

# Credit Distribution through Data Provenance in Relational Scientific Databases

Dennis Dosso<sup>a</sup>, Susan B. Davidson<sup>b</sup>, Gianmaria Silvello<sup>a</sup>

<sup>a</sup>*Department of Information Engineering, University of Padua, Italy*

<sup>b</sup>*Department of Computer and Information Science, University of Pennsylvania, United States*

---

## Abstract

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

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

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

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

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

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

*Keywords:* Data Citation, Data Credit

---

## 1 Introduction

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

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

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

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

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

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

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

The DCD problem differs from the traditional citation setting since:

1. In a traditional setting, when a paper cites another paper, a +1 “credit” is given to the cited paper (and to its authors). It does not matter why or how paper  $p_1$  cites paper  $p_2$ <sup>1</sup>, 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.

---

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



Figure 1: Overview of the credit distribution pipeline.

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

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

The DCD process is summarized in Figure 1:

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

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

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

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

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

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

139 For evaluation, we use a well-known curated database, the IUPHAR/BPS<sup>3</sup>  
 140 Guide to Pharmacology [31], also known as GtoPdb<sup>4</sup>, which contains ex-  
 141 pertly curated information about diseases, drugs, cellular drug targets, and  
 142 their mechanisms of action. We chose GtoPdb for two main reasons: (i) it  
 143 is a widely-used and valuable curated relational database, (ii) many papers  
 144 in the literature use, and cite its data (i.e., families, ligands, and receptors).  
 145 Real queries used in papers can therefore be seen as data citations which, in  
 146 turn, can be used to assign data credit.

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

---

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

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

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

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

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

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

## 165 2. Background

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

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

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

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

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

## 208 2.1. Data Citation in Relational Databases

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



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

In one of its most recent sessions [47], the Working Group (WG) on Data Citation reported that there are various implementations of its guidelines for Data Citation on MySQL/Postgres relational databases. Some of these databases are: DEXHELPP<sup>6</sup> (Social Security Records); NERC (ARGO Global Array); EODC (Earth Observation Data Centre) [29]; LNEC (River dam monitoring); MDS (Million Song Database) [8]; CBMI<sup>7</sup> (Center for Biomedical Informatics); VMC (Vermont Monitoring Cooperative); CCA<sup>8</sup> (Climate Change Center Austria); VAMDC (Virtual Atomic and Molecular Data Center) [25, 56].

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

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

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

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

---

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

<sup>6</sup><http://www.dexhelpp.at/>

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

<sup>8</sup><https://ccca.ac.at/startseite>

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

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

## 2.2. Data Credit

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

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

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

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

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

301 [26] presents a framework to distribute the credit generated by a paper to  
 302 its authors and to the papers in its reference list in a transitive way. Let us  
 303 consider the *citation graph* as the graph where the nodes are papers and the  
 304 links are the citations among them. In this graph, every paper is a source of  
 305 credit, which is then transferred to the neighboring nodes. The quantity of  
 306 credit received by each cited paper depends on its impact/role in the citing  
 307 paper. So far, this theoretical framework is limited to papers, but it can be  
 308 easily extended to a citation graph including both papers and data.

309 [54] proposes the first method to compute credit within a network of  
 310 papers citing data. Adopting a network flow algorithm, they simulate a  
 311 random walker to estimate a score for each dataset, leveraging real-world  
 312 usage data to compute the credit. This is the first step towards an automatic  
 313 credit computation procedure. This proposal is, however, limited to assigning  
 314 credit to whole datasets, and it does not deal with the granularity of data.  
 315 It does not work to assign credit to a single research entity within a dataset.  
 316 Differently from [54], we do not treat the credit computation process, but we  
 317 focus on the distribution process.

### 318 2.3. Data Provenance

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

329 [17] presents four main types of data citation for DBMS: *lineage* [22],  
 330 *why-provenance* [13], *how-provenance* [30] and *where-provenance* [13].

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

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

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

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

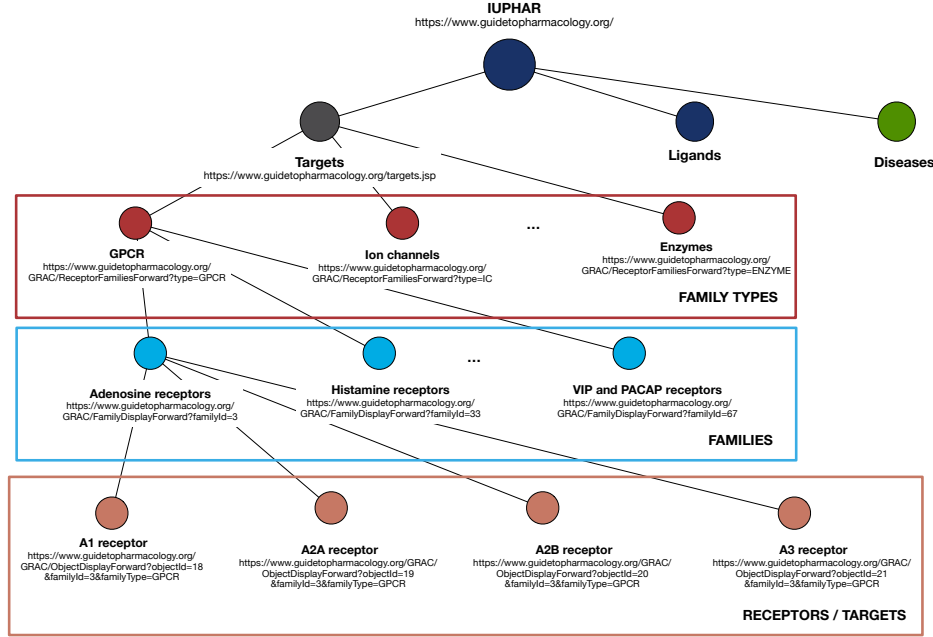


Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

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

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

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

### 362 3. Use Case: GtoPdb

363 As use case we refer to the IUPHAR/BPS Guide to Pharmacology [31]  
 364 or GtoPdb<sup>9</sup>. GtoPdb is a well-known and well structured scientific relational  
 365 database that contains expertly curated information about diseases, drugs  
 366 in clinical use, their cellular targets, and the mechanisms of action on the  
 367 human body. It is curated and maintained by the GtoPdb Committee, and

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

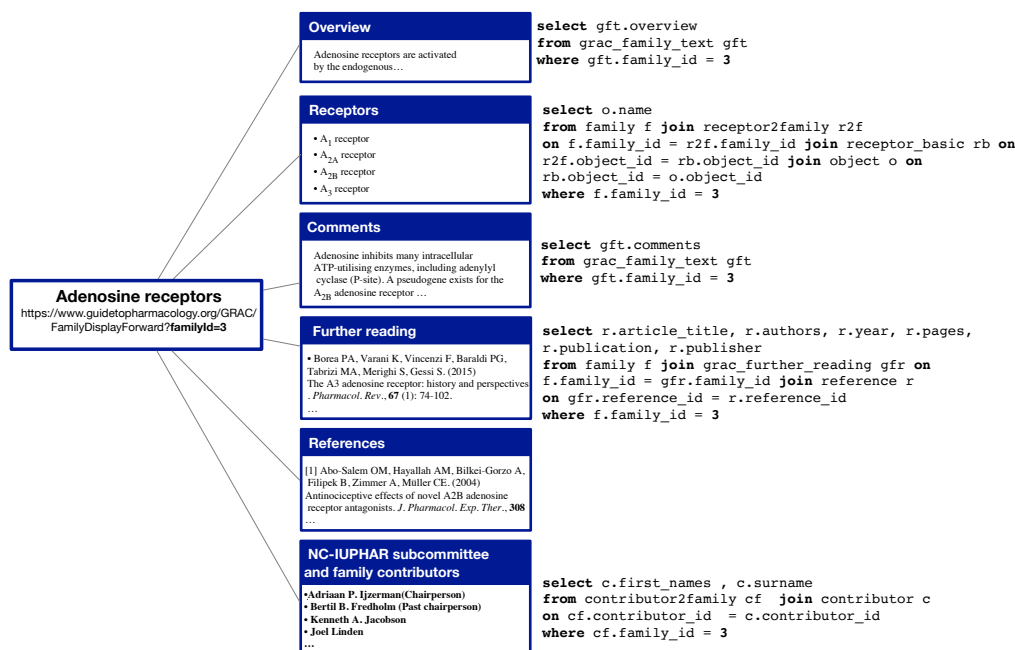


Figure 3: Basic web-page structure of “Adenosine receptors” family (ID 3), with queries used to retrieve the information contained in every section, except references.

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

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

GtoPdb provides access to the webpages corresponding to all these nodes

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

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

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

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

---

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

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

contributor		
id	Name	Country
$c_1$	John Smith	UK
$c_2$	Jim Doe	UK
$c_3$	Hans Zimmerman	Germany
$c_4$	Roberta Rossi	Italy

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

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

#### 4. Data Provenances

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

##### 4.1. Lineage

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

```

Q1: SELECT DISTINCT f.name
FROM family AS f JOIN contributor2family AS c2f
ON f.id = c2f.family_id
JOIN contributor AS c ON c2f.contributor_id = c.id
WHERE c.country = 'UK'

```



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

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

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

#### 445 4.2. Why-Provenance

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

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

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

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

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

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

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

#### 479 4.3. How-Provenance

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

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

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

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

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

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

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

## 5. Credit Distribution and Distribution Strategies

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

### 5.1. Data Credit and Data Credit Distribution

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

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

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

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

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

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

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

## 549 5.2. A Lineage-based Distribution Strategy

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

**Definition 5.2. Lineage-based Distribution Strategy [24]**

*Let  $I$  be a database instance,  $Q$  a query over  $I$ ,  $o \in Q(I)$  an output tuple and  
 $k_o$  the credit associated to  $o$ . Let  $L$  be the lineage of  $o$  and  $t$  be a tuple in  $I$ ,*

then  $t$  receives credit equal to:

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

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

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

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

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

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

### 582 5.3. A Why-Provenance-Based Distribution Strategy

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

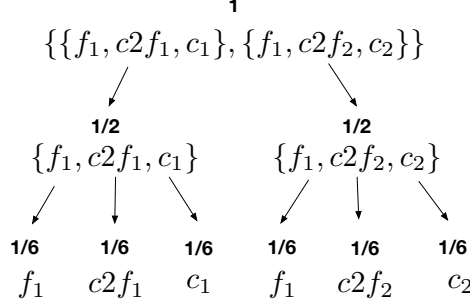


Figure 4: Distribution of credit using why-provenance-based DS for tuple  $o_1$ .

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

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

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

Then tuple  $t$  in  $I$  receives credit equal to:

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

where  $\gamma$  is a function which returns all witnesses  $W$  in which  $t$  appears:

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

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

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

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

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

#### 5.4. A How-Provenance Based Distribution Strategy

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

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

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

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

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

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

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

id	name
$oxs_1$	Dopamine Receptors

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

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

628        Going back to the example of Table 4, consider  $o_1$  with provenance poly-  
629        nomial  $f_1 c2f_1 c_1 + f_1 c2f_2 c_2$ . The how-provenance-based DS firstly divides  
630        the credit between the two monomials. Since the coefficients of each mono-  
631        mial are 1, the credit is split in half. If they were, for example, 1 and 2  
632        respectively, 1/3 of the credit would go to the first monomial, and 2/3 to  
633        the second. Since in our example each variable has exponent 1, the credit  
634        is further divided equally among the three variables. Thus, at the end of  
635        the computation,  $f_1$  receives 1/3, and the other tuples receive 1/6. If, for  
636        example, the first monomial was  $f_1^2 c2f_1 c_1$ , then the portion of credit of this  
637        monomial would be divided in this way: 1/2 to  $f_1$  and 1/4 to each of the  
638        other two tuples.

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

```

643        Q2: SELECT DISTINCT F.name
644        FROM family as F JOIN
645        (SELECT DISTINCT f.name AS name
646        FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
647        JOIN contributor AS c ON c2f.contributor_id = c.id
648        WHERE c.country = "UK") AS R ON F.name = R.name
649        WHERE F.type = "gpcr"

```

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



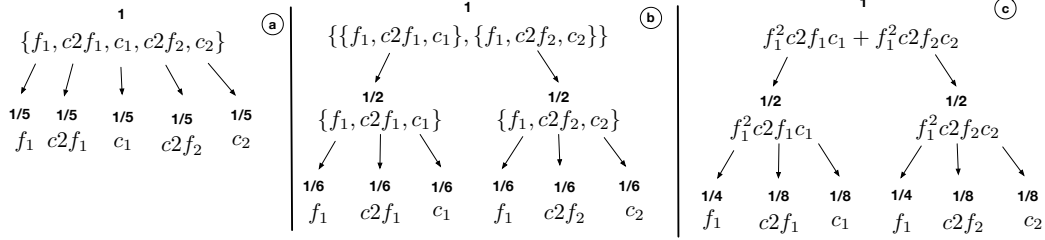


Figure 6: Comparison of different distributions strategies for tuple  $o_1$  produced by query Q2.

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

## 6. Experimental Evaluation: comparing provenances

To understand the tradeoff between these distribution strategies, we performed three sets of experiments using queries over GtoPdb. The first set of experiments used real queries extracted from citations to GtoPdb published in the British Journal of Pharmacology. The second set uses different sets of synthetically produced provenance polynomials, corresponding to more complex queries, apt to highlight the differences between the DS. In the third set of experiments we compare the differences between traditional citations and credit in rewarding data curators.

### 6.1. Real-world queries

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

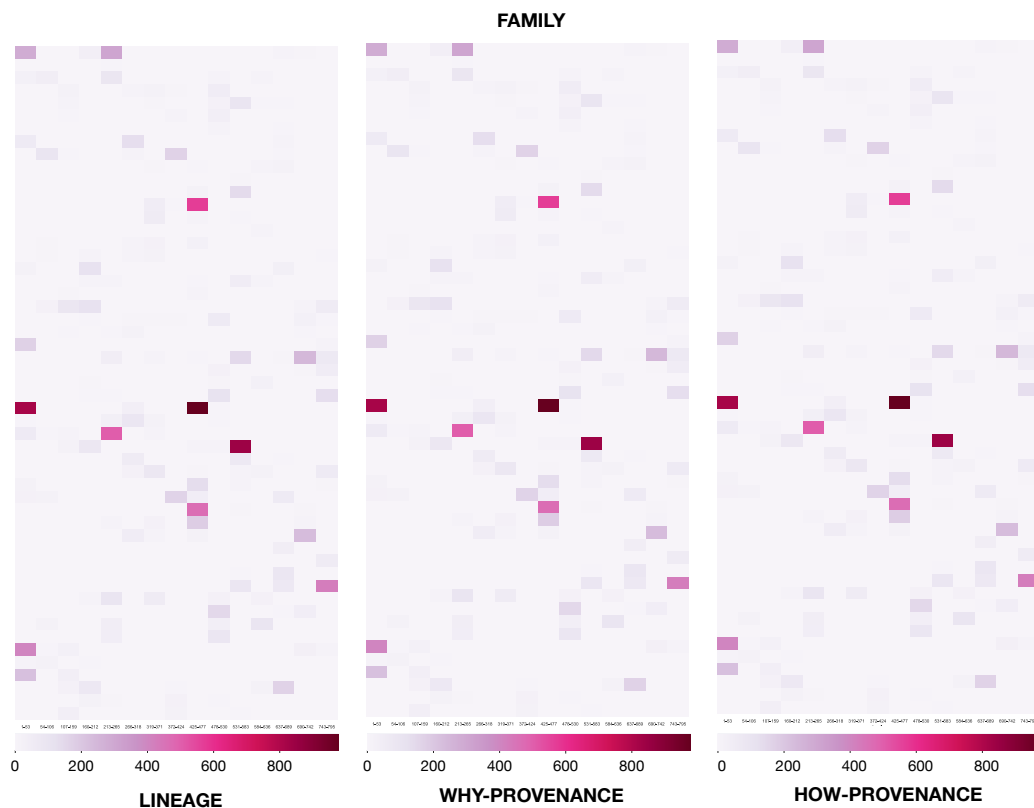


Figure 7: Comparison of three DS on the same table `family` using the distribution given by the queries retrieved from papers.

When a paper uses data from GtoPdb, it can cite the full database, the webpage of interest, or a subset of data extracted with a query. We consider as sources of citations the papers published in the British Journal of Pharmacology (BJP) <sup>11</sup>, since each time they cite a webpage from GtoPdb, they report the URL of that page. From that URL it is possible to reverse-engineer the queries that are used to obtain the data contained in the pages. In particular, we considered all the 889 papers in BJCP citing [31] as of October 2020. [31] is a data journal that describes the structure and evolution of GtoPdb. ERach two years the GtoPdb consortium releases such a journal to describe the evolution of the databases. At the time of writing, this paper

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

692 received more than 1200 citations.

693 The queries that we inferred are the ones building the information that  
694 constitutes a target family webpage are depicted in Figure 3 (with the excep-  
695 tion of the section References). In the figure, we can see how the structure of  
696 one family, “Adenosine receptors”, is mapped into several queries to obtain  
697 the information to build the corresponding webpage. In GtoPdb, all target  
698 family pages share a similar structure (the only differences is that certain  
699 sections, such as “contributors” or “further readings”, may be absent). The  
700 same queries can therefore be used to build all the target family pages by  
701 simply changing the family id used in the query (which, in the example of  
702 Figure 3, is 3). All these queries are SPJ. A total of more than 12K different  
703 queries were built in this way<sup>12</sup>. We assumed that each tuple in the output  
704 of each of these queries carries credit 1.

705 Figure 7 shows the heat-maps obtained by the distribution of credit per-  
706 formed by the three different DS on the GtoPdb table `family`. `family` is  
707 a table describing the characteristics and basic information of the receptor  
708 families and, as can be seen in Figure 3, it is often used in join with other  
709 tables in the queries that build a webpage.

710 It is immediately evident that the result of the distribution is the same  
711 with the three strategies. The same effect is also obtained in the other tables  
712 of the database used by the queries shown in Figure 3.

713 Why is that? It is the case that the conditions in which we produced  
714 this experiment are quite peculiar. The queries that we used share similar  
715 characteristics. They are all SPJ queries, each of them utilizes each table  
716 only once in the join condition (there are no self-joins), and all the joins  
717 are made using key attributes. In this particular condition, each tuple of  
718 the output presents: (i) a how-provenance that is a single monomial with  
719 coefficient 1 and exponent 1 in each variable; (ii) a why-provenance that is  
720 composed by only one witness; (iii) a lineage that coincides with the only  
721 witness in the witness basis. It is easy to see how, given these queries, the  
722 three distributions act in the same way. The credit is always uniformly  
723 distributed among the tuples appearing in each provenance.

724 To better clarify what is happening, let us consider one of the types of

---

<sup>12</sup>For reproducibility purposes, the code we used for our experiments and all the pro-  
duced queries can be found at the following link: [https://bitbucket.org/dennis\\_dosso/credit\\_distribution\\_project](https://bitbucket.org/dennis_dosso/credit_distribution_project).

725 queries used to build the output webpage, as shown in Figure 3:

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

730 Q3 returns a series of 10 tuples from the version of GtoPdb we considered.  
 731 The first tuple produced by this query, <Bertil B., Fredholm>, that has  
 732  $c_{939} \cdot c_{2f_{496}}$  as provenance polynomial.  $c_{939}$  represents the provenance token of  
 733 a tuple in **contributor**, the same for  $c_{2f_{496}}$  in table **contributor2family**.  
 734 It is easy to see that the why-provenance of this tuple is  $\{\{c_{939}, c_{2f_{496}}\}\}$  and  
 735 its lineage is  $\{c_{939}, c_{2f_{496}}\}$ . Therefore, the credit assigned to these tuples is  
 736  $1/2$  using all three DS. This actually happens for each tuple of the output of  
 737 each query of GtoPdb, thus making the distributions equivalent with respect  
 738 to their output.

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

## 743 6.2. Synthetic queries

744 To show how the three DS can actually behave differently, let us consider  
 745 another case shown in Figure 8.

746 The figure reports a distribution of credit performed on the table **family**  
 747 through the generation of 10K *synthetic* polynomials. We randomly gen-  
 748 erated provenance polynomials that might be the how-provenance of ran-  
 749 domly generated synthetic queries, using the three GtoPdb tables **family**,  
 750 **contributor2family**, and **contributor**. An example of such synthetic poly-  
 751 nomial is:

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

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

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

755 its lineage is:  
 756

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

757

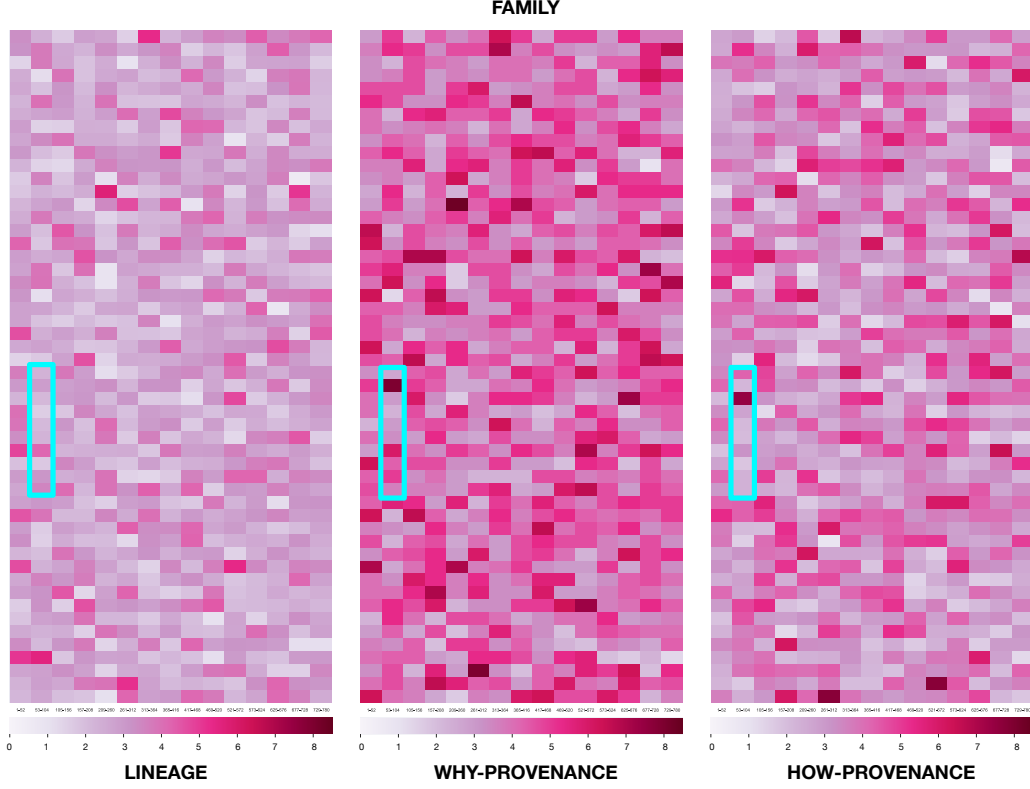


Figure 8: Comparison of three DS on the same table `family` after the distribution computed using 10K synthetic and randomly generated provenance polynomials. The tuples in the blue rectangles are used as example in the discussion connected to Figure 9.

758 These types of polynomials are not impossible to obtain in real life, since  
 759 they can be obtained by nested queries with join and union operations that  
 760 use multiple times the same tuples (thus the presence of exponents bigger  
 761 than 1) and the same combination of operations more than once (thus the  
 762 presence of coefficients for monomials bigger than 1). We randomly generated  
 763 a set of 10K such polynomials.

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

$$f_1 = \frac{59}{315}, f_5 = \frac{1}{18}, c_2 f_1 = \frac{2}{21}, c_2 f_2 = \frac{2}{15}, c_2 f_{17} = \frac{2}{9}, c_1 = \frac{2}{21}, c_2 = \frac{2}{15}, c_{17} = \frac{1}{6}$$

766

767 Using why-provenance, this is the output:

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

768

769 Finally, with lineage, this is the distribution:

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

770

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

776 Generally speaking, the more complex the distribution, the more polar-  
 777 ized the credit is toward the tuples that are used more frequently or with a  
 778 higher impact in the production of the output tuple.

779 Going back to Figure 8, it is thus possible to observe empirically how the  
 780 three provenances behaved differently. We put the maximum value for the  
 781 heat-maps around 8.33, since that is the highest value reached by a tuple  
 782 in all three distributions. We note that lineage is the provenance that gives  
 783 less credit to the tuples of the **family** table. This is due to the fact that the  
 784 DS equally distributes the credit to all the tuples appearing in the lineage.  
 785 Since other two tables are used by these queries, the credit is also given to  
 786 the tuples of those tables.

787 Moving to the heat-map reporting the distribution performed by the DS  
 788 based on why-provenance, we see that this time more credit is given overall  
 789 to the tuples of the table. Actually, this DS is the one that distributes more  
 790 credit to **family** among the three strategies. This is due to the fact that the  
 791 DS based on why-provenance also takes into account the different ways in  
 792 which a tuple is used, e.g. in different joins. If the same tuple is present in  
 793 more than one witness, it is more probable that it will attract more credit,  
 794 “stealing” it from the other tuples in the witness basis. In this case, **family**  
 795 was able to attract more credit, taking it from the other two tables, due to  
 796 the role of its tuples in the queries that were executed.

797 Considering now the heat-map produced by the DS based on how-provenance,  
 798 we see how, although it presents more credit in its tuples than the one present  
 799 in the lineage heat-map, it does not reaches the levels of the why-provenance



Figure 9: Comparison of the distribution of credit performed by the three DSs on a subset of 10 tuples taken from table `family` simulating the passing of time. The number on top of each group of heat-maps represent the number of queries computed.

800 heat-map. This is due to the fact that this DS is even more sophisticated,  
 801 weighting even more than the previous DS the role of tuples in the production  
 802 of the output. The result is a distribution that still rewards the tuples of this  
 803 table more than lineage, but not in the same measure as the DS based on  
 804 why-provenance, since the other tuples in the other tables are able to attract  
 805 more credit due to their roles in the queries.

806 To show how the DS based on different provenances may actually differ  
 807 in their behavior also through the course of time, let us consider Figure 9.

808 In this figure we report four groups of heat-maps. Each group presents  
 809 three maps obtained by selecting the same ten tuples from the GtoPdb  
 810 `family` table after an incremental distribution of credit (the tuples of ranks

811 ranging from 79 to 89). These are the same tuples highlighted in the blue  
812 boxes in Figure 8. In particular, the four groups represents “snapshots”  
813 taken during an incremental accumulation of credit on the database, at dif-  
814 ferent moments chosen when a certain number of executed queries is reached  
815 (specifically, 1K, 2K, 5K and 10K). Figure 8 represents the end of the process.

816 In this way we are simulating the passing of time on a database where  
817 credit distribution is performed. Each group of heat-maps can be thought  
818 as a snapshot of that set of tuples at a certain moment, after a certain  
819 amount of queries are executed. The queries utilized are the same of the  
820 experiment of the previous section. The range of credit in each map goes  
821 from 0 (no credit) to 6 (maximum quantity of credit reached on a tuple at  
822 the “snapshot” reached at 10K queries).

823 Focusing on the 1K and 2K groups, we see that the three DS do not  
824 behave very differently. The tuples highlighted by the three are almost the  
825 same. There are still small differences, in particular in tuple 5.

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

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

847 From this progression we see how, given the peculiar synthetic provenance  
848 polynomials that we presented, it is actually possible to see the differences



849 between the three distribution. These differences become more and more  
850 evident with the passing of time, i.e. the more credit is distributed to the  
851 tuples.

852 Therefore, although all the three DS considered here are effective in dis-  
853 tributing credit, they actually behave differently. The DS based on lineage  
854 is sufficient when a user only wants to highlight the tuples of the database  
855 that are used by a query (and not only visualized in the output). However,  
856 it distributes the credit equally to the tuples of the lineage, therefore losing  
857 information on the role of the tuples in the production of the output.

858 For this, a user may want (depending on the nature of the queries) to  
859 use DS based on why-provenance and how-provenance. The two are more  
860 sensitive than the DS based on lineage (and the DS based on how-provenance  
861 is more sensible than the one based on why-provenance). Therefore, using  
862 these DS, it is possible to change the distribution of credit to the tuple,  
863 rewarding more the tuples that have a more important role in the generation  
864 of the output. These two DS can therefore be preferred when the aim of the  
865 user is to find “hotspots” in the database based on the role of the tuples.

866 In particular, the DS based on why-provenance rewards more the tuples  
867 that are used in different ways by queries (i.e. are members of more wit-  
868 nesses). The DS based on how-provenance also takes into consideration how  
869 many times a tuple is used, adding even more sensibility to the distribution.  
870 One may choose one other the other depending what are the aspects he wants  
871 to highlight in the data.

### 872 6.3. Credit vs Citations

873 For our last set of experiments, we confront traditional citations and  
874 credit, to see if they differ in their behaviour in the case if rewarding the  
875 curators of data. Consider first Figure 10, where we report two radar plots.  
876 The first plot presents the top 20 author (we identify the authors with their  
877 ID instead of their name), ordered based on the normalized value of cita-  
878 tions distributed by the queries taken from the papers published in BJP as  
879 described in Section 6.1, together with their normalized value of credit. An  
880 author transitively receives credit from the data of which he or she is the cu-  
881 rator. That is, the credit assigned to data is then split equally to the authors  
882 of those tuples. As we show in Section 6.1, there is no difference for these  
883 queries in the distribution of credit between the three DS, thus these values  
884 are equal for the three distributions. The second plot is similar to the first  
885 one, but the authors are ordered based on the received credit.

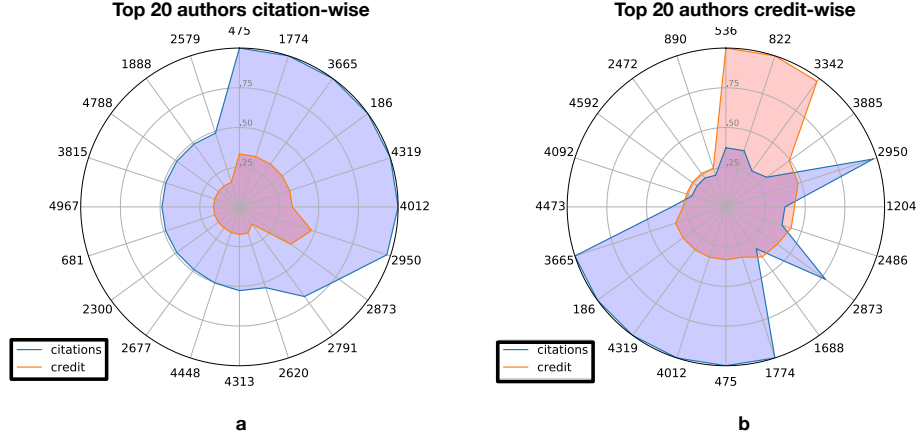


Figure 10: Radars presenting the top 20 authors citation-wise and credit wise, together with their (normalized between 0 and 1) values of citations and credit.

As we see, the quantity of credit does not follow the quantity of citations, i.e. it is not true that an author that presents the highest number of citations also has the highest value of credit. As shown in Figure 10.b, the authors with the highest value of credit do not also have the highest number of citations. This means that there are citations that are much more “valueble” for an author in terms of credit. This is due to the fact that the quantity of credit assigned by these citations is very high, i.e. the impact of those cited data is high.

This shows how credit is able to reward authors that are cited less than others but, nonetheless, have a high impact on the research community.

Consider now Figure 11. In this case, we produced 100, 1K, and 10K synthetic polynomials and we distributed credit and assigned citations through them. Since these polynomials correspond to queries whose authors are not easily identifiable, we created 20 “synthetic” authors, and we randomly assigned one author to each tuple in the database. The authors receive “blocks” of consecutive tuples, with each block of the size varying between 10 and 40. Every time a tuple was used in a provenance polynomial, we assigned one citation to the author corresponding to the tuple. The same author also receives the three different credits assigned to the tuple at the end of the distribution process using the three DS.

The three radar plots in the upper row presents 20 authors are sorted based on the normalized number of received citations, together with the

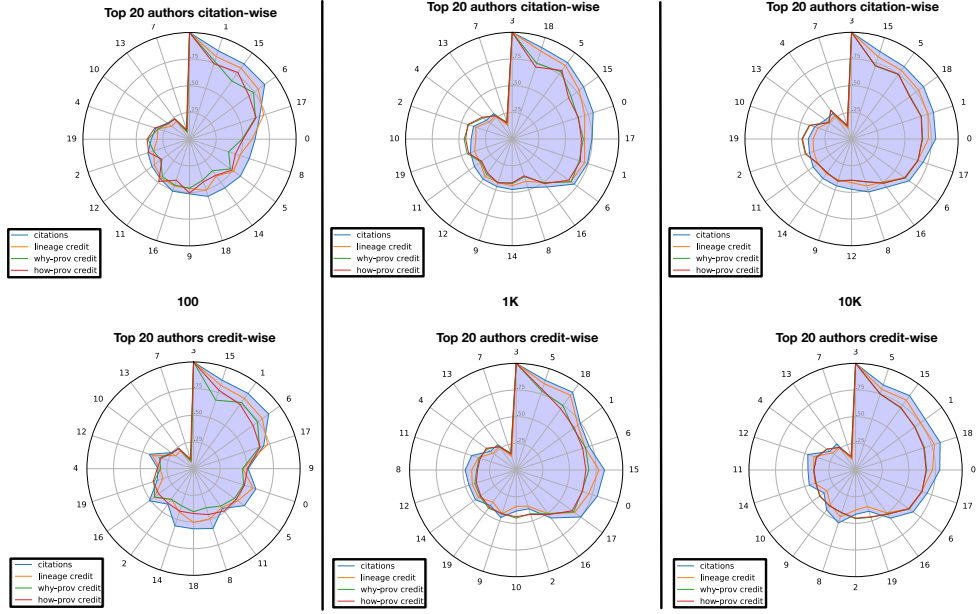


Figure 11: Radars presenting 20 authors ordered citation-wise and credit-wise, together with their (normalized between 0 and 1) values of citations and credit, through the execution of different numbers of polynomials (100, 1K, and 10K). The radar plots ordered by credit consider the credit assigned by the DS based on how-provenance.

corresponding normalized quantities of credits assigned using the 3 different strategies. The ones in the bottom row present the authors ordered based on the quantity of credit assigned by the DS based on how-provenance. In this way we can see the behavior of the different DS in rewarding the authors with the highest levels of citations. Given the synthetic nature of these queries, the correlation between the number of citations and the quantity of credit assigned to the authors appears to be a much stronger with respect to the case with the real-world queries.

Nonetheless, it is possible to note that credit still behaves differently,. In particular we still see that certain authors that are not in the top 10 positions citation-wise are still rewarded with high quantities of credit, showing their importance with respect to their impact. Interestingly, scaling up to 1K and 10K polynomials, it appears that the distribution performed via why-provenance and how-provenance become equivalent for the authors. We can note that, although not exactly equal, the values of credit assigned to the authors by those DS become quite similar with these higher quantities of

924 polynomials.

#### 925 6.4. Execution times

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

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

926 In Table 5 we report the time required to compute the distribution using the DS based on the three provenances as described above. As we see, 927 the execution time grows linearly with the number of polynomials that are 928 submitted to the system. With high values of polynomials (1M), the time 929 required by the DS based on lineage and why-provenance is lower than the 930 time required by the DS based on how-provenance, due to the bigger number 931 of operations the last DS requires during the computations of the portions 932 of credit to assign to the different tuples. We note that, since we created 933 these polynomials on-the-fly, these values do not include the time required 934 to compute the provenances. Therefore, limited to the time required to dis- 935 tribute credit, the three DS are equivalent in terms of performances. The 936 first differences can be seen only with high number of polynomials, when 937 lineage and why-provenance may be preferred if there are no requirements 938 to assign credit with the strategy implemented by the how-provenance-based 939 DS. 940

941 All the experiments were carried on a MacBook Pro 13-inch, 2019 with 942 2.4 GHz processor Intel Core i5 quad-core, 8 GB of memory at 2133 MHz 943 with code written in Java and the support of a PostgreSQL database.

## 944 7. Conclusions

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

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

966 In the second set of experiments, we synthetically produced more complex  
967 provenance polynomials, corresponding to more complex synthetic queries,  
968 that present exponents and coefficients different than 1. In this way, we  
969 showed that, even though all three DS can highlight all the tuples used by  
970 the queries in the database, the three have different behaviors. While the DS  
971 based on lineage rewards all the tuples used by a query in equal measure, the  
972 strategy based on why-provenance tends to reward the tuples more critical  
973 to the query. In particular, why-provenance can consider the different ways  
974 in which one tuple is used in a query. How-provenance is even more sensitive  
975 to the tuples' role: it can also consider the frequency by which a tuple or a  
976 set of tuples is used in the case of more complex queries. Depending on the  
977 goal of a user, one provenance may be preferred to another.

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

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

991 In future work, we plan to explore the different potential applications of  
992 credit on relational databases. One example is the so-called *data pricing*.  
993 Data pricing consists of giving a price to a query submitted by a user who  
994 wants to buy the produced information. Currently, a commonly used strategy  
995 to face data pricing is based on query rewriting. A database stores a set of  
996 views correlated with their price. When a new query arrives, the system tries  
997 to rewrite it using the stored views and obtain a query price. This process  
998 is computationally expensive. We plan to distribute credit through carefully  
999 planned and representative queries and use it as information to define a new,  
1000 faster, and potentially more flexible pricing function.

1001 Another application is *data reduction* [42], concerned with reducing the  
1002 vast mole of data that is produced in the evolving world of research and  
1003 information technology. Data reduction deals with different aspects of dealing  
1004 with huge amounts of data, such as finding reduced and relevant data streams  
1005 from the multiple gigabytes of data produced by big data systems every  
1006 second or dealing with the curse of dimensionality which requires unbounded  
1007 computational resources to uncover actionable knowledge patterns [51].

1008 Data credit can also help to find “hotspots” and “coldspots”. A hotspot  
1009 is data in a database (a tuple or a single attribute, for example) that presents  
1010 a high quantity of credit and is therefore valuable for the set of queries that  
1011 distributed that credit. On the other hand, a coldspot is data that present  
1012 low quantities of credit and can be considered useless or less relevant and can  
1013 therefore be removed or moved in another cheaper and less efficient memory  
1014 location.

## References

- [1] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bernstein, P., Boncz, P., Chaudhuri, S., Cheung, A., Doan, A., Dong, L., Franklin, M. J., Freire, J., Halevy, A., Hellerstein, J. M., Idreos, S., Kossmann, D., Kraska, T., Krishnamurthy, S., Markl, V., Melnik, S., Milo, T., Mohan, C., Neumann, T., Chin Ooi, B., Ozcan, F., Patel, J., Pavlo, A., Popa, R., Ramakrishnan, R., Ré, C., Stonebraker, M., and Suciu, D. (2020). The seattle report on database research. *SIGMOD Rec.*, 48(4):44–53.
- [2] Alawini, A., Davidson, S. B., Hu, W., and Wu, Y. (2017). Automating data citation in citedb. *PVLDB*, 10(12):1881–1884.
- [3] Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., and Wu, Y. (2018). Data citation: A new provenance challenge. *IEEE Data Eng. Bull.*, 41(1):27–38.
- [4] Altman, M., Borgman, C. L., Crosas, M., and Martone, M. (2015). An Introduction to the Joint Principles for Data Citation. *Bulletin of the Association for Information Science and Technology*, 41(3):43–45.
- [5] Baggerly, K. (2010). Disclose all data in publications. *Nature*, 467(7314):401–401.
- [6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D., Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and Goble, C. A. (2013). Why linked data is not enough for scientists. *Future Gener. Comput. Syst.*, 29(2):599–611.
- [7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.
- [8] Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 591–596.
- [9] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Communication*, pages 93–116. De Gruyter Mouton.

- 1047 [10] Buneman, P. (2006). How to cite curated databases and how to make  
1048 them citable. In *18th International Conference on Scientific and Statistical*  
1049 *Database Management, SSDBM*, pages 195–203. IEEE Computer Society.
- 1050 [11] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D.,  
1051 Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn’t  
1052 working, and what to do about it. *Database J. Biol. Databases Curation*,  
1053 2020.
- 1054 [12] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation  
1055 is a computational problem. *Commun. ACM*, 59(9):50–57.
- 1056 [13] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A  
1057 characterization of data provenance. In *Database Theory - ICDT 2001,*  
1058 *8th International Conference*, pages 316–330.
- 1059 [14] Buneman, P. and Silvello, G. (2010). A rule-based citation system for  
1060 structured and evolving datasets. *IEEE Data Eng. Bull.*, 33(3):33–41.
- 1061 [15] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N.,  
1062 Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry,  
1063 R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a.,  
1064 and Wright, D. (2012). Making Data a First Class Scientific Output:  
1065 Data Citation and Publication by NERC’s Environmental Data Centres.  
1066 *International Journal of Digital Curation*, 7(1):107–113.
- 1067 [16] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Jour-  
1068 nals: A Survey. *Journal of the Association for Information Science and*  
1069 *Technology*, 66(9):1747–1762.
- 1070 [17] Cheney, J., Chiticariu, L., and Tan, W. (2009). Provenance in databases:  
1071 Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–  
1072 474.
- 1073 [18] CODATA-ICSTI Task Group on Data Citation Standards and Practices  
1074 (2013). *Out of Cite, Out of Mind: The Current State of Practice, Policy,*  
1075 *and Technology for the Citation of Data*, volume 12.
- 1076 [19] Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., and Simons, N.  
1077 (2019). Bringing citations and usage metrics together to make data count.  
1078 *Data Science Journal*, 18(1).



- 1079 [20] Cronin, B. (1984). *The citation process. The role and significance of*  
1080 *citations in scientific communication*. London: Taylor Graham.
- 1081 [21] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evi-  
1082 dence of a structural shift in scholarly communication practices? *JASIST*,  
1083 52(7):558–569.
- 1084 [22] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of  
1085 view data in a warehousing environment. *ACM Trans. Database Syst.*,  
1086 25(2):179–227.
- 1087 [23] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model  
1088 for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on*  
1089 *Innovative Data Systems Research*. [www.cidrdb.org](http://www.cidrdb.org).
- 1090 [24] Dosso, D. and Silvello, G. (2020). Data credit distribution: A  
1091 new method to estimate databases impact. *Journal of Informetrics*,  
1092 14(4):101080.
- 1093 [25] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The vir-  
1094 tual atomic and molecular data centre (VAMDC) consortium. *Journal of*  
1095 *Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.
- 1096 [26] Fang, H. (2018). A discussion of citations from the perspective of the  
1097 contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–  
1098 1520.
- 1099 [27] Force, M., Robinson, N., Matthews, M., Auld, D., and Boletta, M.  
1100 (2016). Research data in journals and repositories in the web of science:  
1101 Developments and recommendations. *Bulletin of IEEE Technical Com-*  
1102 *mittee on Digital Libraries, Special Issue on Data Citation*, 12(1):27–30.
- 1103 [28] Garfield, E. (1999). Journal impact factor: a brief review. *Can. Med.*  
1104 *Assoc.*, 979-980.
- 1105 [29] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data  
1106 identification and process monitoring for reproducible earth observation  
1107 research. In *2019 15th International Conference on eScience (eScience)*,  
1108 pages 28–38. IEEE.

- 1109 [30] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance  
1110 semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-*  
1111 *SIGART symposium on Principles of database systems*, pages 31–40. ACM.
- 1112 [31] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson,  
1113 A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton,  
1114 S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F.,  
1115 Spedding, M., Davies, J. A., and Nc-Iuphar (2018). The IUPHAR/BPS  
1116 guide to PHARMACOLOGY in 2018: updates and expansion to encom-  
1117 pass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Re-*  
1118 *search*, 46(Database-Issue):D1091–D1106.
- 1119 [32] Hartley, J. (2017). Authors and their citations: a point of view. *Scien-*  
1120 *tometrics*, 110(2):1081–1084.
- 1121 [33] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a  
1122 transformed scientific method.
- 1123 [34] Honor, L. B., Haselgrove, C., Frazier, J. A., and Kennedy, D. N. (2016).  
1124 Data citation in neuroimaging: proposed best practices for data identifi-  
1125 cation and attribution. *Frontiers in neuroinformatics*, 10:34.
- 1126 [35] Joshi-Tope, G., Gillespie, M., Vastrik, I., D’Eustachio, P., Schmidt, E.,  
1127 de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis,  
1128 S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of bio-  
1129 logical pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432.
- 1130 [36] Katz, D. (2014). Transitive credit as a means to address social and  
1131 technological concerns stemming from citation and attribution of digital  
1132 products. *Journal of Open Research Software*, 2(1).
- 1133 [37] Katz, D. S., Hong, N., Clark, T., Fenner, M., and Martone, M. (2020).  
1134 Software and data citation. *Computing in Science & Engineering*, 22 (2):4–  
1135 7.
- 1136 [38] Kosten, J. (2016). A classification of the use of research indicators.  
1137 *Scientometrics*, 108(1):457–464.
- 1138 [39] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S.  
1139 (2011). Citation and Peer Review of Data: Moving Towards Formal Data  
1140 Publication. *International Journal of Digital Curation*, 6(2):4–37.

- 1141 [40] Martone, M. (2014). Joint declaration of data citation principles.  
1142 *FORCE11. San Diego CA. Data Citation Synthesis Group*. [https://www.](https://www.force11.org/datacitationprinciples)  
1143 [force11.org/datacitationprinciples](https://www.force11.org/datacitationprinciples), online September 2020.
- 1144 [41] Meho, L. I. and Yang, K. (2007). Impact of data sources on citation  
1145 counts and rankings of LIS faculty: Web of science versus scopus and  
1146 google scholar. *Journal of the american society for information science*  
1147 *and technology*, 58(13):2105–2125.
- 1148 [42] Milo, T. (2019). Getting rid of data. *Journal of Data and Information*  
1149 *Quality (JDIQ)*, 12(1):1–7.
- 1150 [43] Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D.,  
1151 Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G.,  
1152 Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff,  
1153 D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D.,  
1154 Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson,  
1155 E., McNutt, M., Miguel, M., Paluck, E. L., Simonsohn, U., Soderberg, C.,  
1156 Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers,  
1157 E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research  
1158 culture. *Science*, 348(6242):1422–1425.
- 1159 [44] Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J.  
1160 (2016). Research data explored: An extended analysis of citations and  
1161 altmetrics. *Scientometrics*, 107(2):723–744.
- 1162 [45] Pröll, S. and Rauber, A. (2013). Scalable data citation in dynamic, large  
1163 databases: Model and reference implementation. In *Proceedings of the*  
1164 *2013 IEEE International Conference on Big Data*, pages 307–312. IEEE.
- 1165 [46] Rauber, A., Ari, A., van Uytvanck, D., and Pröll, S. (2016). Identifi-  
1166 cation of Reproducible Subsets for Data Citation, Sharing and Re-Use.  
1167 *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue*  
1168 *on Data Citation*, 12(1):6–15.
- 1169 [47] Rauber, A., Asmi, A., van Uytvanck, D., and Proell, S. (2015). Data  
1170 citation of evolving data: Recommendations of the working group on data  
1171 citation (wgdc). *Result of the RDA Data Citation WG*, 20.
- 1172 [48] Silvello, G. (2018). Theory and practice of data citation. *J. Assoc. Inf.*  
1173 *Sci. Technol.*, 69(1):6–20.

- 1174 [49] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data  
1175 provenance in e-science. *SIGMOD Record*, 34(3):31–36.
- 1176 [50] Spengler, S. (2012). Data Citation and Attribution: A Funder’s Per-  
1177 spective. In of Sciences’ Board on Research Data, N. A. and Information,  
1178 editors, *Report from Developing Data Attribution and Citation Practices*  
1179 *and Standards: An International Symposium and Workshop*, pages 177–  
1180 178. National Academies Press: Washington DC.
- 1181 [51] Ur Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah,  
1182 T. V., and Khan, S. U. (2016). Big data reduction methods: a survey.  
1183 *Data Science and Engineering*, 1(4):265–284.
- 1184 [52] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G.,  
1185 Axton, M., Baak, A., Blomberg, N., Boiten, J., da Silva Santos, L. B.,  
1186 Bourne, P. E., et al. (2016). The fair guiding principles for scientific data  
1187 management and stewardship. *Scientific data*, 3.
- 1188 [53] Wu, Y., Alawini, A., Davidson, S. B., and Silvello, G. (2018). Data  
1189 citation: Giving credit where credit is due. In *Proceedings of the 2018*  
1190 *International Conference on Management of Data, SIGMOD*, pages 99–  
1191 114.
- 1192 [54] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to  
1193 scientific datasets using article citation networks. *Journal of Informetrics*,  
1194 14(2).
- 1195 [55] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of  
1196 new researchers using the zp-index. *Scientometrics*, 106(3):901–916.
- 1197 [56] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for  
1198 Datasets Citation and Extraction Reproducibility in VADMC. *Journal of*  
1199 *Molecular Spectroscopy*, 327:122–137.