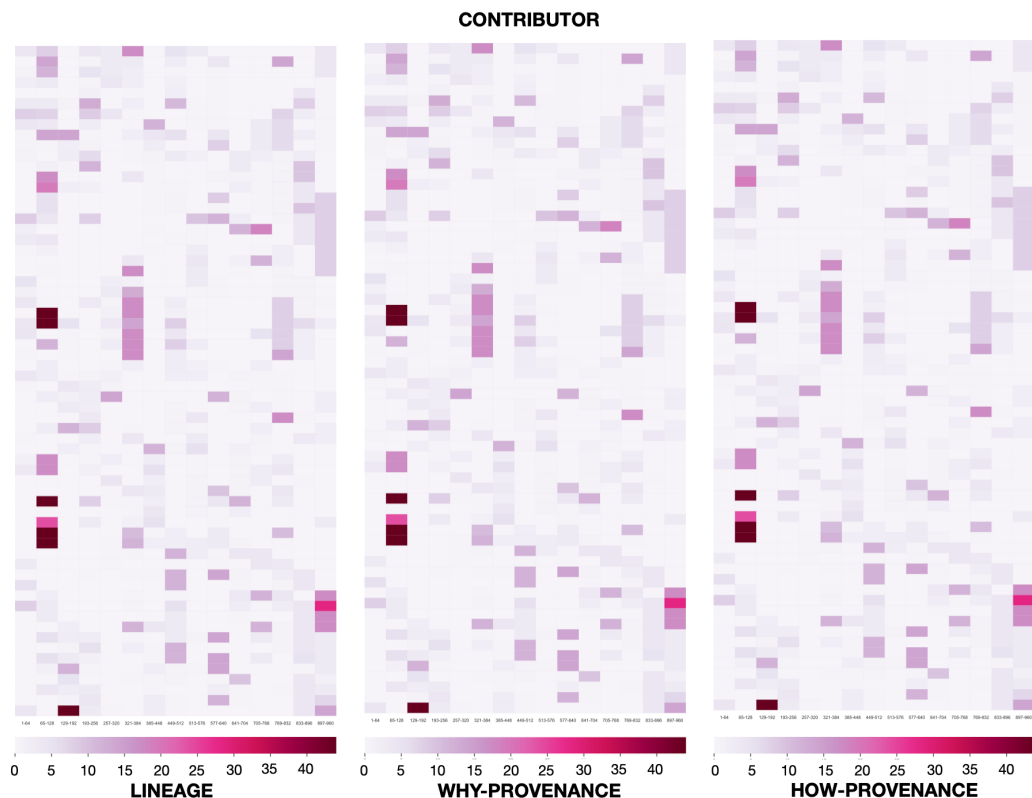


Figure 6: Comparison of three DS on the same table `contributor` using the distribution given by the queries retrieved from papers.

When a paper uses data from GtoPdb, it can cite the full database, the family webpage of interest, or a subset of data extracted with a query. In this work, we consider a full-fledged data citation context in which papers cite the specific *data* subset of interest and not the webpage or the full database acting as data proxies. Therefore, when a paper cites family data, it is citing a set of queries needed to retrieve all the information provided by the family webpage, i.e., one query for each section composing a page, as depicted in Figure 3. In the figure, we can see how the structure of one family, “Adenosine receptors”, is mapped into several queries to obtain the information to build the corresponding webpage. In GtoPdb, all family pages share a similar structure (the only differences may be the presence/absence and length of the receptors lists, further readings, and contributors sections). Therefore, the same queries are used to build all other pages by simply changing the

703 family id (which, in our example, is 3). All these queries are SPJ.

704 As already stated, many papers that draw information from the GtoPdb
705 website¹¹ cite papers published every two years by the GtoPdb Committee on
706 Receptor Nomenclature and Drug Classification (NC-IUPHAR). To obtain
707 a set of citations capable of representing what happens, we consider a paper
708 subset citing the 2018 GtoPdb [31] data paper. At the time of writing, this
709 paper received more than 1200 citations.

710 As explained in Section 3, in the papers published in the British Journal of
711 Clinical Pharmacology, that cite GtoPdb, the name of families are hyperlinks
712 that point to the corresponding webpages. We considered all the 460 papers
713 in BJCP citing [31] as of February 2020. We automatically extracted the
714 URL references to family pages were automatically extracted to guide in
715 building the queries to produce corresponding webpages. A total of 5,945
716 different queries were built in this way.¹²

717 Figure 6 shows the heat-maps obtained by three different DS on the table
718 **contributor**. It is immediately evident that the result of the distribution is
719 the same with the three strategies. The same effect is also obtained in the
720 other tables of the database used by the considered queries. Why is that? It
721 is the case that the conditions in which we produced this experiment are quite
722 peculiar. The queries that we used share similar characteristics. They are all
723 SPJ queries, each of them utilizes each table only once in the join condition
724 (there are no self-joins), and all the joins are made using key attributes.
725 In this particular condition, each tuple of the output presents: (i) a how-
726 provenance that is a single monomial with coefficient 1 and exponent 1 in
727 each variable; (ii) a why-provenance that is composed by only one witness;
728 (iii) a lineage that coincides with the only witness in the witness basis. It is
729 easy to see how, given these queries, the three distributions act in the same
730 way. The credit is always uniformly distributed among the tuples appearing
731 in each provenance.

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

734 Q3: `SELECT c.first_names, c.surname`

¹¹<https://www.guidetopharmacology.org>

¹²For reproducibility purposes, the code we used for our experiments and all the produced queries can be found at the following link: https://bitbucket.org/dennis_dosso/credit_distribution_project.

```

735 FROM contributor2family AS cf JOIN contributor AS c ON
736 cf.contributor_id = c.contributor_id
737 WHERE f.family_id = 3

```

738 Q3 returns a series of 10 tuples from the version of GtoPdb we considered.
739 The first tuple produced by this query, <Bertil B., Fredholm>, has $c_{939} \cdot$
740 $c_{2f_{496}}$ as provenance polynomial. c_{939} represents the provenance token of a
741 tuple in **contributor**, the same for $c_{2f_{496}}$ in table **contributor2family**. It
742 is easy to see that the why-provenance of this tuple is $\{\{c_{939}, c_{2f_{496}}\}\}$ and its
743 lineage is $\{c_{939}, c_{2f_{496}}\}$. Therefore, the credit assigned to these tuples is 1/2
744 using all three DS. This actually happens for each tuple of the output of each
745 query of GtoPdb, thus making the distributions equivalent.

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

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

```

752 Q4: SELECT f.name AS name
753 FROM family AS F JOIN
754 (SELECT DISTINCT f.family_id, f.name
755 FROM "family" AS f JOIN contributor2family AS cf ON
756 f.family_id = cf.family_id
757 JOIN contributor c ON
758 cf.contributor_id = c.contributor_id
759 WHERE c.country = 'UK') AS R
760 ON F.name = R.name

```

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

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

$$\begin{aligned}
&f_{625}(f_{625}c_{2f_{656}}c_{184} + f_{625}c_{2f_{113}}c_{180} + f_{625}c_{2f_{283}}c_{198} + \\
&\quad + f_{625}c_{2f_{550}}c_{865} + f_{625}c_{2f_{573}}c_{101} + f_{625}c_{2f_{95}}c_{109})
\end{aligned}$$

766 As already discussed, the different monomials represent possible *alternatives*
767 of combinations of tuples that produce the considered output tuple.

768 Tuple f_{625} is used each time with different joins, thus it appears in each
 769 monomial. The last join, performed in the outmost query, is responsible
 770 for the final multiplication of f_{625} with the rest of the polynomial between
 771 parenthesis.

772 From this polynomial we compute the why-provenance as a set of six
 773 different witnesses:

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

774 And corresponding lineage:

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

775 This was only one tuple among the 86 obtained from this query. If we
 776 assign credit 1 to all these tuples and distribute it with the different strategies,
 777 we obtain the result shown in Figure 7 for the table **contributor**. At first
 778 sight, it may appear that the three distributions produce the same result.
 779 This is only partially true: the heat maps appear equal, but the absolute
 780 values assigned to each tuple are different. This is more evident if we look
 781 at the legend of each heat-map, where the maximum quantity of credit is
 782 different for each distribution. The one performed through lineage is around
 783 1.8, the why-provenance's one is around 1.4, and the one based on how-
 784 provenance is around 1.1.

785 To understand what is happening with this query in this specific ta-
 786 ble, consider the output tuple **<Histamine receptors>** and its provenances,
 787 as discussed above. Let us focus on its lineage. There are a total of six
 788 authors for this family and 13 tuples in total in the lineage. Thus, using
 789 the lineage-based DS, each tuple belonging to the **contributor** table (i.e.
 790 $c_{184}, c_{180}, c_{198}, c_{865}, c_{101}, c_{109}$) receives credit equal to $1/13$. Tuple f_{625} too re-
 791 ceives a portion of credit equal to $1/13$.

792 Let us consider now why-provenance. Tuple f_{625} appears six times in
 793 six different witnesses composed of 3 elements each. From each witness it
 794 receives a portion of credit equal to $1/18$, thus its total credit is $1/3$. On the
 795 other hand, all the authors appear only once in each witness, thus each of
 796 them receives credit $1/18$. In this case, why-provenance is recognizing more

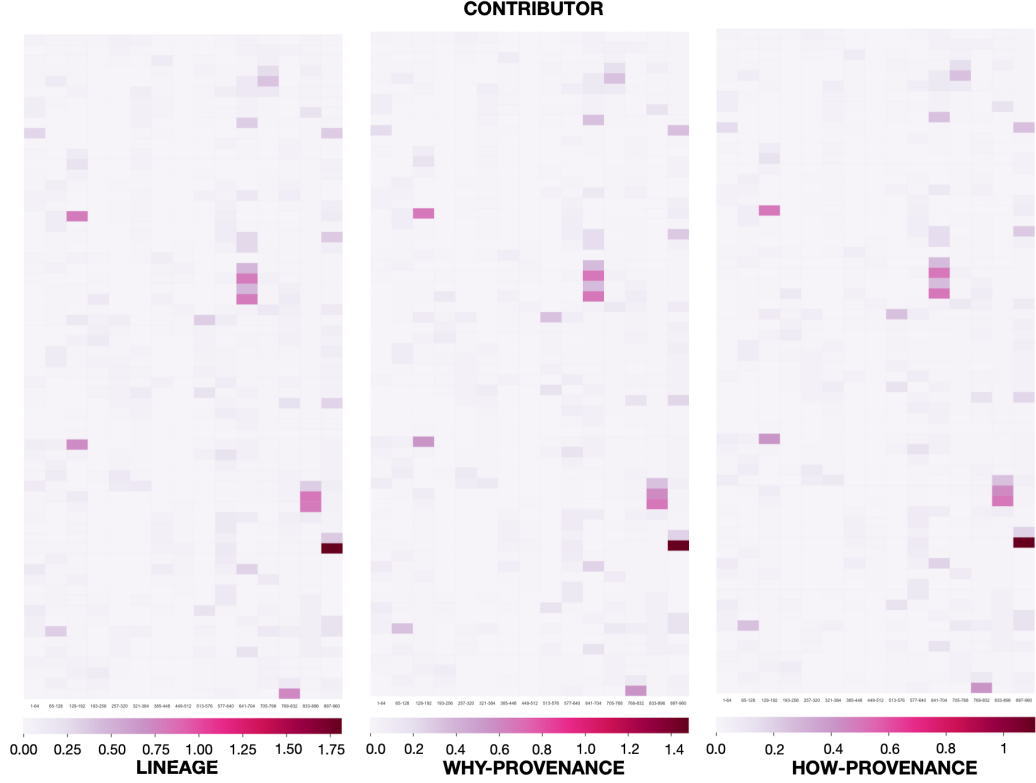


Figure 7: Comparison of three DS on the same table `family` after the distribution of the credit connected to query Q4.

credit to tuple f_{625} , since it appears in each witness. The consequence is that this distribution is equally *subtracting* credit from the other tuples in the witnesses and giving it to f_{625} . In Figure 7 we are only looking at table `contributor`. This same effect is reproduced for each tuple of the output of query Q4, thus the *absolute* credit values on the tuples vary depending on the deployed strategy. What happens is that the tuples in table `contributor` receive less credit than the one received using lineage, but in the same proportions. The heat map appears thus equal to the one obtained with lineage. This same effect is also present with the how-provenance-based CDS. In this case, tuple f_{625} is rewarded even more, since it appears with an exponent 2 in each monomial, thus attracting even more credit.

This is also why when we look at the legend for each part of Figure 7, the maximum value reached with the lineage-based DS is higher than the

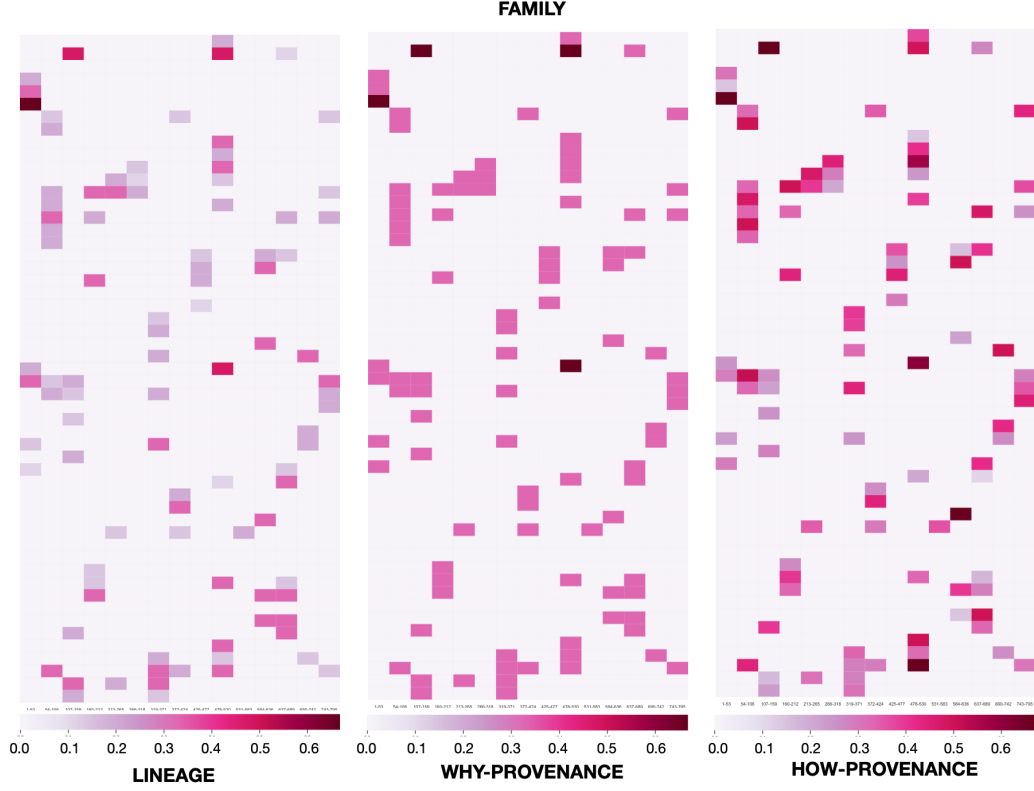


Figure 8: Comparison of three DS on the same table **family** after the distribution computed on provenances randomly generated.

one reached with the why-provenance-based DS, which in turn is higher than the one obtained with the how-provenance. This is because the different strategies reward less and less the tuples of table **contributor** and more the ones in table **family**.

This clearly shows the ability of the different strategies to adapt to situations. All three of them can highlight the relevant tuples in the table. However, they differ in the way they reward the tuples. Depending on the task, one provenance can be preferred to the other. If the only interest is to highlight the relevant tuples, lineage is sufficient. If the interest is also to reward more the tuples that are fundamental to the output, one can also choose why- or how-provenance, knowing that how-provenance rewards even more than why-provenance the relevant tuples that are indispensable for the output.

Let us consider another interesting case we show in Figure 8. The figure reports a distribution of credit performed on **family** through the generation of *synthetic* polynomials. In this last case, we did not produce full-fledged queries. Rather, we randomly generated provenance polynomials that might be the how-provenance of randomly generated synthetic queries. An example of such synthetic polynomial is:

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

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

$$\{\{f_1, c_2f_1, c_1\}, \{f_1, c_2f_2, cf_2\}, \{f_5, c_2f_{17}, c_{18}\}\}$$

its lineage is:

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

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

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

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

Using why-provenance, this is the output:

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

Finally, with lineage, this is the distribution:

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

To highlight how the distributions behave differently with these polynomials, consider tuple f_5 . f_5 receives the highest quantity of credit when we use

844 the lineage-based distribution. Why-provenance and how-provenance reduce
845 its quantity of credit since more information is available for the computation
846 and the algorithms weigh less and less its role.

847 Generally speaking, the more complex the distribution, the more polar-
848 ized the credit is toward the tuples that are used more frequently or with a
849 higher impact in the production of the output tuple. Looking at the heat-
850 maps of Figure 8, it appears that lineage tends to distribute credit more
851 “equally” among the tuples, with only one or two tuples receiving higher
852 quantities of credit, primarily because they are used in many different queries.

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

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

868 For our last set of experiments, consider Figure 9. We still use the 100
869 polynomials described above and the credit distributed through them. Since
870 these polynomials correspond to queries whose corresponding authors are not
871 easily identifiable, we considered 20 “synthetic” authors, and we randomly
872 assigned one author to each tuple in the database. The authors receive
873 “blocks” of consecutive tuples, with each block of the size varying between
874 10 and 40. Every time a tuple was used in a provenance polynomial, we
875 assigned one citation to the author corresponding to the tuple. The same
876 author also receives the three different credits assigned to the tuple at the
877 end of the distribution process using the three DS.

878 Figure 9 presents the radar plot where the 20 authors are sorted based on
879 the normalized number of received citations, together with the corresponding
880 normalized quantities of credits. Credit presents a different behavior from
881 one of the citations, and each form of credit, i.e., the credit obtained from

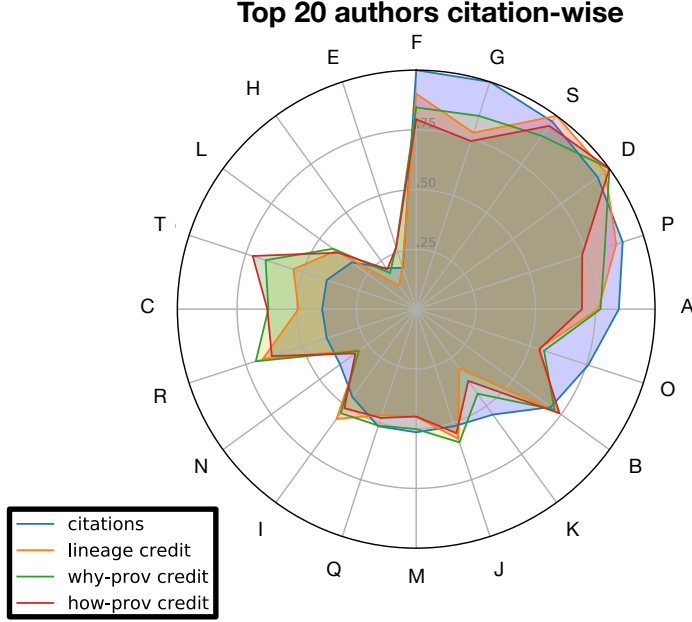


Figure 9: Top 20 authors by number of citations and their credit given through the three different DS.

the different DS, behaves differently from the others. For example, it appears that authors T, C, and R that are low in the number of citations are still rewarded more than other more cited authors in terms of credit. Even if the tuples of these authors received fewer citations, they still received more credit than other more cited tuples. This shows how credit can be an effective new method to use together with traditional citations to reward curators, highlighting aspects lost using the traditional bibliometrics.

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

898 7. Conclusions

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

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

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

933 In the fourth set of experiments, we showed how, compared with tra-
934 ditional citations, the credit distributed with the three strategies works as

935 a new tool highlighting different aspects of an author’s role in the research
936 context identified by queries. Authors with a limited number of citations
937 can still have a high quantity of credit due to the importance of the data to
938 which they contributed to the queries.

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

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

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

963 References

- 964 [1] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bern-
965 stein, P., Boncz, P., Chaudhuri, S., Cheung, A., Doan, A., Dong, L.,
966 Franklin, M. J., Freire, J., Halevy, A., Hellerstein, J. M., Idreos, S., Koss-
967 mann, D., Kraska, T., Krishnamurthy, S., Markl, V., Melnik, S., Milo,
968 T., Mohan, C., Neumann, T., Chin Ooi, B., Ozcan, F., Patel, J., Pavlo,
969 A., Popa, R., Ramakrishnan, R., Ré, C., Stonebraker, M., and Suciu, D.

- (2020). The seattle report on database research. *SIGMOD Rec.*, 48(4):44–53.
- [2] Alawini, A., Davidson, S. B., Hu, W., and Wu, Y. (2017). Automating data citation in citedb. *PVLDB*, 10(12):1881–1884.
- [3] Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., and Wu, Y. (2018). Data citation: A new provenance challenge. *IEEE Data Eng. Bull.*, 41(1):27–38.
- [4] Altman, M., Borgman, C. L., Crosas, M., and Martone, M. (2015). An Introduction to the Joint Principles for Data Citation. *Bulletin of the Association for Information Science and Technology*, 41(3):43–45.
- [5] Baggerly, K. (2010). Disclose all data in publications. *Nature*, 467(7314):401–401.
- [6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D., Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and Goble, C. A. (2013). Why linked data is not enough for scientists. *Future Gener. Comput. Syst.*, 29(2):599–611.
- [7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.
- [8] Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B., and Lamere, P. (2011). The million song dataset.
- [9] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Communication*, pages 93–116. De Gruyter Mouton.
- [10] Buneman, P. (2006). How to cite curated databases and how to make them citable. In *18th International Conference on Scientific and Statistical Database Management, SSDBM*, pages 195–203. IEEE Computer Society.
- [11] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D., Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn’t working, and what to do about it. *Database*.

- 1000 [12] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation
1001 is a computational problem. *Commun. ACM*, 59(9):50–57.
- 1002 [13] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A
1003 characterization of data provenance. In *Database Theory - ICDT 2001*,
1004 *8th International Conference*, pages 316–330.
- 1005 [14] Buneman, P. and Silvello, G. (2010). A rule-based citation system for
1006 structured and evolving datasets. *IEEE Data Eng. Bull.*, 33(3):33–41.
- 1007 [15] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N.,
1008 Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry,
1009 R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a.,
1010 and Wright, D. (2012). Making Data a First Class Scientific Output:
1011 Data Citation and Publication by NERC’s Environmental Data Centres.
1012 *International Journal of Digital Curation*, 7(1):107–113.
- 1013 [16] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Jour-
1014 nals: A Survey. *Journal of the Association for Information Science and*
1015 *Technology*, 66(9):1747–1762.
- 1016 [17] Cheney, J., Chiticariu, L., and Tan, W. (2009). Provenance in databases:
1017 Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–
1018 474.
- 1019 [18] CODATA-ICSTI Task Group on Data Citation Standards and Practices
1020 (2013). *Out of Cite, Out of Mind: The Current State of Practice, Policy,*
1021 *and Technology for the Citation of Data*, volume 12.
- 1022 [19] Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., and Simons, N.
1023 (2019). Bringing citations and usage metrics together to make data count.
1024 *Data Science Journal*, 18(1).
- 1025 [20] Cronin, B. (1984). *The citation process. The role and significance of*
1026 *citations in scientific communication*. London: Taylor Graham.
- 1027 [21] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evi-
1028 dence of a structural shift in scholarly communication practices? *JASIST*,
1029 52(7):558–569.

- 1030 [22] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of
1031 view data in a warehousing environment. *ACM Trans. Database Syst.*,
1032 25(2):179–227.
- 1033 [23] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model
1034 for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on*
1035 *Innovative Data Systems Research*. www.cidrdb.org.
- 1036 [24] Dosso, D. and Silvello, G. (2020). Data credit distribution: A
1037 new method to estimate databases impact. *Journal of Informetrics*,
1038 14(4):101080.
- 1039 [25] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The vir-
1040 tual atomic and molecular data centre (VAMDC) consortium. *Journal of*
1041 *Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.
- 1042 [26] Fang, H. (2018). A discussion of citations from the perspective of the
1043 contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–
1044 1520.
- 1045 [27] Force, M., Robinson, N., Matthews, M., Auld, D., and Boletta, M.
1046 (2016). Research data in journals and repositories in the web of science:
1047 Developments and recommendations. *Bulletin of IEEE Technical Com-*
1048 *mittee on Digital Libraries, Special Issue on Data Citation*, 12(1):27–30.
- 1049 [28] Garfield, E. (1999). Journal impact factor: a brief review.
- 1050 [29] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data
1051 identification and process monitoring for reproducible earth observation
1052 research. In *2019 15th International Conference on eScience (eScience)*,
1053 pages 28–38. IEEE.
- 1054 [30] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance
1055 semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-*
1056 *SIGART symposium on Principles of database systems*, pages 31–40. ACM.
- 1057 [31] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson,
1058 A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton,
1059 S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F.,
1060 Spedding, M., Davies, J. A., and Nc-Iuphar (2018). The IUPHAR/BPS

- 1061 guide to PHARMACOLOGY in 2018: updates and expansion to encom-
 1062 pass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Re-*
 1063 *search*, 46(Database-Issue):D1091–D1106.
- 1064 [32] Hartley, J. (2017). Authors and their citations: a point of view. *Scien-*
 1065 *tometrics*, 110(2):1081–1084.
- 1066 [33] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a
 1067 transformed scientific method.
- 1068 [34] Honor, L. B., Haselgrove, C., Frazier, J. A., and Kennedy, D. N. (2016).
 1069 Data citation in neuroimaging: proposed best practices for data identifi-
 1070 cation and attribution. *Frontiers in neuroinformatics*, 10:34.
- 1071 [35] Joshi-Tope, G., Gillespie, M., Vastrik, I., D’Eustachio, P., Schmidt, E.,
 1072 de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis,
 1073 S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of bio-
 1074 logical pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432.
- 1075 [36] Katz, D. (2014). Transitive credit as a means to address social and
 1076 technological concerns stemming from citation and attribution of digital
 1077 products. *Journal of Open Research Software*, 2(1).
- 1078 [37] Katz, D. S., Hong, N., Clark, T., Fenner, M., and Martone, M. (2020).
 1079 Software and data citation. *Computing in Science & Engineering*, 22 (2):4–
 1080 7.
- 1081 [38] Kosten, J. (2016). A classification of the use of research indicators.
 1082 *Scientometrics*, 108(1):457–464.
- 1083 [39] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S.
 1084 (2011). Citation and Peer Review of Data: Moving Towards Formal Data
 1085 Publication. *International Journal of Digital Curation*, 6(2):4–37.
- 1086 [40] Martone, M. (2014). Joint declaration of data citation
 1087 principles. *FORCE11*. San Diego CA. *Data Citation Syn-*
 1088 *thesis Group*. doi: <https://doi.org/10.25490/a97f-egykh>, url:
 1089 <https://www.force11.org/datacitationprinciples> (visited on 2020/03/17).
- 1090 [41] Meho, L. I. and Yang, K. (2007). Impact of data sources on citation
 1091 counts and rankings of LIS faculty: Web of science versus scopus and

- 1092 google scholar. *Journal of the american society for information science*
1093 *and technology*, 58(13):2105–2125.
- 1094 [42] Milo, T. (2019). Getting rid of data. *Journal of Data and Information*
1095 *Quality (JDIQ)*, 12(1):1–7.
- 1096 [43] Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D.,
1097 Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G.,
1098 Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff,
1099 D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D.,
1100 Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson,
1101 E., McNutt, M., Miguel, M., Paluck, E. L., Simonsohn, U., Soderberg, C.,
1102 Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers,
1103 E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research
1104 culture. *Science*, 348(6242):1422–1425.
- 1105 [44] Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J.
1106 (2016). Research data explored: An extended analysis of citations and
1107 altmetrics. *Scientometrics*, 107(2):723–744.
- 1108 [45] Pröll, S. and Rauber, A. (2013). Scalable data citation in dynamic, large
1109 databases: Model and reference implementation. In *Proceedings of the*
1110 *2013 IEEE International Conference on Big Data*, pages 307–312. IEEE.
- 1111 [46] Rauber, A., Ari, A., van Uytvanck, D., and Pröll, S. (2016). Identi-
1112 fication of Reproducible Subsets for Data Citation, Sharing and Re-Use.
1113 *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue*
1114 *on Data Citation*, 12(1):6–15.
- 1115 [47] Rauber, A., Asmi, A., van Uytvanck, D., and Proell, S. (2015). Data
1116 citation of evolving data: Recommendations of the working group on data
1117 citation (wgdc). *Result of the RDA Data Citation WG*, 20.
- 1118 [48] Silvello, G. (2018). Theory and practice of data citation. *J. Assoc. Inf.*
1119 *Sci. Technol.*, 69(1):6–20.
- 1120 [49] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data
1121 provenance in e-science. *SIGMOD Record*, 34(3):31–36.
- 1122 [50] Spengler, S. (2012). Data Citation and Attribution: A Funder’s Per-
1123 spective. In of Sciences’ Board on Research Data, N. A. and Information,

- 1124 editors, *Report from Developing Data Attribution and Citation Practices*
 1125 *and Standards: An International Symposium and Workshop*, pages 177–
 1126 178. National Academies Press: Washington DC.
- 1127 [51] Ur Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah,
 1128 T. V., and Khan, S. U. (2016). Big data reduction methods: a survey.
 1129 *Data Science and Engineering*, 1(4):265–284.
- 1130 [52] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G.,
 1131 Axton, M., Baak, A., Blomberg, N., Boiten, J., da Silva Santos, L. B.,
 1132 Bourne, P. E., et al. (2016). The fair guiding principles for scientific data
 1133 management and stewardship. *Scientific data*, 3.
- 1134 [53] Wu, Y., Alawini, A., Davidson, S. B., and Silvello, G. (2018). Data
 1135 citation: Giving credit where credit is due. In *Proceedings of the 2018*
 1136 *International Conference on Management of Data, SIGMOD*, pages 99–
 1137 114.
- 1138 [54] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to
 1139 scientific datasets using article citation networks. *Journal of Informetrics*,
 1140 14(2).
- 1141 [55] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of
 1142 new researchers using the zp-index. *Scientometrics*, 106(3):901–916.
- 1143 [56] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for
 1144 Datasets Citation and Extraction Reproducibility in VADMC. *Journal of*
 1145 *Molecular Spectroscopy*, 327:122–137.