AMPLIFYING DATA CURATION EFFORTS TO IMPROVE THE QUALITY OF LIFE SCIENCE DATA

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2018

By
Mariam S. Alqasab
School of Computer Science

Contents

Word Count: 31,654

List of Tables

List of Figures

Abstract

AMPLIFYING DATA CURATION
EFFORTS TO IMPROVE THE QUALITY OF LIFE SCIENCE DATA
Mariam S. Alqasab
A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2018

The massive amount of data received from the biomedical literature raises the issue of maintaining data quality. This leads biomedical database providers to curate their data, whether by using tools or hiring domain experts (humans who are known as curators). It should be noted that the curation process is not affordable for all databases, as it is an expensive and time-consuming task, especially when human experts perform curation. Carrying out curation is crucial in all domains and is not limited to biocuration. In the biomedical field, keeping data curated can prevent harmful problems. For example, if a protein name is miswritten in a data records, a scientist may then use the incorrect name in all their experiments, causing confusion. In short, relying on data that has not received curation can cause the production of incorrect results. The importance of performing data curation leads many researchers to focus their efforts on providing approaches to help speed up the curation process, make it more reliable and make it more efficient.

In this thesis, we aim to amplify the use of the curation efforts in curated sources and put these methods into a format that can be used by others without requiring extra input from data curators. Among all the available suggestions to improve data curation, to the best of our knowledge no model exists to help measure the level of maturity in the curation process. In this thesis, we first propose a maturity model that describes the maturity levels of biomedical data curation. The proposed maturity model aims to help data providers to identify limitations in their current curation methods and enhance their curation process. The maturity model was built based on information gathered from five different biomedical databases and surveying the biocuration literature, and did not require extra input from curators. Second, we explore one possible approach

to maximising the value obtained from human curators (IQBot) by automatically extracting information about data defects and corrections arising from the work that the curators carry out. This information is packaged in a source-independent form, allowing it to be used by the owners of other databases. To extract this information, we compared data from two consecutive versions of the data records. We ran IQBot to monitor a real-world database (UniProtKB) to extract defects and defect corrections. When we compared the extracted defects and defect corrections with data from other databases, we found that the databases still had out-of-date data in their records.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see `http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487`), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see `http://www.manchester.ac.uk/l` and in The University's policy on presentation of Theses.

To those who raised me up
and helped me to be the person who I am today.

To Mum and Dad.

Acknowledgements

I would like to extend my gratitude to my supervisor Dr Suzanne Embury for her productive guidelines and ultimate supports. Suzanne guidance is like the light, which brightens my way through the darkness. I also would like to extend my gratitude to my second supervisor Dr Sandra Sampaio for her instructions, motivations and emotional supports. Thank you both for giving me the reasons to continue this journey, and I am lucky to have the chance to be your student.

Before anyone else, I would like to thank GOD for giving me the strength to be far away from my beloved family and country. Needless to say a BIG THANKS from the bottom of my heart to my parents (Saleh and Noura) to believe in me and being in my side. I also would like to thank my siblings (Fatimah, Aziz, Jasim, and Abody) to be patient with my mood swings. Specially Fatimah as she spent her summer vacation in the last four years to stay with me and support me. Thanks, my family.

Not to forget those who really know what this journey went through, my office colleagues (Ayesha, Iliada, Rene, and Zety) and the Saudi girl's group in the department. We shared our feelings together smiles and tears. Congrats to those who succeeded and good luck for the rest.

Last but not least, I would like to extend my gratitude to the Royal Commission of Jubail to support me financially and give me the opportunity to peruse a higher degree.

Chapter 1

Introduction

Quality is not an act, it is a habit.

Aristotle

As part of managing changes in data, data resource owners may use data curation. The term 'data curation' appears in the literature in relation to raising the standard of data to a more mature level (the term is explained later in this chapter). As we are focused on the biomedical domain, we use the term 'biocuration' to refer to biomedical data curation.

Although using data curation has various benefits for data resources, carrying out data curation is an expensive process. Data curation requires the handling of a lot of resources, including the tools required to perform various curation tasks and the hiring of domain experts to apply their knowledge to the process. Furthermore, data curation is an ongoing and time-consuming process. The problem is that not all data resource owners have the necessary means to afford data curation. In this thesis, we focus on providing ways to maximise the effectiveness of curation efforts in the biomedical domain, and on restructuring the curation process into a format that can be used across a wider range of disciplines.

At the beginning of this chapter, we introduce the term 'data curation' (Section 1.1), and explain how data curation is carried out in the biomedical domain (Section 1.2). This leads us to discuss the problems related to biocuration (Section 1.3). To solve these problems, we define the aims of the thesis (Section 1.3.1) and state the research contributions (Section 1.3.3). The main contributions were published as listed in section 1.4. Finally, we outline the structure of the rest of the thesis (Section 1.5).

1.1 Data Curation

The Cambridge Dictionary defines curation as “the selection and care of objects to be shown in a museum or to form part of a collection of art, an exhibition, etc.” [1]. The term ’curation’ is used more broadly here in the context of data curation, which is our concern in this thesis. According to Shreeves and Cragin, “Data curation is the active and ongoing management of data through its lifecycle of interest and usefulness to scholarship, science, and education, which includes appraisal and selection, representation and organisation of these data for access and use over time” [SC08, p. 93].

Data curation plays an important role in enhancing data quality. Currently, a number of database providers use different ways to improve their data curation, using methods that are automatic, manual, or a blend of both. Automation in curation can solve general quality problems that are easy to detect and solve. For instance, automation can be useful for checking problems related to straightforward data completeness, i.e., querying whether all required attributes have values in all records and are not null. However, the available forms of automated curation cannot fix more complex quality problems. These need human knowledge, and expertise in the domain of the database, to be detected and solved. Not all quality issues come under simple categories like filling in missing values or fixing misspelt values. More complex issues require experts in the domain to examine the data, apply their knowledge to find defects, and correct the data. As a result, some data resource custodians hire human experts to curate their data manually, using their knowledge of the domain. These human experts are known as data curators.

Databases providers use data curation as part of ensuring data quality. Data curation is not limited to fixing defects in the data at one stage, but is an ongoing process. As mentioned above, the curation process can be carried out either automatically (using tools), manually (by curators), or semi-automatically (curators supported by tools). It is better for data resources to avoid only curating data automatically, as not all types of quality issue in data can be identified and corrected using curation tools. Some cases related to inaccuracy need curators to spot the defects in data and assign corrections. In short, carrying out the curation process for a data source is not a simple task, as it requires time, effort, and other resources.

[1] https://dictionary.cambridge.org/dictionary/english/curation

1.2 Data Curation in the Biomedical Domain

Generally speaking, data curation is handled by different communities from various domains. Even within communities, data curation can vary depending on the type of maturity of the data resources. This thesis focuses on data curation in the biomedical domain, which is also referred to as biocuration. Specifically, we aim to amplify the use of curation efforts applied to biomedical databases to help raise data quality. We have chosen the biomedical domain because it is very active in terms of data curation. Many biomedical databases dedicate part of their work to curation. They utilise curation tools and hire curators to work on their data. Since biocuration is an active field, a new society called the International Society for Biocuration[2] was established in 2008. Biocuration, as defined by the society, ”involves the translation and integration of information relevant to biology into a database or resource that enables integration of the scientific literature as well as large data sets” [ISB].

In the biomedical domain, researchers are actively studying issues related to biocuration and working to enhance it. Biomedical communities follow different approaches to data curation; some communities perform curation manually by hiring data curators, while others perform it with the assistance of automatic curation tools. Some biomedical communities, such as Egas [CLN$^+$13] and IMEX [OKA$^+$12] have grouped together to curate data (more details can be found in the 2 chapter).

’Biocurator’ is a term that refers to human experts in the biomedical domain who perform curation. Biocurators may be paid to do this curation or, in some cases, may donate their time to curate data related to their field of interest. Hiring data curators costs money, and this leads data resource owners who cannot afford to employ biocurators to ask domain experts to donate their time. However, in this case, data resource owners cannot guarantee that they will receive suitable offers of help.

1.3 Research problem

There are many advantages of providing data curation, but performing it can be a difficult task. Data curation is high in cost and may require data resource managers to employ curators to do the job. The data need to be curated recursively, and curators need to handle a huge amount of data, which makes it a time-consuming task. A significant amount of research has been done to overcome these issues, including the

[2] www.biocuration.org

following:

- Research proposing different ways to enhance curation whether by providing collaboration between data resources to perform curation such as Phenex [BDD$^+$14] or seeking help by asking authors of papers to participate [BGF$^+$12].

- Dealing with a large amount of literature to curate. Some research has focused on implementing tools to support curation to automate part of the process, such as RLIMS-P [TLL$^+$14] and DataTamer [SBI$^+$13].

The research mentioned above, and most of the literature on biocuration (which is discussed in the next chapter), aims to solve issues related to the curation process. However, to the best of our knowledge, less attention has been paid to wider issues beyond the main idea of improving the curation process or to considering other aspects related to curation. The list below shows some problems that we noticed, which we argue could be resolved by changes to the curation process:

- To lower the cost of biocuration.

- To reduce the time spent on curation.

- To organise the curation process.

- To help data resources owners achieve curation at a level that suits their needs.

1.3.1 Research Aims and objectives

We aim to solve the research problems mentioned above by finding ways to expand the use of current curation efforts. Based on our aim, the thesis objectives are stated as follows:

- To summarise the literature of biocuration.

- To find a way to use the practices of data curation to assess the curation process.

- To find a way to maximise the effectiveness of curation efforts and apply them to a wider range of data sources.

1.3.2 Research Questions

To overcome the research problems mentioned previously, and achieve the research aims and objectives, we formulated the following research questions:

- How can we use the practices of data curation to assess the maturity of the curation process?
- How can we reuse existing curation efforts and package them in a way that is applicable to other resources?
- Can we infer the defects detected in data by a curator by examining the changes made between stable versions of the data resource?
- Can we also infer the corrections identified for the defects by the curators, from changes made between stable versions?

1.3.3 Research Contributions

This thesis presents the following contributions:

Design a maturity model for biocuration. The first contribution of this thesis is a model that focuses on measuring the level of maturity in the curation process, and, at the same time, allows individual data resource managers to define their desired maturity level. The model does not provide a general description of the curation process, but focuses on showing different maturity levels of the basic components of the biocuration process. We amplify the effectiveness of curation efforts by reviewing and gathering information from the literature on biocuration and putting existing curation practises into one model. To the best of our knowledge, the proposed maturity model is the first model that specifically relates to biocuration.

A mechanism for reusing the curation efforts in other databases. We proposed, IQBot, a mechanism that focuses on extracting the curation efforts from a curated data source by following three steps: first, detect defects in data; second, find defect corrections; third, determine the reason behind the defect corrections, subject to the availability of the needed information.

It should be noted that the maturity model represents a way to help biocuration communities assess the maturity of their curation process, and at the same time curation practices that can be used in the curation process.The IQBot is considered as a curation practice which helps efficiency by reusing the curation efforts applied to a curated data source.

1.4 List of Publications

The research described in this thesis has been published in the following papers:

1. Alqasab, M., Embury, S.M. and Sampaio, S., 2017. Amplifying data curation efforts to improve the quality of life science data. Int. J. Data Curation, 12, pp.1-12. (winner of the best paper award)

2. Alqasab, M., Embury, S.M. and Sampaio, S., 2017. A Maturity Model for Biomedical Data Curation. International Conference of Biomedical Ontologies, Newcastle 2017.

1.5 Thesis Structure

The rest of this thesis is structured as follows:

Chapter 2 reviews the current situation for curation for the biomedical domain, as described in the scientific literature, and highlights gaps in the research.

Chapter 3 introduces the concept of the maturity model. The rest of the chapter proposes and describes BIOC-MM, a maturity model for the curation of biomedical databases. We also discusses all the components of the maturity model at each level of maturity, and explain how to use the maturity model in a community.

Chapter 4 presents the results of our effort to evaluate the model, BIOC-MM, with the help of experts in biocuration. First, we survey the literature on evaluating maturity models. We then describe the evaluation process from two different viewpoints: the model builder (off-line); and human experts (expert evaluation).

Chapter 5 introduces our second contribution, which is represented by IQBot, a component that works as a bridge between curated data sources and their consumers.

An IQBot component extracts the results of the curation efforts made on a database (whether the curation is carried out manually or automatically). We then outline the architecture of an IQBot and describe how an IQBot can detect defects and corrections in data. We also show the procedures followed to determine the reason for the detected defect's correction.

Chapter 6 shows how the IQBot described in chapter 5 can be connected to a real-world curated database (UniProtKB). The chapter also discusses the difficulties and challenges involved in dealing with a curated database when handling its data and providing a list of defects and corrections.

Chapter 7 concludes the overall thesis with an explanation of the challenges faced, and outlines possible future work to be carried out.

Chapter 2

Biocuration Literature Review

Necessity... the mother of invention.

Plato

This chapter reviews the existing literature relating to the curation of biomedical databases. The literature review mainly focuses on obtaining an overall idea of what research has been conducted on biocuration, the problems and issues in the domain of biomedical data curation, and the proposed solutions to address these issues.

This chapter is structured as follows. Section 2.1 gives a definition of biocuration and points out some differences between existing curation procedures in biomedical databases. Research on curation workflow, which describes the full curation pipeline and also describes how the curation process operates in respect to literature curation, is presented in Section 2.1.1. Section 2.2 reviews the literature for biocuration, and has been divided to three parts. The first part relates to triage and bio-entity literature (Section2.2.1), which concerns curation procedures used to search for literature and extract data. The second part describes issues relating to the annotation of data expressions, with a focus on the literature related to the identification of relationships in data (Section 2.2.2). The third part, evidence extractions, concerns in the literature on providing proofs of the curation,as described in Section 2.2.3. Finally, the chapter concludes by highlighting the gaps in the biocuration literature.

2.1 Biocuration Definition

The International Society of [1] refers to biocuration as follows: "Biocuration involves the translation and integration of information relevant to biology into a database or resource that enables integration of the scientific literature as well as large datasets; accurate and comprehensive representation of biological knowledge, as well as easy access to this data for working scientists and a basis for computational analysis, are primary goals of biocuration"[2].

The task of biomedical data curation goes beyond fixing defects in data (although this is part of the curator's mission); it covers all the stages of the data life-cycle, such as data collection, data storage, and so on. Besides, some curation tasks must be done by humans who are experts in the domain of the data, and are therefore capable of interpreting the scientific literature, resolving conflicting interpretations, and reflecting the results in the data. The curation task is time-consuming, and it is not always easy to recruit curators with the breadth and depth of expertise to be able to do the job well.

Different database providers carry out the curation process in different ways. For example, FlyBase invites people from outside the database (i.e., the articles' authors) to participate in the curation process. UniProt, on the other hand, divides the curation process into two parts, automatic and manual curation. The Rat Genome Database and PomBase apply OntoMate and Canto, respectively, as dedicated tools.

2.1.1 Biocuration Workflow

When we reviewed the literature relating to biocuration, we found a number of biocuration workflows that outline the steps followed to perform curation. These workflows varied in scope, from those describing full curation pipelines (Section 2.1.1.1) to those with a focus on describing the literature curation pipeline (Section 2.1.1.2).

2.1.1.1 Full Curation Workflow

This section compares the biocuration workflows of two databases: UniProtKB [3] and Rat Genome [4] Databases. UniProtKB uses TrEmbl, where all automatic curated records are stored, and Swiss-Prot, where manually curated data records are stored.

[1] www.biocuration.org

[2] www.biocuration.org

[3] www.uniprot.org

[4] www.rgd.mcw.edu

UniProtKB differentiates data records in TrEmbl from those in Swiss-Prot by referring to them as unreviewed and reviewed, respectively. Figure 2.1 shows the curation workflow of UniProtKB. Data records can receive two types of curation; manual and automatic. The automatic curation concerns issues related to protein sequence, and two rule-based systems are used: Unified Rules (UniRule) and the Statistical Automatic Annotation System (SAAS). The UniRule approach uses rules created by domain experts. However, the SAAS approach is designed based on a decision tree algorithm. In addition to these two systems, InterPro [5] classification is applied. The manual curation focuses on issues that cannot be curated automatically, such as curating literature and family-based curation.

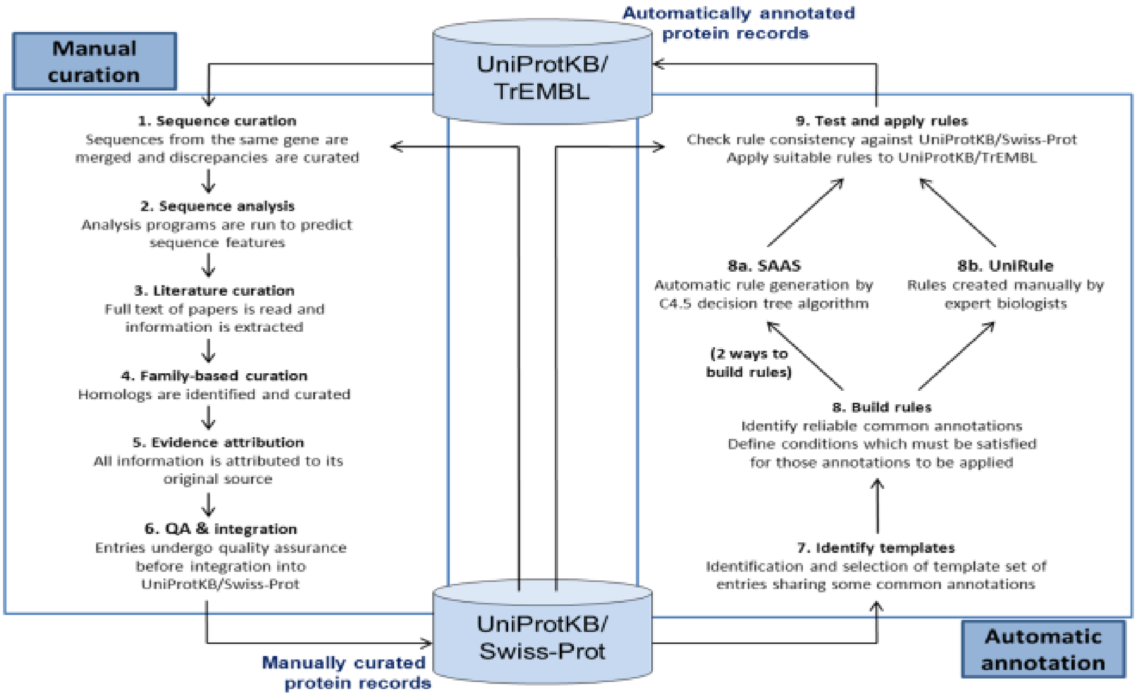

Manual curation
UniProtKB/ TrEMBL
Automatically annotated protein records
1. Sequence curation
Sequences from the same gene are merged and discrepancies are curated
2. Sequence analysis
Analysis programs are run to predict sequence features
3. Literature curation
Full text of papers is read and information is extracted
4. Family-based curation
Homologs are identified and curated
5. Evidence attribution
All information is attributed to its original source
6. QA & integration
Entries undergo quality assurance before integration into UniProtKB/Swiss-Prot
9. Test and apply rules
Check rule consistency against UniProtKB/Swiss-Prot
Apply suitable rules to UniProtKB/TrEMBL
8a. SAAS
Automatic rule generation by C4.5 decision tree algorithm
8b. UniRule
Rules created manually by expert biologists
(2 ways to build rules)
8. Build rules
Identify reliable common annotations
Define conditions which must be satisfied for those annotations to be applied
7. Identify templates
Identification and selection of template set of entries sharing some common annotations
UniProtKB/ Swiss-Prot
Manually curated protein records
Automatic annotation

Figure 2.1: UniProt Curation Workflow[6].

Although many biomedical databases curate their data, they follow different workflows to achieve this. If we compared the UniProt workflow with another database (for example, the Rat Genome database) we can identify differences in the curation workflow or in the procedures performed to curate data. Figure 2.2 shows the main steps for the curation of the Rat Genome Database. Rat Genome focuses their work using a

[5] www.ebi.ac.uk/interpro/

[6] www.uniprot.org/help/biocuration

curation mechanism called OntoMate. OntoMate operates as the main interface where the entire curation process starts. Currently, OntoMate selects and curates abstracts from the literature, but the work of extracting data from full-text sources is ongoing. If this goal is achieved, OntoMate will be the first tool in the domain able to curate full-text articles. It should be noted that UniProtKB separates the data curated manually to the data curated automatically, while Rat Genome does not distinguish between these two methods of curation.

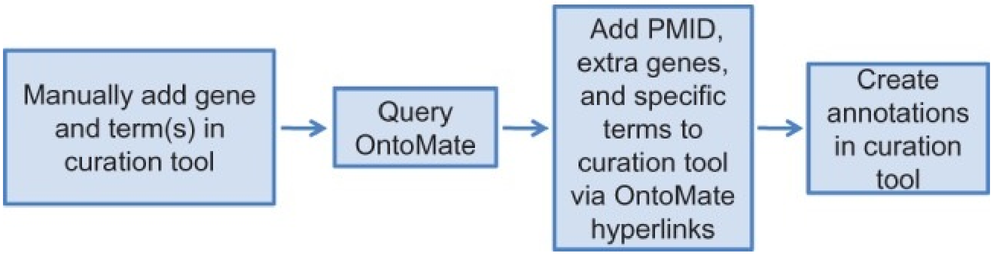

Manually add gene and term(s) in curation tool
Query OntoMate
Add PMID, extra genes, and specific terms to curation tool via OntoMate hyperlinks
Create annotations in curation tool

Figure 2.2: Rat Genome Curation Workflow [LLH+15].

2.1.1.2 Literature Biocuration Workflows

The section describes literature curation workflows that make use of various approaches to speed up the curation process. Hirschman et al. propose the incorporation of embedded text-mining approaches into the biocuration workflow to support the curation process [HBK+12]. The authors start by giving a general overview of the primary procedures of the biocuration workflow, which was constructed based on information gathered in two ways. First, the authors reviewed the methods of data curation used in eight biomedical databases. Second, they looked at the existing text-mining approaches that help to overcome curation problems. The authors list a number of text-mining tools that support some literature curation tasks, and are embedded in the curation workflow when applicable. From these two sources of information, five main stages of literature curation were identified that were common to all the reviewed biomedical databases, as shown in Figure 2.3. These are described as follows:

1. Triage, which has the aim of ranking and prioritising the articles selected from the literature. The articles are selected based on a mechanism of text categorisation.

2. Bio-entity identification and normalisation, which consists of two steps: 1) entity

tagging, which detects specific information in the article selected in the previous step; 2) normalisation, which provides a unique identifier for the entity tags.

3. Finding expressions and relations related to proteins from the articles used in the previous step. The process covers aspects such as extracting protein interactions.

4. Evidential qualifier association, which is responsible for extracting evidence of information accuracy and relationships between data (in other words, providing proof associated with the data curation, such as referring to the experiment that leads to data extracted in step 2 and 3.)

5. Adding the information resulting from the previous steps into the corresponding data records.

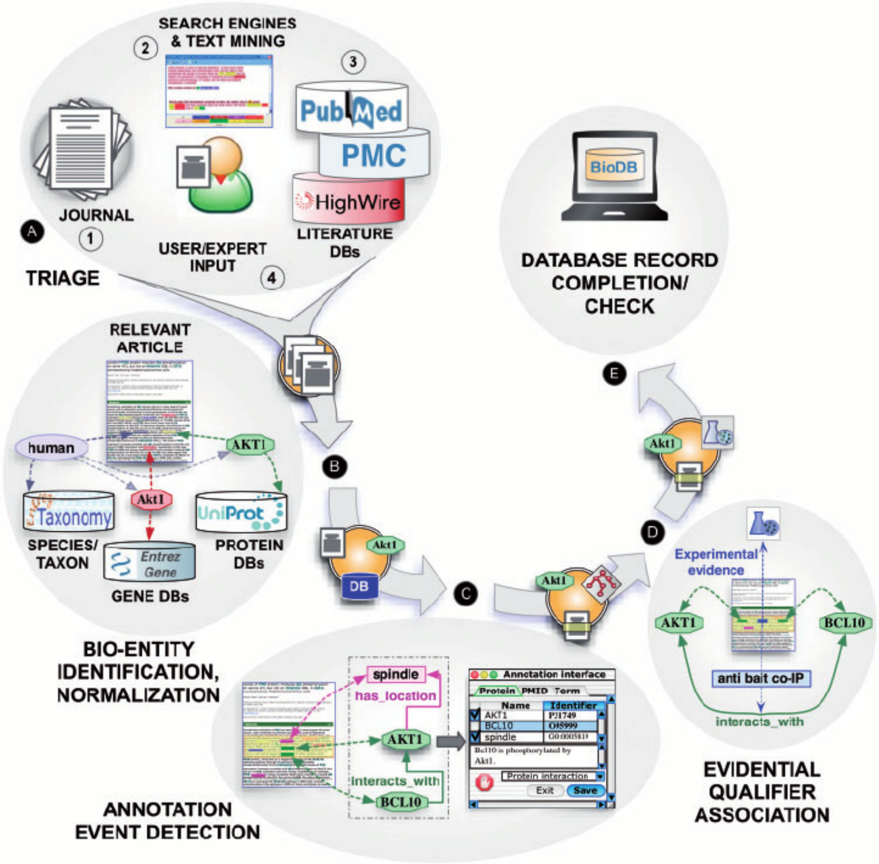


Figure 2.3: General literature curation workflow [HBK+12].

Other researchers focus on describing workflows that serve specific communities and/or integrate text-mining approaches into the curation workflows. McQuilton et al. explain the biocuration workflow of the FlyBase database and discusses some curation problems, before making suggestions for future enhancements, that could be integrated with text-mining tools, in order to overcome them [McQ12]. This curation process is carried out weekly in order to deal with the massive amount of incoming literature in the area. Figure 2.4 shows the steps followed to curate the literature in FlyBase when new articles are selected. The authors of the articles are invited, via email, to fill in information about the articles they have written. The authors then are required to provide information, such as antibody information, in a Fast-Track Your Paper' (FTYP) form. When the authors respond with this information, the curation process will continue in order to curate the full text. However, not all authors reply, and this requires curators to skim the articles to determine if the article is curatable or not before curating the full text. The FlyBase team has started to integrate text-mining tools into the curation workflow. The first tool used is the PaperBrowse tool, which uses natural language processing to extract data such as gene mentions from articles. The second tool organises the articles based on triage through the use of a support vector machine (SVM). The tool accepts data from both authors and curators, and the articles are organised and categorised for full curation based on this data. This tool was build based on collaboration with WormBase[7] and Textpresso[8].

Dowell et al. also discuss the integration of text-mining tools into the biocuration workflow [DMHH^{+}09]. The authors used the biocuration workflow of Mouse Genome Informatics (MGI) as a basis for examining how text-mining tools could fit into the curation process and, in particular, how they could enhance data indexing. Figure 2.5 shows the pipeline of gene indexing in the MGI biocuration workflow. The gene indexing goes through two stages of triage. At the first stage, a collection of articles are selected that are related to specific gene names, such as 'mouse'. Then, the curators are responsible for categorising them. To proceed with the gene indexing, the curation team need to assign the gene symbols to the articles. When the indexing process is completed, the articles can then be curated. The authors' primary focus is to integrate a Name Entity Recognition (NER) tool into the workflow [DMHH^{+}09]. The authors summarise their findings into two systems which meet the needs of MGI: the Open Biomedical Annotator (OBA) from NCBO, and ProMiner from Fraunhofer

[7] www.wormbase.org

[8] www.textpresso.org

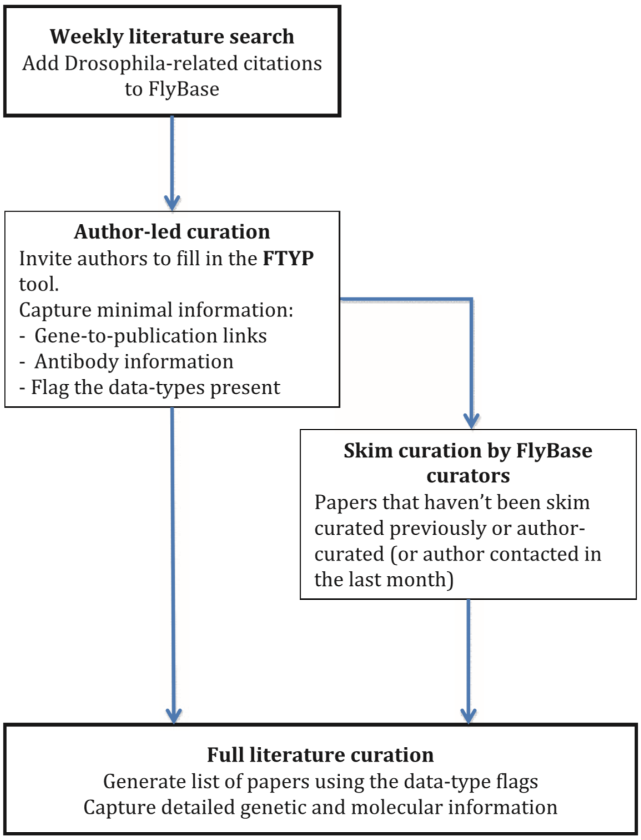

Weekly literature search
Add Drosophila-related citations to FlyBase
Author-led curation
Invite authors to fill in the FTYP tool.
Capture minimal information:
- Gene-to-publication links
- Antibody information
- Flag the data-types present
Skim curation by FlyBase curators
Papers that haven't been skim curated previously or author-curated (or author contacted in the last month)
Full literature curation
Generate list of papers using the data-type flags
Capture detailed genetic and molecular information

Figure 2.4: FlyBase literature curation pipeline [McQ12].

SCAI. Both systems showed good results, but ProMiner showed better results in speeding up the process of data indexing.

In addition to the previous studies, Rak et al. also examine the use of text-mining approaches in the biocuration workflow [RBNR$^+$14]. Argo offers many biocuration features, such as providing a manual annotation editor that allows modification and editing of already-defined annotations. The authors explain the Argo workflow in detail, and show how the workflow could be adopted by other biomedical communities, as shown in Figure 2.6. The workflow covers text-mining approaches to searching for, reading, and annotating data. A Kleio search is used to identify data from articles extracted from PubMed, and textpresso is used to assist in identifying proteins in the articles.

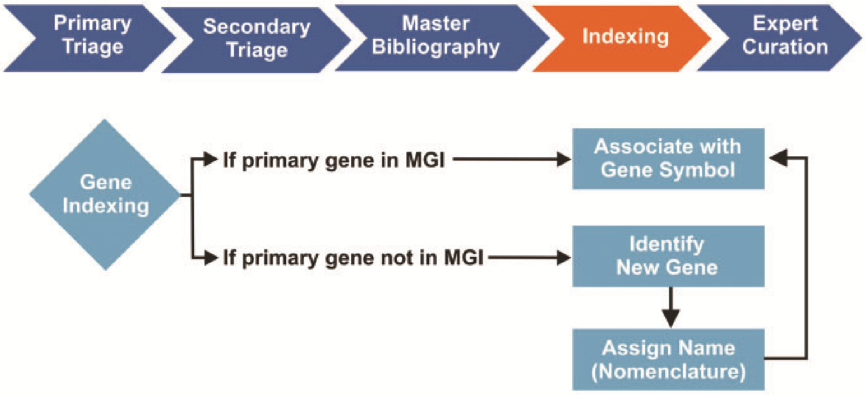
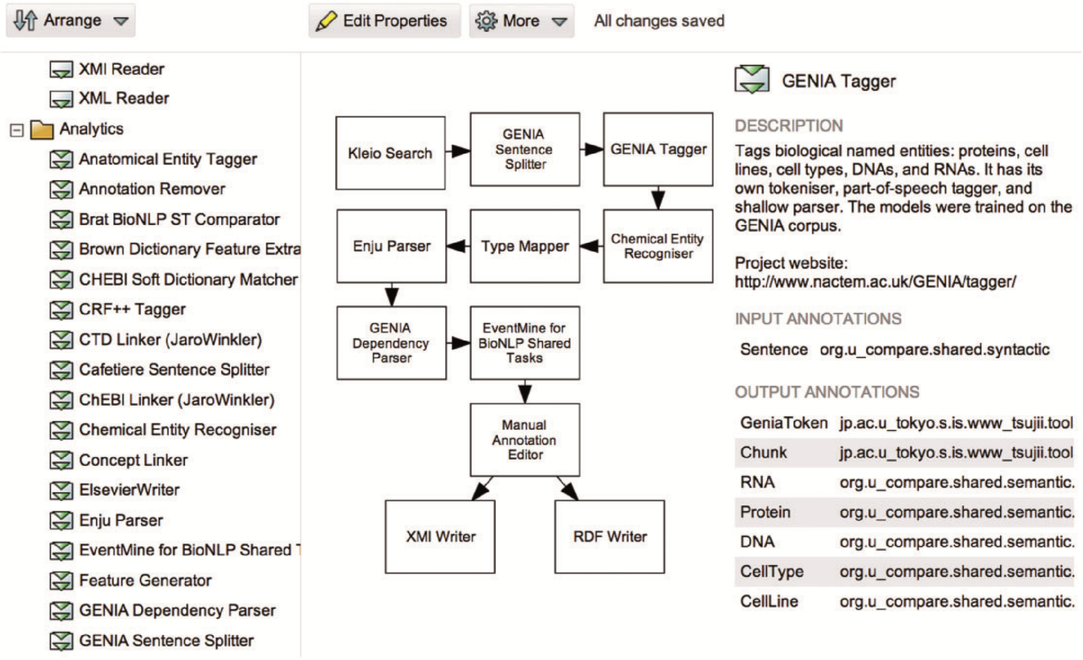


Figure 2.5: Gene Indexing in the MGI Biocuration Workflow [DMHH+09].

Pillai et al. have developed a biocuration workflow called AgBase [PCT+12], designed for annotating agricultural gene products. AgBase aims to raise the efficiency of the curation process by shortening the time needed for curation. It uses text-mining approaches to select and rank the literature; specifically, it uses eGIFT (Extracting Genic Information From Text) to improve the process. The tool was designed by Tudor et al. [TSVS10]. eGIFT is described as “a web-based tool that associates informative terms, called i Terms, and sentences containing them, with genes” [TSVS10].

In contrast to the work described so far, Sernadlela et al. focus on lowering the cost of curation in the biocuration workflow through the use of a semantic-based approach [SO17]. The authors propose a semantic-based architecture that shows the process to curate data.The architecture covered three aspects: knowledge discovery, semantic integration, and semantic services (as shown in Figure 2.7).



DESCRIPTION

Tags biological named entities: proteins, cell lines, cell types, DNAs, and RNAs. It has its own tokeniser, part-of-speech tagger, and shallow parser. The models were trained on the GENIA corpus.

Project website:
http://www.nactem.ac.uk/GENIA/tagger/

INPUT ANNOTATIONS

Sentence	org.u_compare.shared.syntactic

OUTPUT ANNOTATIONS

GeniaToken	jp.ac.u_tokyo.s.is.www_tsujii.tool
Chunk	jp.ac.u_tokyo.s.is.www_tsujii.tool
RNA	org.u_compare.shared.semantic.
Protein	org.u_compare.shared.semantic.
DNA	org.u_compare.shared.semantic.
CellType	org.u_compare.shared.semantic.
CellLine	org.u_compare.shared.semantic.

Figure 2.6: Argo Curation Workflow [RBNR+14].

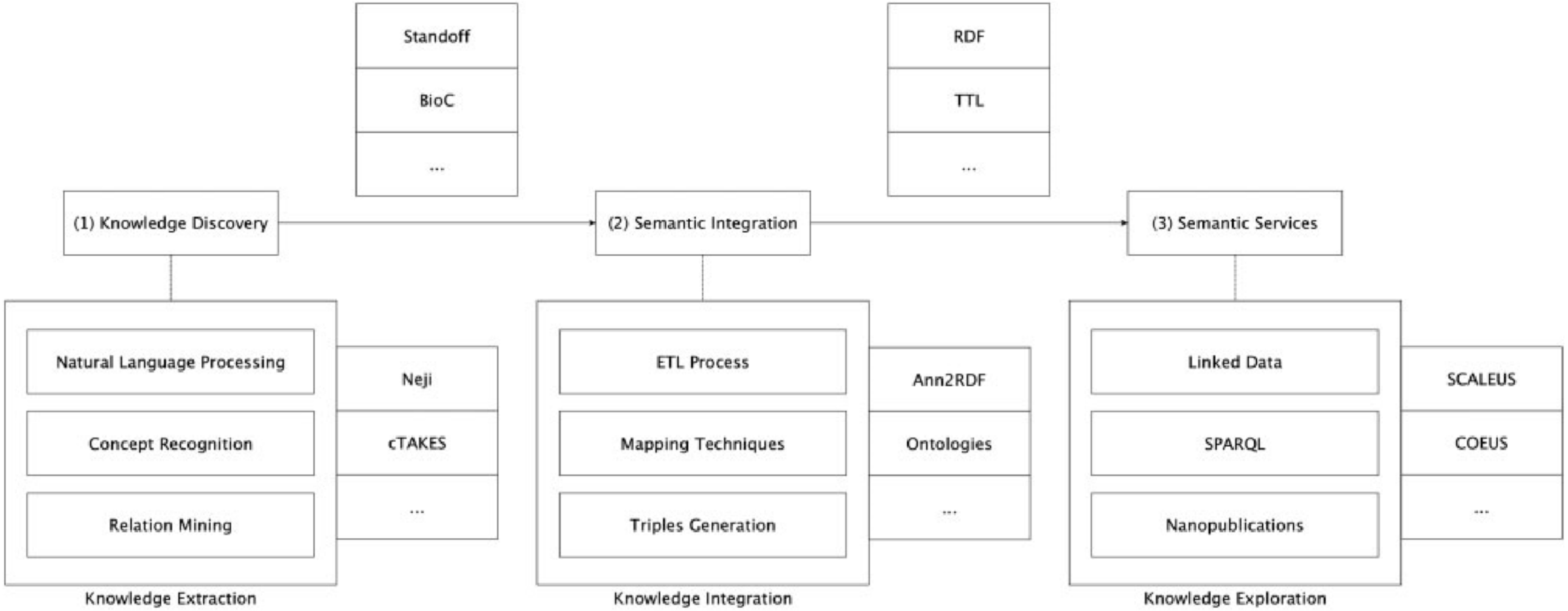


Figure 2.7: The semantic-based architecture [SO17].

2.2 Biocuration Literature

Based on the biocuration workflows mentioned in the previous section, we structured the biocuration literature into the following categories: 1) triage and bio-entity literature, which is concerned with research related to the selection of articles to curate (triage), and extraction of data from the selected articles (Section2.2.1); 2) annotating data expressions, which focuses on extracting protein expressions and interactions (Section2.2.2); and 3) evidence extraction, which focuses on reviewing ways to find evidence associated with the curated data (Section2.2.3).

2.2.1 Triage and Bio-entity Literature

Scientific literature is one of the primary means of communicating new scientific knowledge between scientists. Most of the literature available on biocuration concerns curation using biomedical literature, as this literature is the primary source of curation information. The problem is that the process of determining whether an article is curatable or not is a time-consuming task for biocurators. Besides, identifying the data from the text requires time and effort from biocurators. This leads researchers to focus on research concerned with improving methods for selecting and extracting data from the literature. The section is divided as follows:

1. Collaboration and sharing curation.

2. Specific data extraction.

3. Specialist approaches.

2.2.1.1 Collaborative and Shared Curation

A great deal of data curation is carried out on behalf of communities which share a common interest in maintaining a high quality data resource in a specific domain. This has lead researchers to focus on providing a collaborative and shared environment to curate data, in order to harness the full power of the communities involved.

Orchard et al. aim to find a way to share and reuse curation efforts and ensure removal of redundant data [OKA$^+$12]. The authors designed an approach to serve the International Molecular Exchange (IMEx)[9] consortium. “The IMEx consortium is an international collaboration between major public interaction data providers to

[9] www.imexconsortium.org

share curation effort and make a non-redundant set of protein interactions available" [OKA+12, p. 1]. In essence, the idea is to divide the work of curating the literature between the consortium members. The members are responsible for selecting and curating the journal(s) relevant to their interests. However, to facilitate task sharing (as had been proposed by BioC [CaKK+08], [LCaM+10]), a common data format, in this case PSI-MI XML, is essential. The members receive literature that needs to be curated and share their curation efforts with the rest of the databases through the IMEx platform. This reduces the work needed by curators to locate relevant literature. However, it only targets the participant databases.

Several researchers have explained the concept of collaboration as the base of developing approaches at all curation stages, and not only those related to curation based on the literature. For example, Campos et al. propose Egas, a platform for collaborative curation [CLN+13], [CLMO14]. Egas was first introduced as a means of allowing a team of curators to work on a shared curation project. Later, Egas became a complete collaborative web-based platform, supporting both manual and automatic literature curation and allowing curators to collaborate in real-time. The idea of Egas is to permit a number of curators to work on a curation project. It allows curators to query and retrieve abstracts and full-text articles through its interface. It also allows for adjustments to the permission levels that allow or deny curators access to a specific project. Each project contains annotation guidelines created by the project administrator, and the curators who have joined the project can then start to curate data.

Other teams have created platforms which are targeted at specific biomedical communities. Orchard et al. have initiated a project called MIntAct [OAA+13], which merges the IntAct Molecular Interaction database[10] with the MINT database of verified protein-protein interactions[11]. MINT is manually curated by experts in the scientific literature. The MIntAct project focused on sharing the curation efforts of 11 different databases in order to gain maximum value from the curation work performed at each source.

McQulton et al. also use the idea of sharing curated data, by proposing BioSharing [MGBRS+16]. BioSharing is based on combining three registries, with all registries following the same standard data format. BioSharing, as with other shareable approaches, provides an interface that allows both users and consumers to browse and share data. In addition, BioSharing enables users from the community to participate in

[10] www.ebi.ac.uk/intact

[11] mint.bio.uniroma2.it

curating data records and report any issues.

Lee et al. propose Web Apollo, a genomic annotation editing platform [LHR$^+$13]. Web Apollo works with the help of JBrowse for visualisation. Web Apollo is based on the idea of collaborative curation, in which curators in different geographical locations collaborate to curate data. The data feeds into Web Apollo from various sources and files.

The concept of collaborative curation covers more than just a framework for curators. Collaboration can be used as a solution to other curation problems, such as the need to speed up curation. One such approach is to allow the authors of articles to participate in the curation process. Bunt et al. propose automation of the process of contacting authors of articles undergoing curation to invite them to participate in the curation process [BGF$^+$12]. This method is applied by the FlyBase database; if a new article is selected for curation, an e-mail is sent to the author, which directs them to fill in a form with some information about the article. Based on the information supplied by the authors, articles are prioritised and ranked to prepare them for curation. This helps curators not only to speed up the curation process but also to start with the more important articles. Dai et al. also use the idea of author participation [DTW$^+$13]. However, the aim here was not to lower cost or reduce workload, but to utilise the knowledge of the community and encourage authors to participate in the curation process. Dai et al. have developed a tool called Author Reward to meet this aim, which is built as an extension to multiple biological wikis. Karp et al. propose the use of crowd-sourcing and author participation to curate data [Kar16]. Karp et al. discuss the findings of both approaches, crowd-sourcing and author participation, by outlining the number of attempts to use or adopt crowd-sourcing curation by multiple groups. The authors conclude that the results of using a crowd-sourcing approach do not show sufficient author responses to make a meaningful difference to curation workflow. However, if the author participation approach is followed, it could lower the cost of curation by reducing the time taken to carry it out.

Some authors have proposed general curation platforms oriented around the concepts of collaboration and sharing. An example is the work of Abrams et al., who have proposed DataShare, which was developed at the University of California as a platform for researchers to enhance data curation [ACS$^+$14]. DataShare helps curators to: “(1) prepare for curation by reviewing best practice recommendations for the acquisition or creation of digital research data; (2) select datasets using intuitive file browsing and drag-and-drop interfaces; (3) describe their data for enhanced discoverability in terms of the DataCite metadata schema; (4) preserve their data by uploading to a public access collection in the UC3 Merritt curation repository; (5) cite their data in terms of persistent and globally-resolvable DOI identifiers; (6) expose their data through registration with well-known abstracting and indexing services and major internet search engines; (7) control the dissemination of their data through enforceable data use agreements; and (8) discover and retrieve datasets of interest through a faceted search and browse environment" [ACS+14, p. 1]. DataShare also provides an interface to submit and download datasets.

Another such platform is the DataStaR (Data Staging Repository) platform [KCCR+11]. The authors, Khan et al., follow other researchers in using the idea of sharing to enhance data curation. However, unlike the other proposals discussed above, DataStaR adds more features by providing metadata for the insertion of additional information. The authors applied DataStaR to the biomedical domain, but they claim it could be adapted for use in other domains. As with the other platforms, DataStaR provides an interface for users to store and publish data, and uses an XML format to present the data.

Other research focuses on sharing biomedical data curation. Comeau et al. adopted a markup language format for the BioC system that aims to share data curation [CIDC+13]. The implementation of BioC uses an XML format as it is one of the most common formats. The key factor of using XML format is to ease the job of the interaction with data.

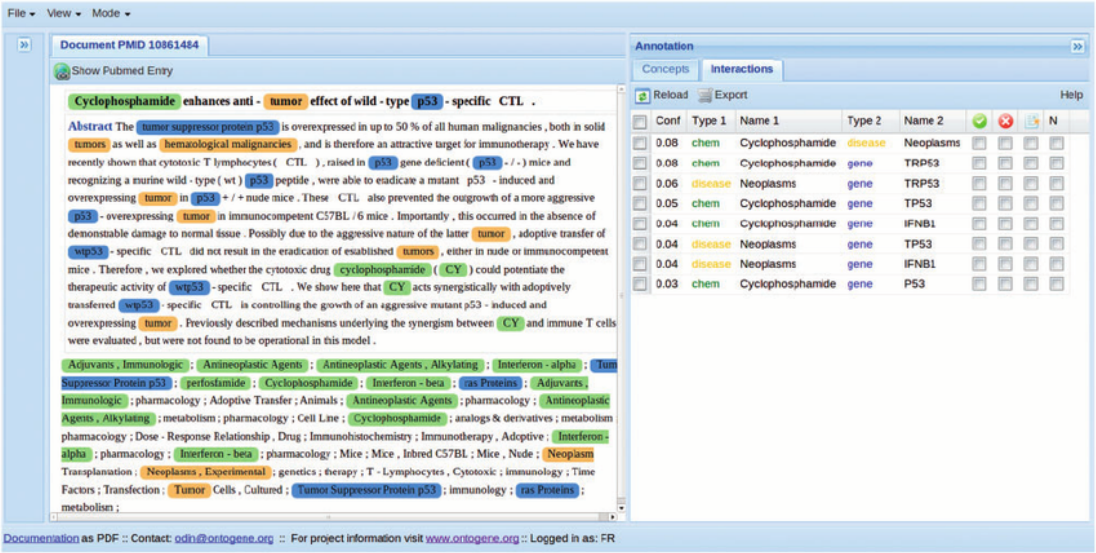
2.2.1.2 Specific data extraction

Some researchers have focused on proposing approaches that support extracting specific data, such as extracting a particular gene name or protein name. Torii et al. implement a rule-based tool, called RLIMS-P, to extract protein phosphorylation data from the literature [TLL+14]. "Protein phosphorylation is central to the regulation of most aspects of cell function" [TLL+14, p. 1]. The primary role of RLIMS-P is to automatically find, select, and extract data from articles related to protein phosphorylation. The tool offers a website to access its services, and allows users to search for articles using PubMed ID or keywords. The tool can search and sort the retrieved articles based on assigned criteria. The abstracts of the resulting list of articles are then highlighted in terms of the data related to protein phosphorylation. Links are provided to the corresponding data in external data resources, such as UniProt.

Singhal et al. have investigated a way to extract disease-gene-variant triplets from the literature [SSL16]. However, since prior to their work there had been few attempts to extract triplets from the biomedical literature. Singhal et al. focused on raising the accuracy of the extracted triplets by looking not only in the selected articles but also in all other relevant articles in PubMed.

Wu et al. suggest an algorithm to make the curation process faster by extracting information in Parkinsons and Alzheimers disease. To achieve this goal, an automated approach for highlighting information in PDF documents is proposed [WOG+17]. The approach is based on using linguistic and semantic features to detect the relevant data in the PDF documents. To identify the relevant names in the PDF (i.e., information related to Parkinsons and Alzheimers disease), name entity recognition approaches are used. These include the approach used by the National Center for Biomedical Ontology (NCBO) and the named entity model from the Natural Language Toolkit (NLTK). After extracting the data, it is sorted based on spatial boosting.

Rinaldi et al. have implemented ODIN, a tool that extracts customised data and relationships from the literature [RDS+13]. ODIN, as with several other tools, uses PubMed to retrieve articles. ODIN reviews the annotated text, which is presented as highlighted text or a table, as shown in Figure 2.8. In addition, the users can reorder the results based on their preference. The tool runs within a web browser.



Cyclophosphamide enhances anti - tumor effect of wild - type p53 - specific CTL .

Abstract The tumor suppressor protein p53 is overexpressed in up to 50 % of all human malignancies , both in solid tumors as well as hematological malignancies , and is therefore an attractive target for immunotherapy . We have recently shown that cytotoxic T lymphocytes (CTL) , raised in p53 gene deficient (p53 - / -) mice and recognizing a murine wild - type (wt) p53 peptide , were able to eradicate a mutant p53 - induced and overexpressing tumor in p53 + / + nude mice . These CTL also prevented the outgrowth of a more aggressive p53 - overexpressing tumor in immunocompetent C57BL / 6 mice . Importantly , this occurred in the absence of demonstrable damage to normal tissue . Possibly due to the aggressive nature of the latter tumor , adoptive transfer of wtp53 - specific CTL did not result in the eradication of established tumors , either in nude or immunocompetent mice . Therefore , we explored whether the cytotoxic drug cyclophosphamide (CY) could potentiate the therapeutic activity of wtp53 - specific CTL . We show here that CY acts synergistically with adoptively transferred wtp53 - specific CTL in controlling the growth of an aggressive mutant p53 - induced and overexpressing tumor . Previously described mechanisms underlying the synergism between CY and immune T cells were evaluated , but were not found to be operational in this model .

Adjuvants , Immunologic ; Antineoplastic Agents ; Antineoplastic Agents , Alkylating ; Interferon - alpha ; Tumor Suppressor Protein p53 ; perfosfamide ; Cyclophosphamide ; Interferon - beta ; ras Proteins ; Adjuvants , Immunologic ; pharmacology ; Adoptive Transfer ; Animals ; Antineoplastic Agents ; pharmacology ; Antineoplastic Agents , Alkylating ; metabolism ; pharmacology ; Cell Line ; Cyclophosphamide ; analogs & derivatives ; metabolism ; pharmacology ; Dose - Response Relationship , Drug ; Immunohistochemistry ; Immunotherapy , Adoptive ; Interferon - alpha ; pharmacology ; Interferon - beta ; pharmacology ; Mice ; Mice , Inbred C57BL ; Mice , Nude ; Neoplasm Transplantation ; Neoplasms , Experimental ; genetics ; therapy ; T - Lymphocytes , Cytotoxic ; immunology ; Time Factors ; Transfection ; Tumor Cells , Cultured ; Tumor Suppressor Protein p53 ; immunology ; ras Proteins ; metabolism ;

□	Conf	Type 1	Name 1	Type 2	Name 2	✓	✗		N
□	0.08	chem	Cyclophosphamide	disease	Neoplasms	□	□	□	□
□	0.08	chem	Cyclophosphamide	gene	TRP53	□	□	□	□
□	0.06	disease	Neoplasms	gene	TRP53	□	□	□	□
□	0.05	chem	Cyclophosphamide	gene	TP53	□	□	□	□
□	0.04	chem	Cyclophosphamide	gene	IFNB1	□	□	□	□
□	0.04	disease	Neoplasms	gene	TP53	□	□	□	□
□	0.04	disease	Neoplasms	gene	IFNB1	□	□	□	□
□	0.03	chem	Cyclophosphamide	gene	P53	□	□	□	□

Documentation as PDF :: Contact: odin@ontogene.org :: For project information visit www.ontogene.org :: Logged in as: FR

Figure 2.8: ODIN interface [RDS+13].

OntoMate is another tool designed to assist biocuration for the Rat Genome Database[12] [LLH+15]. OntoMate supports curators in extracting and annotating PubMed articles with a concept-based approach that uses natural language processing. The tool covers all the stages of the curation process, from collecting data for curation to retrieving the curation results.

Yepes et al. have proposed an approach to data curation based on the use of supplementary materials [YV13]. This paper is an extension of the prior work of the authors, adding the features of access to the provided tables and supplementary materials, and of extracting data for curation. The method followed here is to obtain information from tables and supplementary materials in articles and store it in an XML document. This approach converts any data received to text. The reason for using the data in a text file format is because the work is based around the EMU tool for extracting data, proposed by Doughty et al. [DKFB+10], which can only accept text files as input. This creates issues if the materials contain images, as they cannot currently be processed via this method approach.

Other groups have aimed to design approaches to extracting information related to specific genes or proteins. For example, Dai et al. have proposed LiverCancer to assist in annotating data related to liver cancer disease [DWL+14]. The relevant data in the articles are highlighted and extracted using a combination of text-mining tools.

Verspoor et al. proposed a schema for annotating the biomedical literature on the human variome and for identifying the relationship of the human variome to diseases [VJYC+13]. The schema includes each article's abstract as well as the full text. The articles are extracted from PubMed, and the data are annotated with the BRAT annotation tool [13]. The schema deals with entities, which are data related to the human variome, and relationships, which is information that indicates the relationships between entities. The schema covers eleven aspects of the entity (such as gene, mutation, and age) and thirteen types of relationships. Guidelines for using the schema are provided. The schema may provide a clearer understanding of the entities and relationships to be used in implementing text-mining approaches for curation.

Jamieson et al. propose a semi-automatic text-mining approach to curate *pain* relevant data, which is based on combining text-mining tools [JRR+13]. The approach has three main stages: 1) document scoring, in which the articles are sorted based on the presence of specific terms in the title, abstract or text; 2) data extraction, in which

[12] WWW.rgd.mcw.edu

[13] WWW.brat.nlplab.org/

a text-mining tool is used to identify information about molecular interactions; and 3) data visualisation, in which results are shown through the use of the MediaWiki framework. The primary focus is to support curation of molecular interaction data. This means that this approach could be applied to different biomedical fields looking at molecular interactions, and is not only relevant to research into pain mechanisms.

2.2.1.3 Special purpose approaches

A few research teams have focused on tools to generate research that conforms to a specific type of data quality, such as accuracy or consistency. Keseler et al. propose an approach to assess the accuracy of curation annotations, and to validate them by checking that the information mentioned when curating data is available in the referenced publication [KSW^{+}14]. The method was applied to the EcoCyc and CGD databases. The process of validation was carried out manually, as a validator was need to check the accuracy of data by going through the articles mentioned in the data records. However, this approach is time-consuming and costly. In contrast, Balhoff et al. apply notions of collaboration and sharing to ensure data consistency and reduce errors when curating data [BDD^{+}14].

Some researchers have tried to increase the accuracy and efficiency of the selection of articles for curation. Zarva et al. propose an approach to improve the ranking of evidence available in biomedical literature [ZBNDA17]. The approach combines two other approaches: rule induction, which uses rule-based to extract dependency relationships between data; and the Random Forest classifier proposed by Liaw and Wiener [LW^{+}02]. This hybrid approach provides a score for measuring the uncertainty for the extracted evidence.

Corst et al. have aimed to produce curated data with fewer errors by developing a graph-based approach to finding problems in data records [CRR15]. This approach focused on the changes in data when integrating it with other resources.

Others researchers have aimed to make use of existing approaches to improve the overall quality of curated data. In trials, PubTator showed an ability to improve the speed and efficiency of the curation process and smoothed the work of curators [WKL13]. However, PubTator developers found some problems that PubTator is not capable of solving. Firstly, PubTator can only extract data from the title and abstract, not the full text of a paper. Secondly, it can currently only be applied in the biomedical field. Thirdly, the pre-annotation process can only be performed for a limited number of concepts (names). Despite these issues, PubTator and other similar tools mean that rather than spending time finding out which articles need to be curated, curators can have more time to spend on actual curation.

Kare et al. use a crowd-sourcing technique to scale curation of drug indication data [KBA+15]. This approach uses the Amazon Mechanical Turk (MTurk) platform to curate data. It also provides a guideline document with the user interface. The authors claim that using a crowd-sourcing technique helps in lowering the time and cost of curation.

Poux et al. have sought to solve the problem of coping with the rapidly increasing number of new articles that need to be curated [PAM+17]. The authors suggest using a multiple literature triage approach with the help of the PubTator tool with UniProt. With the help of PubTator to annotate articles, some changes can be made to meet the requirements of UniProt, such as changing the record identifiers to UniProt accession numbers. Also, PubTator indicates to curators whether the article needs to be manually curated or not, and divides the resultant articles into curateable, not priority, and not curateable. As this takes place at the outset, it helps curators to be more selective as to which articles they devote curation time to, which raises the overall quality of the curated data. A further example of researchers tackling this problem of the volume of new articles to be curated is found in the work of Wei et al., who evaluate the performance of PubTator in curating genes in PubMed abstracts [WHL+12]. The evaluation was carried out by looking at the precision, recall, and f-measure metrics. Based on these three metrics, the results obtained using PubTator were compared. The authors found that the results of using PubTator showed a 40% improvement in curating data and, more importantly, using PubTator did not affect quality.

Rutherford et al. propose an online literature curation tool, Canto, which is dedicated to curating the fission yeast database and PomBase [RHL+14]. As with other tools discussed above, Canto accesses PubMed articles. Rinaldi et al. used text-mining techniques within the curation pipeline to speed the curation process, as there are a large number of incoming articles that need to be curated and the workload cannot be completed by available human resources [RLGC+17]. The goal here is to “find information related to curation in adaptive interface and use sentence similarity technique to create interlinks across articles [RLGC+17].

Neves et al. made use of existing tools to enhance data curation for the SABIO-RK[14] database by implementing BLAHmun [RW16]. The main role of BLAHmun is to search for data and annotate it. It does this by integrating the Medicate and TextAI

[14] WWW.SABIO-RK database

tools to perform data curation by searching for and curating data.

2.2.2 Annotating Data Expressions Literature

As part of the curation process, the relationships between data and the protein/gene expressions need to be extracted, whether manually by curators or automatically using tools. In this section, we review the literature surrounding approaches to extracting relationships (Section 2.2.2.1) and approaches to extracting protein interactions (Section 2.2.2.2).

2.2.2.1 Extracting Relationships

Several systems extract biological expressions from the literature, such as BELMiner [RRML17]. By biological expression, we mean the relationship between data extracted and evidence of these data in the biomedical text. BELMiner is an example of a system that extracts biological expressions from the biomedical literature using a rule-based semantic parser [RRML17]. The semantic parser extracts data through the use of the Biological Expression Language[15] (BEL). The BEL "is a language for representing scientific findings in the life sciences in a computable form" [16]. The authors discuss how to develop such a framework, and examine its efficiency in curating data. However, the authors mention that the results of adopting a rule-based approach were not promising; performance was poor in extracting BEL statements. Furthermore, the rules used to extract data from the text do not cover all aspects of curation.

Neves et al. propose CellFinder, a curation pipeline designed specifically for extracting gene expressions related to the cell and anatomical parts of the kidney [NDM+13]. CellFinder supports a six-stage process: 1) triage, which involves querying MedlineRanker [FBSS+09] for articles; 2) pre-processing, which uses the OpenNLP toolkit [17] to split the text based on sentences; 3) named-entity recognition, in which named-entity information is extracted based on ontology approach; 4) post-processing, which focuses on acronym resolution, ontology mapping, and blacklist filtering; 5) event extraction, which finds the relationships between named-entity information; and 6) manual validation, which uses "Bionotate [CMB+09], a collaborative open-source text annotation tool" [NDM+13, p. 6].

[15] WWW.openbel.org

[16] WWW.openbel.org

[17] WWW.opennlp.apache.org

2.2.2.2 Extracting Protein Interactions

PIMiner is a web-based tool for extracting protein interaction data from the biomedical literature [CZT+13]. PIMiner is designed to retrieve abstracts from PubMed and to search for the required information in the abstract.

Rinaldi et al. is concerned with detecting protein-protein interaction in the literature, but differs in dealing with large-scale literature [RFC15]. The approach focuses on providing three features: extraction of protein interaction information; sorting the results; and providing a GUI to allow users to curate data.

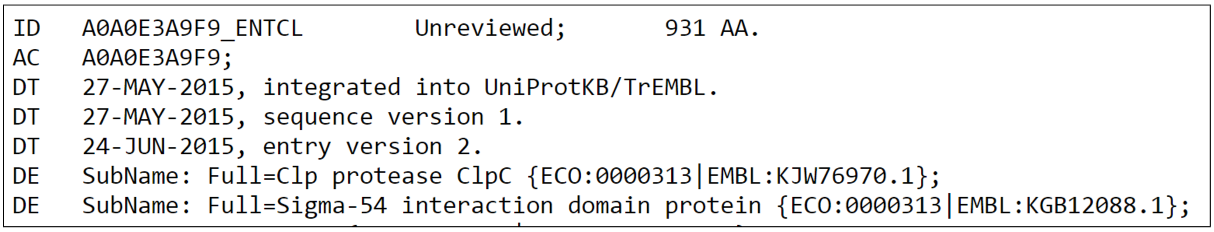
The BioQRator tool is also designed to extract protein-protein interaction (PPI) data [KKS+14]. BioQRator serves a general annotation purpose, providing several features for adding, finding, sharing, and downloading documents. In addition to these features, BioQRator is integrated with text-mining resources to order the articles and to extract entity/relationships using a triage module and entity/relationships module, respectively. BioQRator is designed to accept both PubMed and BioC formats.

Dogan et al. have proposed a method to annotate protein-protein and genetic interaction records from the biomedical literature [IDKCa+17]. The method consists of two iterations. Four curators are responsible for annotating different articles in each iteration. The curators manually extract protein-protein interactions (PPIs) and genetic interactions (GIs) data, and use a tool that helps keep a record of the extracted data. The data is presented as passages that contain the extracted data, evidence, keywords, and other information such as gene/protein and organisms/species. Then, the results from the iterations are compared and assessed. Mottin et al. also focus on protein interaction [MPG+17], examining PPIs and post-translational modifications (PTMs). Mottin et al. not only focus on extracting this kind of information but also proposed an approach to sort and order the articles containing related information for PPIs and PTMs. The approach uses triage to rank articles relevant to the curator's needs. The articles are retrieved from MEDLINE through BioMed. Generally, the articles are ranked based on two metrics; the vector-space search engine, and density.

2.2.3 Evidence Extraction

In the early stages of biocuration for a data resource, the curation team focuses on the process of curating data and meeting the immediate needs of its community. Further on, however, the need to consider other aspects to enhance data curation may start to appear, and curators may began to use ontology terms when annotating data. Using ontology terms helps in many ways, as they can be used to refer to a data source where the data has been extracted, or to provide metadata about the data presented in the record. For example, using the term ECO:0000250, from Evidence and Conclusion Ontology (ECO), means that the data have received curation based on curator observation of experimental evidence. There are many ontologies developed to serve different biomedical concepts, such as GO[18], ECO[19], Sequence Ontology (SO)[20], and Ontology for Biomedical Investigations (OBI)[21].

Currently, many biomedical databases use terms from different ontologies. UniProt, for example, uses the ECO and Gene Ontology (GO) terms while annotating data. Curators provide the ontology terms to the annotated data in the record. Figure 2.9 shows an example of a protein entry, with accession number A0A0E3A9F9, in UniProt. In the entry, an ECO code appears next to the protein SubName. The ECO term gives the reason why the data has been changed in the data record. If we look at the translation of the term ECO:0000313—EMBL:KJW76970.1, the first part, ECO:0000313, indicates that the data has been taken from another biomedical source (the one stated in the second part of the annotation). This is given as EMBL:KJW76970.1, which refers to the EMBL database. Figure 2.10 shows an example of another protein entry in UniProt, with the accession number P0C7Y4, which contains the GO term. If we look at the entry, we can see the GO term is GO:0019835. The definition of the term is "the rupture of cell membranes and the loss of cytoplasm"[22].



```
ID   A0A0E3A9F9_ENTCL        Unreviewed;       931 AA.
AC   A0A0E3A9F9;
DT   27-MAY-2015, integrated into UniProtKB/TrEMBL.
DT   27-MAY-2015, sequence version 1.
DT   24-JUN-2015, entry version 2.
DE   SubName: Full=Clp protease ClpC {ECO:0000313|EMBL:KJW76970.1};
DE   SubName: Full=Sigma-54 interaction domain protein {ECO:0000313|EMBL:KGB12088.1};
```

Figure 2.9: UniProt Entry, A0A0E3A9F9, which contains ECO term

Much research has focused on findings ways to ease and speed up the selection of ontology terms. Rutherford et al. have implemented the Canto tool which suggests GO terms the curator might wish to use while they are annotating data [RHL^{+}14]. A

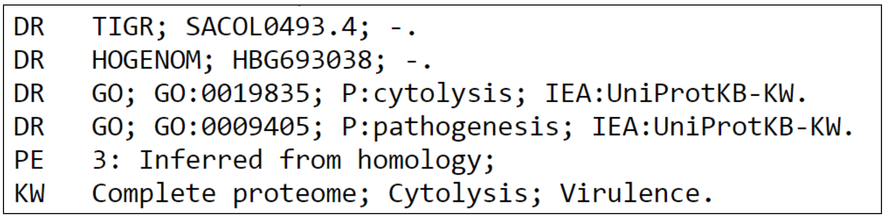
[18] www.geneontology.org

[19] www.evidenceontology.org

[20] www.sequenceontology.org

[21] http://obi-ontology.org

[22] www.amigo.geneontology.org/amigo/term/GO:0019835



```
DR   TIGR; SACOL0493.4; -.
DR   HOGENOM; HBG693038; -.
DR   GO; GO:0019835; P:cytolysis; IEA:UniProtKB-KW.
DR   GO; GO:0009405; P:pathogenesis; IEA:UniProtKB-KW.
PE   3: Inferred from homology;
KW   Complete proteome; Cytolysis; Virulence.
```

Figure 2.10: UniProt Entry, P0C7Y4, which contains GO term.

CLucene index is used inside Canto to search for ontology. Neves et al. have developed CellFinder based on ontologies, using CELDA (Cell: Expression, Localization, Development, Anatomy) in the structure of CellFinder. The use of CELDA ensures access to the cell and anatomy domain ontologies. Liu et al. have built the OntoMate tool to annotate data from the literature (designed primarily for the Rat Genome Database) [LLH+15]. The tool uses a dictionary-based ontology to allow the curator to look for ontology terms and add missing ontology terms if required. OntoMate contains four ontologies which are: “the rat strain ontology, the clinical measurement ontology, the measurement method ontology, and the experimental condition ontology” [LLH+15]. Sernadela et al. propose a semantic layer that uses ontology-based annotation techniques, which help in providing the ontology terms for the annotated data and relationships [SLC+15]. Wu et al. propose an approach to extract data from PDF documents, and as part of extracting names, they suggested using the National Center for Biomedical Ontology (NCBO) annotator [WOG+17]. In PhenoMiner, while extracting phenotype data, the tool obtains ontology terms associated with the extracted phenotype [LLS+13]. The ontology terms are taken from the following ontologies: rat strain ontology, clinical measurement ontology, measurement method ontology, and experimental condition ontology.

Since curators use ontology terms when annotating data, some issues related to finding and select ontology terms have started to appear in the literature. Ravagli et al. [RPM16] and Balhoff et al. [BDD+14] propose approaches to deal with cases where curators cannot find the required ontology terms. Balhoff et al. suggest a feature allowing data curators to ask ontology providers to add new ontology terms, or even edit existing ontology terms if needed [BDD+14]. This is achieved by implementing an Ontology Request Broker (ORB). The role of the ORB is to send the request in the form of an HTTP request, which is then sent to BioPortal API where the request is processed. However, Ravagli et al. focus on dealing with ontologies from a different perspective [RPM16]. The authors propose OntoBrowser, a web-based tool for curating ontologies. The idea of the tool is to allow domain experts to collaborate on curating and reporting issues related to ontology terms. OntoBrowser provides a shareable and collaborative environment via a central database, where users/curators can work on a single copy of the data. The tool is supported with a user-friendly interface, which allows curators to select the ontology they want to work on.

Other researchers have tried to manage the differences between ontologies by gathering a collection of related ontologies into one place. For example, Noy et al. have introduced BioPortal[23], an online repository that assembles a collection of biomedical ontologies into one access point [NSW$^+$09]. BioPortal provides a number of features related to ontologies, in addition to its main repository role. It allows users to access its services either programmatically or through the website. The users can integrate with the contents of the repository to map and edit ontologies. Similarly, the Ontology Lookup Service (OLS) provides a repository for biomedical ontologies [CJAH06]. It uses an AJAX-based approach to look up and suggest ontology terms in the database. As in BioPortal, OLS offers programmatic access and reports the results in Open Biomedical Ontology[24] (OBO) format. Welter et al. propose the Genome-Wide Association Studies (GWAS) catalogue, which is dedicated to supporting curation of information on single-nucleotide polymorphisms (SNPs) [WMM$^+$13]. The catalogue mainly focuses on three aspects: catalogue data, ontology, and data access. The GWAS catalogue searches for literature in PubMed. In the early stages of producing the catalogue, the extraction of SNP data from the literature was carried out manually. Now, however, the data is extracted and stored automatically. As there are different ontologies available, GWAS has a schema ontology which offers the ability to map terms from different ontologies. The ontology schema is built based on EFO application ontology.

Some researchers have focused on the idea of making sure that ontology users are accurately using the terms. For instance, Auken et al. propose an approach to find accurate GO annotations from articles [VASM$^+$14]. The idea is to provide metadata with the GO terms extracted from the articles. The metadata includes the GO term, the GO evidence code, and the gene.

[23] https://bioportal.bioontology.org/

[24] WWW.obofoundry.org/

2.2.3.1 General Curation Platforms

Several researchers have focused on designing and implementing curation platforms to serve different purposes of data curation, and some research provides approaches that can be applied to any field and are not specific to a particular domain.

Bour et al. [BMSN+16] and Stonebake et al. [SBI+13] both present curation platforms that serve a general purpose and are not designed purely for the biomedical domain. Bourgonje et al. have proposed a curation platform that helps curators to minimise the time required for curation [BMSN+16]. The design of the platform is based on a combination of three approaches: 1) Natural Language Processing; 2) Information Retrieval; and 3) Machine Translation. Moreover, Stonebake et al. propose DataTamer, an end-to-end curation platform. DataTamer focuses on dealing with four issues in data curation, which are scalability through automation, data cleaning, non-programmer orientation and incremental integration [SBI+13]. DataTamer is designed to lower the rate of human interaction required, as it uses a DataTamer Administrator (DTA) to provide the system with data to be curated. The system should also have one or more Domain Experts (DE), who can be contacted if there is an issue requiring human expertise that cannot be solved by the system. However, few cases to require human experts to interact with the system.

2.3 Conclusion

In this chapter, we discussed the existing research that has been conducted in the field of biocuration. Researchers have used several approaches to improve and expedite the way in which curation teams search for articles, extract genes/protein data and relationships, and find evidence in the articles referred to in the extracted data. In short, this research is focused on providing a better method of curation.

Most of the literature is concerned with curating data based on extracting information from published articles. The primary aim of much of the research is to find different ways to improve the curation process and make it more straightforward, efficient and cost-effective. Other research has looked at different aspects of biocuration: for example, by proposing approaches to help facilitate the process of finding ontology terms when annotating data or when curating data from other resources.

The first gap in the literature that we observed was that although some researchers have proposed biocuration workflows, less attention has been paid to providing a general model to describe the curation process across all components and to describe the things that need to be done to perform curation at a sensible level that meets the community needs. In other words, to the best of our knowledge, no resource has so far been designed to help biocuration communities assess the maturity of their curation process and reach the target maturity level that they require.

Another gap in the literature is that, less attention has been paid to reuse curation efforts to benefit other databases, as the curation efforts were limited to the databases which collaborate with each other to curate data, such as in MIntAct [OAA$^+$13] and DataShare [ACS$^+$14]

In summary, our literature review found that research in biocuration aims to improve the efficiency of the curation process, speed up the process, enhance methods of extracting data, and integrate curation approaches with the curation process (automating the curation process).

Chapter 3

A Maturity Model for Biocuration

The soul never thinks without a picture.

Aristotle

The previous chapter argued for not having a general description of the biocuration process components that shows the flow of the overall process, as different communities use different ways to curate the same data.

Here we describe how we addressed the gap by proposing BIOC-MM, a maturity model for the curation of biomedical databases. BIOC-MM does not focus on describing the curation process; instead, it provides a way to assess specific curation processes in biomedical communities and presents a summary of the standard procedures of data curation in biomedical communities. The chapter begins by giving a brief introduction to the concept of maturity models and surveys the literature (Section 3.1). Then, we introduce and explain the motivation for BIOC-MM (Section 3.2) and give more details of BIOC-MM by focusing on the maturity levels and model components (Section 3.3). Section 3.4 illustrates the curation processes followed at each maturity level. Note that we describe how to use a maturity model (Section 3.5). This ends the chapter in the 3.6 section.

3.1 Introduction to Maturity Models

This section introduces the concept of a maturity model. We focus on answering two questions: what is a maturity model, and what is the structure and process that need to be followed to develop a maturity model? According to Paulk et al., “the Capability Maturity Model for Software provides software organisations with guidance on how to gain control of their processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence” [PCCW93, p. 5]. The model helps assess the maturity level of the process, and thus guide communities to reach their target level.

Paulk et al. first defined the term *maturity model* in the context of software process [PCCW93]. According to [PCCW93], a maturity model can have five different maturity levels, which are shown in Figure 3.1 and described as follows:

1. “Initial - The software process is characterised as **ad hoc**, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.” [PCCW93, p. 8].

2. “Repeatable - Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.” [PCCW93, p. 9].

3. “Defined - The software process for both management and engineering activities is documented, standardised, and integrated into a standard software process for the organisation. All projects use an approved, tailored version of the organisation’s standard software process for developing and maintaining software.” [PCCW93, p. 9].

4. “Managed - Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.” [PCCW93, p. 9].

5. “Optimizing - Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.” [PCCW93, p. 9].

These five maturity models mentioned above should be taken into account when building a maturity model. Figure 3.2 shows an example of a maturity model. This model has five maturity levels, as described previously, and contains six components. The components represent the core aspects that need to be considered in each level.

The design of a maturity model can consist of one or more components. Using a maturity model is not confined to one purpose. Pöppelbuß and Röglinger [PR11]

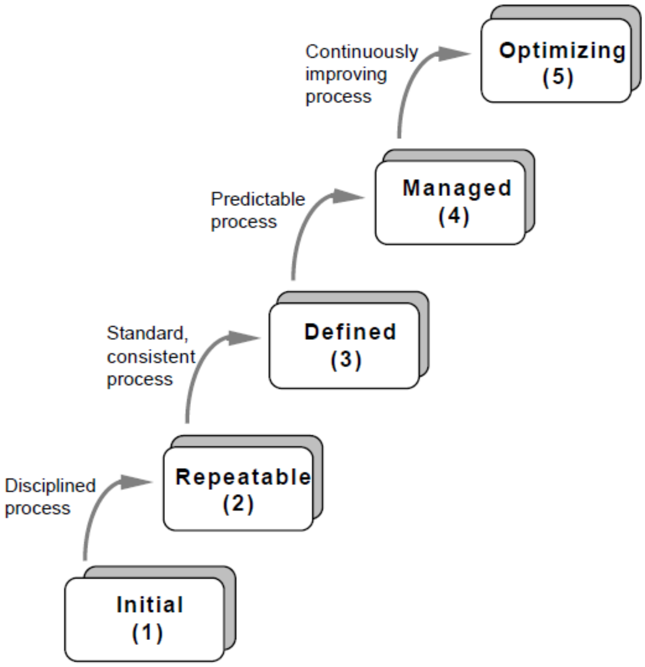


Figure 3.1: The Five Levels of Software Process Maturity [PCCW93].

summarised the uses of maturity models based on the work of Becker et al. [BKP09], De Bruin et al. [DBFKR05], Iversen et al. [INN99] and Maier et al. [MMC09]:

1. Descriptive purpose, which focuses on as-is assessment when designing the maturity model.

2. Perspective purpose, which focuses on the idea of measuring and enhancing the maturity of communities.

3. Comparative purpose, which focuses on comparing and benchmarking the model with the other models available.

Having a maturity model could help communities to work based on their needs, while avoiding the performance of tasks that consume their resources. The maturity model could also help to determine the correct level to meet communities' needs.

3.1.1 Literature on Maturity Models

Previous research on maturity models can be divided into three areas.

1. Many researchers have studied methods for designing maturity models.

2. Researchers have focused on implementing a maturity model that targets a specific aspect of a community and cannot be generalised to different communities.

3. Some researchers have proposed different ways to evaluate maturity models.

De Bruin et al. proposed a methodology that explains the main phases in the creation of maturity models [DBFKR05]. They divided the process into 6 phases: scope, design, populate, test, deploy, and maintain. However, it is essential to determine the purpose of the maturity model at the design stage, as this will help to clarify the procedures for developing such a model. The authors explain the proposed methodology for developing maturity models through two examples.

Pöppelbuß and Röglinger introduced a set of general design principles (DPs) for maturity models [PR11]. They claimed that using DPs while implementing maturity models will help to ensure that the resulting maturity models meet the criteria and intended purposes. The DPs are divided into categories that suit each of the three purposes of using maturity models. The proposed DPs in this study were applied to a specific domain, but the DPs are not limited to a specific domain and can be applied to different domains.

The Institute of Internal Auditors also describes a sequence of steps to build maturity models [The13]. The paper proposes eight steps for building and using maturity models:

1. To build a maturity model, the purpose of the model needs to be stated clearly. The authors proposed a few questions, the answers to which allow the purpose of the mode to be formulated. Part of answering the question involves defining the components to include in the model.

2. Determine the scale of each component by determining the aim of each component in each level.

3. Develop the expectation for each component level by providing the curation practices of each component level. By this stage, the maturity model is ready to use.

4. To use a maturity model, the organisation needs to clearly indicate their target maturity level for each component.

5. Based on the target stated in the previous step, each component needs to be assessed against each level.

6. At this stage, the organisation should point out any missing aspects in the model.

7. A report indicating all the steps taken to follow the maturity model should be produced and kept in the organisation record. Based on this report, the organisation should have a clear vision of their actual level of maturity and the target level they would like to achieve.

8. It is important to check that the organisation is following the model in order to perform the required practices.

The authors illustrated the eight steps through the use of three examples of maturity models: a Fortune 100 company process capability maturity model; a compliance and ethics programme maturity model; and a public sector internal audit capability maturity model.

Becker et al. also focus on designing maturity models as a guide for IT support [BKP09]. The authors propose a procedure for developing maturity models by following the eight requirements for maturity model developments. They compare and analyse the design process of 51 different maturity models.

Other groups of researchers have focused on building maturity models to serve different goals: in education, business, software development, and other areas. The following section will discuss several of these different types of maturity models.

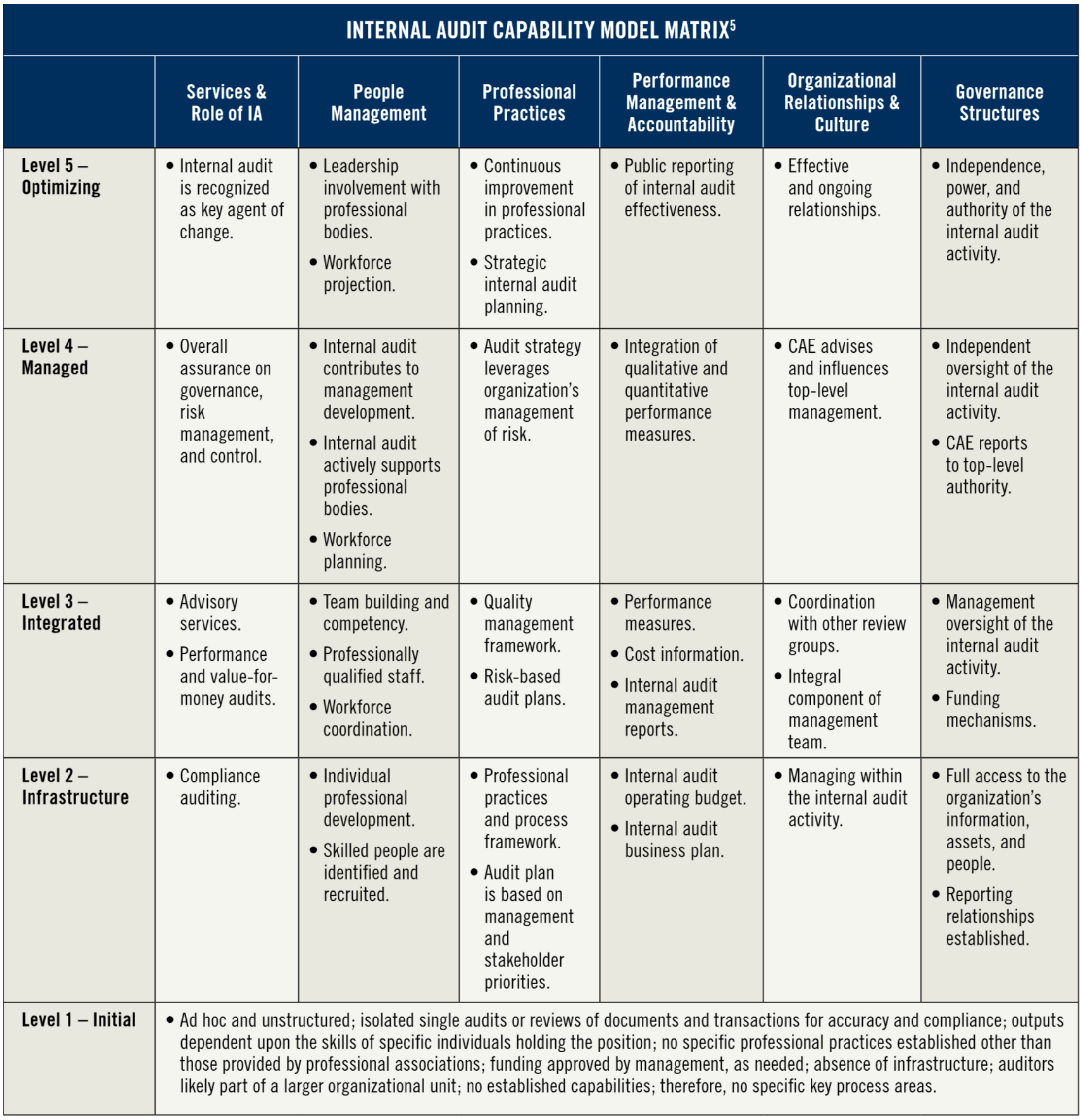
Huang et al. focus on providing a maturity model that helps in improving the quality of the documentation process of software [HT03]. As delivering documentation for software development is part of the process, the documentation maturity model (DMM) assesses the quality of the provided documentation. The paper follows the five maturity levels and adapts them to suit the DMM's purpose.

In the education field, there have also been some efforts to design maturity models. Marshall et al. introduces a maturity model for e-learning [MM02] which follows a similar process as the original capability maturity model proposed by Paulk et al. [PCCW93]. The suggested maturity model focused on two aspects of learning by looking at the institution and the courses.

Clarke et al. focus on another aspect of education [CNS13]. Their model deals with different issues found in educational institutions, aiming to improve the education process by assessing students' behaviour. The authors benchmarked the model with the previous research and overcame some of the gaps.

Other research has focused on improving maturity models for data quality management. Ofner et al. describe the adoption and adaptation of an existing maturity model to meet specific requirements [OHO09]. The authors adopt the corporate data quality management maturity model proposed by [HOO09]. This combines the use of two methods (Design Science Research (DSR) and Method Engineering (ME)) to satisfy the condition of dealing with complexity in corporate data quality management. Moreover, Ofner et al. propose a maturity model for enterprise data quality management [OOÖ13]. In the process of implementing the model, some existing data quality management maturity models were compared. The authors also use some data quality management approaches to implement the model.

To the best of our knowledge, despite all these maturity model design efforts, there is as yet no maturity model for biocuration.



INTERNAL AUDIT CAPABILITY MODEL MATRIX[5]

	Services & Role of IA	People Management	Professional Practices	Performance Management & Accountability	Organizational Relationships & Culture	Governance Structures
Level 5 – Optimizing	• Internal audit is recognized as key agent of change.	• Leadership involvement with professional bodies. • Workforce projection.	• Continuous improvement in professional practices. • Strategic internal audit planning.	• Public reporting of internal audit effectiveness.	• Effective and ongoing relationships.	• Independence, power, and authority of the internal audit activity.
Level 4 – Managed	• Overall assurance on governance, risk management, and control.	• Internal audit contributes to management development. • Internal audit actively supports professional bodies. • Workforce planning.	• Audit strategy leverages organization's management of risk.	• Integration of qualitative and quantitative performance measures.	• CAE advises and influences top-level management.	• Independent oversight of the internal audit activity. • CAE reports to top-level authority.
Level 3 – Integrated	• Advisory services. • Performance and value-for-money audits.	• Team building and competency. • Professionally qualified staff. • Workforce coordination.	• Quality management framework. • Risk-based audit plans.	• Performance measures. • Cost information. • Internal audit management reports.	• Coordination with other review groups. • Integral component of management team.	• Management oversight of the internal audit activity. • Funding mechanisms.
Level 2 – Infrastructure	• Compliance auditing.	• Individual professional development. • Skilled people are identified and recruited.	• Professional practices and process framework. • Audit plan is based on management and stakeholder priorities.	• Internal audit operating budget. • Internal audit business plan.	• Managing within the internal audit activity.	• Full access to the organization's information, assets, and people. • Reporting relationships established.
Level 1 – Initial	• Ad hoc and unstructured; isolated single audits or reviews of documents and transactions for accuracy and compliance; outputs dependent upon the skills of specific individuals holding the position; no specific professional practices established other than those provided by professional associations; funding approved by management, as needed; absence of infrastructure; auditors likely part of a larger organizational unit; no established capabilities; therefore, no specific key process areas.					

Figure 3.2: Public Sector Internal Audit Capability Maturity Mode [The13].

3.2 A Maturity Model for Biomedical Curation

This section describes BIOC-MM, a maturity model for biomedical curation. The purpose of BIOC-MM is to help communities which have no prior knowledge of biocuration to understand the basic components of curation and understand the curation process. At the same time, it is intended to help communities that already practise curation to assess the maturity level of the curation processes and help them to achieve the level of maturity that meets their goal and resources. BIOC-MM also provides a number of available techniques and approaches needed at each level of maturity in order to satisfy the maturity goal.

Our aim when building the maturity model was to use available information about the biocuration process, without contacting curators. We focused in finding this information by reviewing the existing biocuration literature. When we first looked at the literature, we found that it was mostly focused on automating the curation process, and on curating data based on literature. This led us to create an initial model with components focused on literature curation, such as searching for articles and extracting data from article abstracts. The levels focused on automating the curation process. However, when we looked at the biocuration workflow proposed by Hirschman et al. [HBK$^+$12] and investigated the curation process used in five different biomedical databases, we found that the curation process covered areas other than automation and literature curation.

It should be noted that the maturity model went through several development and refinement stages until it reached the state shown in Table 3.1. When designing the model, two main factors affected its formulation. First, the existing literature on biocuration from the last five years was reviewed. The review covered aspects such as techniques and approaches to improve data curation. Second, we investigated the curation process used in five different biomedical databases:

1. The Universal Protein Resource (UniProt) [1]. UniProt contains a collection of protein sequences. The structure of UniProt consists of three databases: UniProt Knowledgebase (UniProtKB), UniProt Archive (UniParc), and UniProt Reference Clusters (UniRef).

2. BioGrid [2]. “BioGRID is a freely accessible database of physical and genetic interactions.” [SBR$^+$06, p. 1].

[1] www.uniprot.org

[2] www.thebiogrid.org

3. FlyBase [3]. “FlyBase provides an integrated view of the fundamental genomic and genetic data on the major genetic model Drosophila melanogaster and related species.” [Con03, p. 1].

4. Saccharomyces Genome Database (SGD)[4]. “SGD provides Internet access to the complete Saccharomyces cerevisiae genomic sequence, its genes and their products, the phenotypes of its mutants, and the literature supporting these data.” [CAB$^+$98, p. 1].

5. Rat Genome Database (RGD)[5]. “The focus of RGD is to provide a disease-centric perspective of the rat genome for the community, which is generally organized scientifically by disease speciality’.’ [TLS$^+$02, p. 127].

We examined all the documentation provided regarding the curation processes used for these biomedical databases, whether these were online curation guidelines or publications produced by the consortia responsible for managing the data. The model also went through a couple of refinement stages, in which domain experts reviewed the model to ensure that it reflected the actual needs of real biomedical communities.

The five databases were chosen for two reasons. First, each one of these databases use a different way to curate data. UniProt, for example, focuses on dividing the curation process into manual curation and automatic curation. On other hand, curators need to search for new publications and then invite the authors to participate in the curation process by emailing and asking them to fill in some information about their publication. The rest of the databases use a tool to automatically extract articles from PubMed. However, BioGrid curators decide whether an article is curatable before it is curated in full. SGD curators curate the extracted articles manually. RGD takes the extracted articles and extracts data from the abstract automatically, after which curators are responsible for manually curating the rest of the articles. Second, the five databases provide clear documentation on how they carry out curation. This helped us to understand their curation process without any need to directly contact curators (one of our aims was to not request extra work from curators and to try to focus on the use of available resources. In general, these databases show a variety of ways to curate data, and each one follows the curation practices which meet its needs. The differences between these methods of curation help us to better understand the various ways of curating data.

[3] www.flybase.org

[4] www.yeastgenome.org

[5] www.rgd.mcw.edu

3.3 BIOC-MM Structure

BIOC-MM is designed in a matrix form; the columns in the matrix refer to the levels of the maturity model. Following Paulk et al., we divided the proposed maturity model into five maturity levels, ranging from level 1, the lowest, to level 5, the highest [PCCW93]. The rows in the matrix represent the main components of curation, which were extracted from the review of the biocuration literature and the biomedical databases, as mentioned previously. The following section describes the rows and columns of BIOC-MM in detail.

3.3.1 The BIOC-MM Maturity Levels

As mentioned previously, the columns of the matrix refer to the maturity levels. The model, in general, contains five levels, where each level is designed to satisfy specific goals. Paulk et al. state that the maturity levels in a maturity model need to convey five aspects of maturity [PCCW93]. In BIOC-MM, we renamed the five maturity levels to meet the goal of the levels in relation to the task required at each level. The five maturity levels of BIOC-MM are as follows:

- Level 1: ad-hoc curation.
- Level 2: standardised curation.
- Level 3: curation at scale.
- Level 4: collaborative curation.
- Level 5: analytical curation.

When we look at BIOC-MM, we can see that the model starts with a bottom-level, ad-hoc curation. Here, the curation process is very basic and the curation team work individually. Moving to the next level, standardised curation, the team starts to work as a group rather than working individually. The reason why we chose this level for curators to work as a team is because the curators need to take the most relevant practises from level 1 so that the entire curation team can use them. In level 3, after the curation process becomes more organised, curators can focus on solving issues related to curation, such as dealing with huge amounts of data, as the curation process is a time-consuming task. The fourth level concerns the provision of a collaborative environment to reduce the amount of redundant work across databases. The highest level moves from concerns about the practises of the curation process to the analysis of the curation results. This level also involves the use of other factors that are not core aspects in the curation process, such as incorporating reviewers' feedback on publications while curating data. All these kinds of information could be used to extract more useful data, enhancing the curation process by improving the understanding of other factors which can be directly or indirectly connected to curation.

3.3.1.1 Level 1: Ad-hoc Curation

A biocuration operating at level 1 is concerned with performing basic curation procedures. The curation team at this level does not have agreed procedures for curating data, with each curator working based on her/his own perspective and domain knowledge. In other words, curators work individually to find and correct defects in data. In doing so, each may take a very different approach and make different decisions. The individual curation process does not have clear guidelines defining the things that curators should focus on while curating data.

3.3.1.2 Level 2: Standardised Curation

At the standardised curation level, the curation team focuses on achieving a consistent form of curation by making curators work as a team rather than as individuals. This level's goal is achieved by providing clear guidelines for curators to follow and by developing standard curation processes.

3.3.1.3 Level 3: Curation at Scale

At this level, curation teams focus on dealing with other curation aspects that need to be managed while curating data to improve the overall process. This level describes and explains the action required from curators to deal with the large volume of literature and data available for curation. Thousands of new articles become available all the time and it is a difficult task to cope with all incoming literature. In other words, this level's goal is to provide various strategies to efficiently deal with with the massive amount of literature and data to curate.

3.3.1.4 Level 4: Collaborative Curation

After the processes followed in previous levels have achieved clear, organise and control way to perform data curation, the community at this level is ready and prepared to be interactive and to collaborate with other biomedical communities that share the same interest. In other words, communities can collaborate and share the curation work between them to avoid redundant work.

3.3.1.5 Level 5: Analytical Curation

The most mature level in our model does not focus on the curation process itself, but rather on the results of the curation process. It uses the curation results and other external factors to produce a higher degree of information and understanding, and use this awareness to serve the curation process. This leads us to refer to this level as analytical curation, as it involves analysis of the available curation results alongside other factors.

3.3.2 The BIOC-MM Components

As mentioned previously, the BIOC-MM was designed based on a review of the biocuration literature and of the curation protocols of five biomedical databases. The rows of the model refer to the curation components, which are summarised into six main steps that any biomedical curation team must undertake The components are as follows:

- Literature-based curation.
- Data-based curation.
- Quality assurance.
- Data management.
- Computational annotation.
- Community building.

3.3.2.1 Literature-Based Curation

This component concerns curating data with relevance to the literature. Literature-based curation covers things such as identifying related articles, ranking them, and obtaining data from them. It should be noted that not all aspects are included in all levels, depending on the level's goals. A more detailed view of the component levels will be presented later in this chapter.

3.3.2.2 Data-Based Curation

This component concerns curating the data itself by finding and fixing errors that are either present in the data, or based on findings from data in other resources. It also involves deriving data from sources such as UniProt using tools to derive data related to sequence analysis during manual curation.

3.3.2.3 Quality Assurance

This component concerns the quality of the curated source, which means dealing with the management and organisation of the curated source. This component covers issues such as providing guidelines and dealing with the data format.

3.3.2.4 Data Management

This component covers the management aspects of the data, rather than the curation process. Data management covers aspects such as data quality, data collection, and data access and retrieval.

3.3.2.5 Computational Annotation

This component describes the use of computational annotation, such as ontologies, when annotating data. Computational annotation is added as a component because it contributes to an essential part of the curation process.

3.3.2.6 Community Building

This component does not focus on the data curation process, but rather on how the curation team act to curate the data and the things that a community needs to follow to manage the curation process.

3.4 BIOC-MM Description

Before describing the model components at each level, we should mention that we divided the curation practices by themes. According to Braun and Clarke, “thematic analysis is a method for identifying, analysing and reporting patterns (themes) within data.” [BC06, p.79]. When we read the literature and investigated the five databases mentioned previously, we grouped these practises into themes based on the model components mentioned previously (Section 3.3.2). We then classified our findings for each component in relation to its relevant level. The rest of describes the model components at each level, with examples of available curation practices.

3.4.1 Level 1: Ad-hoc Curation

As mentioned previously, the first level is the ad-hoc level, at which all components follow basic manual procedures to perform the curation process. In the literature-based curation component, curators work individually to detect defects and fix them based on the literature, following manual procedures to select and curate the related literature. At this level, curators still do not have a clear description of the method for curating data and the key aspects to focus on while doing so. Similarly, in terms of quality assurance, the quality aspects of the curation process are not defined; when a quality issue appears in the curation process, it will be resolved by curators if required. Data management at this level follows ad-hoc procedures, and not all aspects of data management are satisfied. This level focuses on the collection of data by the curation team. The community provides access to the most recent version of the curated data only. However, there is no attempt to use computational annotation while annotating data; adding computational annotation terms while curating is a procedure that requires the organisation of the community team. The community overall has not formalised the curation process at this level, and the curation process is carried out by individual curators without any parameters to help them to work in an organised way.

3.4.2 Level 2: Standardised Curation

The goal of this level is to move from the ad-hoc stage of the work to a formalisation of the curation process. This is achieved by setting criteria and standardise the curation process by setting rules for the curation team regarding how to curate data. At this level, data curation approaches start to have some automation in all components (unlike the previous level, in which the entire curation process was manually operated).

Starting with the first component of the model (literature-based curation), new approaches are set at this level which includes the following aspects:

1. Defining approaches to search, select, and rank the relevant literature.

2. Determining if the selected articles are curateable or not, so curators do not have to go through non-relevant articles.

3. Approaches to extracting data from some parts of the articles, such as title and abstract.

In the reviewed literature and the five biomedical databases, these aspects were tackled to some extent by by several communities and research groups. The following list shows some of the existing approaches that satisfy this level's goal:

1. PubTator is a web-based text mining tool for assisting biocuration [WKL13]. PubTator helps curators who have limited experience in text-mining by providing a user-friendly interface. The tool covers different aspects related to searching, retrieving and annotating articles.

2. Canto is an online tool for community literature curation produced by PomBase [RHL$^+$14]. Canto provides an interface that allows users to select and curate articles, and to invite the authors of new literature to participate in the curation process.

3. OntoMate is a literature search engine tool that extracts literature and tags abstracts from the literature [LLH$^+$15].

4. Szakonyi et al. proposes the KnownLeaf literature curation system, which is built based on two tools: knownleaf and leafNet [SVLB$^+$15].

5. Neves et al. introduces a curation pipeline for the CellFinder database, based on the literature, that extracts some data such as cell types, cell lines, and organs. The pipeline consists of text mining tools, and the results of curation are validated manually [NDM$^+$13].

6. ODIN is a graphical interface for literature curation, which can be run within a web browser [RDS$^+$13]. ODIN is dedicated to biomedical data curation and uses a combination of text-mining tools and techniques.

7. PIMiner is a web-based tool designed to extract protein interaction information and relationships from PubMed abstract articles [CZT$^+$13].

8. RLIMS-P is an online text-mining tool for literature-based extraction of protein information which helps curators identify biomedical research relevant articles [TLL$^+$14].

Curators operating at the second level, data-based curation, start to follow a more organised approach. A curation team may create documentation describing the processes used for data curation. By providing documentation, the curators within a community will start to work as a team, rather than working individually as in the previous level. One form of documentation is curation guidelines. These guidelines can be general or specific; for instance, they may focus on discussing the curation of a particular element of data.

For example, UniProt consortium provides both general and specific guidelines as it gives a general guideline that describes dealing with manual curation such as UniProt Manual Curation SOP [Con14b]. UniProt also provides guidelines that serve a specific goal, such as protein name formatting [Con17]. The naming guidelines cover issues such as conventions on protein names (i.e., the name should not contain disease names) and procedures for dealing with legacy names that bunch the convention. The provided guidelines are not fixed; new guidelines can be added or edited based on the community requirements.

The third component, quality assurance, is concerned with ensuring quality in the curation process. Communities may add audit trails at this level, which allow data curators to add comments and justifications when they curate data. The comments help curators from other teams, communities, and data consumers to understand the reason for a change in data. Also, this level includes adding guidelines for quality assurance, in line with the goal of standardising the curation process. The guidelines for this component cover wider aspects than those included in the previous component. Here, the guidelines should discuss aspects related to the whole curation process and not only those related to data.

The fourth component, data management, focuses on data collecting, data querying, and checking fundamental data quality aspects. These aspects are examined by pulling data from the literature. Another data management aspect is data querying, and ensuring basic quality aspects of data such as completeness and consistency.

The fifth component, computational annotation, curators at this level start to use computational annotation terms while annotating data. Communities can adopt existing computational annotation terms or develop their own, such as the Gene Ontology (GO)[6], and the Evidence and Conclusion Ontology (ECO)[7]. When adopting terms from other ontologies, the community should take note of whether the ontology

[6] www.geneontology.org

[7] www.evidenceontology.org

provider is actively maintaining the ontology.

The last component, community building, focuses on providing guidelines to data consumers for submitting data, giving feedback, or reporting errors. UniProt, for example, gives instructions for researchers to send it updates, while Flybase allows their data consumers to fill in a form reporting any comments. Sowe et al. propose a cloud platform to support community-based curation, which is designed to serve a community of disaster response scientists [SZ13].

3.4.3 Level 3: Curation at Scale

We now explain the procedures followed in each curation component by teams working the *curation-at-scale level*. In the level previously described, the curation follows a standard process; this helps curators within a community to follow a clear procedure for curation. However, level 3 focuses on the way a community should deal with a large body of literature and data for curation.

Curators deal with the first component, literature-based curation, by following two processes to organise and share literature curation with communities of the same interest. First, the curation community must manage the literature by ordering and prioritising it, as the amount of incoming literature is huge and curators should be selective in their approach. Second, the curation process continues to support automation at this level to adopt a shareable environment for curation.

The rest of this section highlights some existing approaches that cover the aim of level 3. Poux et al. propose the use of multiple literature triage approaches, aided by the PubTator tool, to prioritise articles in a better way [PAM+17]. The literature is classified into three categories: curatable, not a priority, or not curatable. The community benefits from more automation, both in the curation process and to provide a shareable environment for curation communities. The sharable environment will help communities to deal with the literature, if communities agree to divide the curation task between them. For example, communities which curate the same list of journals can ease the workload by assigning the curation of a journal to a specific community, who then share the curation results with the others. This saves time and effort.

LiverCancerMarkerRIF, a text mining-based curation system, offers the ability to curate data using a user-friendly interface [DWL+14]. The aim of LiverCancerMarkerRIF is that it allows both users and curators to find evidence on liver cancer while they use the PubMed website.

UniProt, for example, provides a document which describes how a community like

UniProt works in terms of literature-based curation, as it follows two types of curation: manual and automatic [C$^+$14].

Jamieson et al. focus on assisting manual curation by providing a semi-automatic approach that uses text-mining to create a custom topic-specific molecular interaction database [JRR$^+$13]. “The approach demonstrates that combining existing text mining tools with domain-specific terms and wiki-based visualisation can facilitate rapid curation of molecular interactions to create a custom database” [JRR$^+$13, p. 1].

Singhal et al. propose an approach focused on both the local and global context of each mutation to extract genes and proteins [SSL16]. They also developed a text-mining based machine learning approach to integrate protein sequence validation with disease association. The proposed approach uses PubMed abstracts to find disease-gene-variant triplets.

Rinaldi et al. presented an approach to deal with incoming literature by providing scaling procedures to identify domain entities and semantic relationships [RFC15]. The approach focuses on providing three features: extraction of protein interaction information, sorting the results, and providing a GUI to allow users to curate data.

Dai et al. proposed AuthorReward, an extension to MediaWiki, which gives authors the chance to participate in the curation process [DTW$^+$13]. The system sends a form to the author asking them to provide information which helps curators to determine whether to curate the paper or not, and so speeds up the curation process. However, using this method (i.e., of asking the authors to undertake part of the work needed to curate their results) might not work as well as a professional curator. Also, author participation is optional, so responses cannot be guaranteed.

Urban et al. discussed the addition of some features to the Pathogen-Host Interactions database (PHI-base) [UPR$^+$14]. These summarised into four key features: covering more taxonomic ranges and controlled vocabulary, covering more species, linking to other external sources, and functional analysis of PHI-base accessions. The authors also suggest some enhancements at the technical level, which includes developing data curation and release management, and mapping PHI-base phenotypes to Ensembl Genomes.

In terms of the data-based curation component of the model, teams working at this level consider adding more automation to the curation process, with semi-automated approaches being used to detect errors in data and propose corrections. However, this does not replace the role of human curators altogether. The approaches focus more on smoothing the process of curation, as it saves curators time in identifying issues in data and suggests solutions. However, curators need to check and authorise these to ensure the process is performed correctly. For example, an automatic tool which detects an incorrect protein name and gives suggestions for the correct protein name requires the curators need to see and agree on the suggested protein name. The following list shows examples of existing approaches which helps in speeding up an scaling up the curation process:

1. Yepes et al. propose the use of tables and supplementary materials provided with the literature to curate data [YV13]. The authors suggested a text-mining approach to annotate tables and supplementary materials provided with the publications. Both tables and supplementary materials are converted to text and processed via the EMU[DKFB+10] tool.

2. Some researchers, such as Keseler et al., suggest assessing the accuracy of the curation process [KSW+14]. The method for checking accuracy focuses on manually examining arbitrary data records from two databases: the Escherichia coli database (EcoCyc) and the Candida Genome Database (CGD). The accuracy is tested by checking the availability of the data in the database's records against the information from the cited publication.

3. Khare et al. proposed the use of the Amazon Mechanical Turk (MTurk) environment to apply to crowd-sourcing techniques to scale drug annotation [KBA+15]. The authors mentioned that their method successfully achieved 95% accuracy in annotating drugs.

The quality assurance component, at this level, has to follow standardised data representation. By standardising the data representation, data consumers and other communities will use and share data more easily, as the ambiguity of having unclear data representation will be decreased. Also, automation can be added at this level to detect and fix some quality issues. Argo is an example of this, providing a user-friendly graphical interface for the biocuration process [RBNR+14]. The interface is built based on text-mining solutions, which help in assisting curation.

The data management component has two aspects: data submitting and data archiving. The former relates to how data is submitted in a data resource. PhenoMiner is an example of a tool that allows researchers to submit data through a form (built by Rat Genome DB) [LLS+13]. The second aspect is data archiving. In the previous level, this component focused on providing access to the current version of data only.

However, data archiving at this level provides access to archives for the current and previous versions of the data, which helps keep track of any changes made. Examples of providing archives to satisfy the goal of automation include:

1. "The UniProt Archive (UniParc) is a comprehensive and non-redundant database that contains most of the publicly available protein sequences in the world." [Con16]. It provides an archive to access all versions of proteins records.

2. Unisave is a tool which compares two versions of a protein record and allows to download the comparison results [LNZA06].

3. Poux et al. discuss a case study that deals with data errors and consistency in data when integrating with other sources [PMA+14].

4. "MetaboLights is the first general-purpose open-access curated repository for metabolomic studies." [SHC+13, p. 1]. The authors focus on the challenges posed by curating in metabolomics, such as dealing with different sizes of data submitted to the repository to curate.

5. Croset et al. aim to resolve issue raises in data when it is integrated from multiple resources. [CRR15]. The authors used a graph-based approach to ensure that no errors occur.

In level two, curation teams use ontologies when annotating data. However, sometimes curators cannot find the appropriate computational annotation terms for the desired purpose. At this level, the computational annotation component covers aspects other than the selection of annotation terms. The curators can request annotation terms from the annotation providers if required. For example, Phenex [BDD+14] is based around an Ontology Request Broker (ORB). The ORB allows curators to request or edit ontology terms; the requested terms will then be checked and added for future use.

Since many curators need to deal with various data, the need to use terms from different ontologies arises. This requires curation teams to consider gathering the required ontologies into one catalogue. Such a catalogue allows users to retrieve ontology terms from different ontologies listed in the catalogue. The GWAS catalogue [WMM+13], BioPortal [NSW+09] and Ontology Lookup Service (OLS) [CJAH06] are popular examples of an accessible platform that allows users to search for different ontology terms and request the addition of missing terms. Also, Friedman et al. describe an approach known as Marshal, specifically dealing with how to improve and repair ontologies automatically [FBB15].

In general, teams working at level 3 try to build a more consistent community that can deal with the difficult tasks faced by the curation team. The teams start to standardise data submission in a way that allows data submitters to participate in the curation process; data submitters are required to fill in relevant information when submitting data. The teams also set priorities to curate data in a community. Examples of these ideas in action include:

1. Data Tamer is an example of an end-to-end curation system that combines curation components [SBI$^+$13]. As part of the system, this follows a standard process to present the text through a parser.

2. BioGrid: the Biological General Repository for Interaction Databases [CABO$^+$14].

3. The SEAD project provides data services which minimise the curation efforts made [MHA$^+$15].

4. Poelchau et al. discuss the Web Apollo software, which enables smaller, focused genome groups to coordinate manual annotation efforts across geographically-distributed labs [PCM$^+$14].

5. BLAHmuc investigated the use of two recently-developed tools: 1) Medicate, a search engine for navigating through publications from MED-LINE; and 2) TextAI, a machine learning-enriched extension of the TextAE annotation tool [RW16].

6. Web Apollo is a tool that provides a real-time genomic collaborative annotation between curators in the same place, or in different places [LHR$^+$13]. Guidelines are provided on how to annotate data using Web Apollo.

3.4.4 Level 4: Collaborative Curation

At this level, curation teams care about the expansion of the use of curation efforts. We now discuss how the curation components work to satisfy this level's aims. The curation teams work through different ways to carry out collaborative curation, such as using a common curation platform where a number of curation teams can share and collaborate. The following list shows examples of approaches use collaboration to enhance curation:

1. MIntAct, a common curation platform, on which 11 different biomedical databases collaborate and share their curation efforts [OAA$^+$13].

2. O'Reilly et al. propose a collaborative literature curation framework to help to establish a shareable curated corpora of annotations. The framework is designed specifically for neuroscience literature [OIH17].

3. Islamaj Doğan et al. also propose collaborative curation, focusing on ensuring that the process is of high quality [IDKCa$^+$17].

Other tools help in curating data by providing integration environment between resources, such as BioQRator [KKS$^+$14]. BioQRator is designed to support general biocuration tasks rather than focusing on one aspect for annotating [KKS$^+$14]. BioQRator can annotate both data and relationships in the literature, and it can be integrated with other systems.

Communities can benefit from the curation efforts made by other biomedical communities. Since some communities curate data earlier than others, they can notify other communities when another source is updated. Alqasab et al. propose the idea of the IQBot, which helps in spreading the curation efforts made with a biomedical resource to other resources [AES17].

The aim of the quality assurance component, at this level, is to share best practice between curation teams. DataShare is an example of a tool that allows users to share their curation practices [ACS$^+$14].

Mottin et al. propose ranking the literature based on a new method, specifically designed to extract articles related to protein-protein interactions and post-translational modifications [MPG$^+$17]. The extracted articles are ranked based on two metrics; the vector-space search engine and the measuring density.

Since curation teams at this level collaborate with each other to expand the use of curation, the data management component focuses on three aspects:

1. Linking data records between sources: many curated biomedical databases are linking their data records to those from different databases, in order to provide more information about their data (as in the UniProt database, where the data records provide links to data from other resources).

2. Providing API access: providing programmatic access for data consumers to all versions of data is important at this level, as it helps other users and communities to retrieve the data they need. It also allows users of the curated resources to

access data records programmatically by providing APIs. Four out of the five communities we reviewed provide APIs for users to access data.

3. Considering data provenance: this provides curators with the ability to track changes and find the origins of errors in data, if applicable.

Collaborative curation across communities and tighter linkage between data sources leads to a need to map between computational ontologies. Such mappings help curators to understand annotation terms that come from sources other than those used by a specific curator. This gives the curator a better understanding of annotation terms coming from different communities. The Ontology Xref Service (OxO)[8] and Zooma[9] are examples of approaches to ontology mapping. OxO maps between ontology terms by using the Ontology Lookup Service and relates it to the Unified Medical Language System (UMLS)[10] mapping. Zooma provides mapping to ontology terms while annotating text.

It can be said that, at level 4, curation teams are concerned with having a shareable and collaborative environment for curation in order to expand the range of available curation partners, as exemplified by MIntAct. MIntAct provides a curation platform that allows 11 biomedical databases to work in collaboration to share their curation efforts[OAA$^+$13]. In addition, “Egas is a web-based platform for biomedical text mining and assisted curation with highly usable interfaces for manual and automatic in-line annotation of concepts and relations” [CLMO14, p. 1]. Egas provides real-time collaboration service, allows curators to communicate with each other, and offers on-demand management and guidelines.

BioSharing is a search portal for biomedical data [MGBRS$^+$16]. BioSharing provides access to different types of data through the use of the core of the FAIR principle and the linking of three registries (Standards, Databases, and Policies).

3.4.5 Level 5: Analytical Curation

The maturity model levels covered so far deal with approaches that improve the curation process and make it easier and faster to carry out high-quality curation within and across communities. However, level 5 is different because it it primarily concerned with using the outputs of the curation process to improve the curation process itself.

[8] www.ebi.ac.uk/spot/oxo

[9] www.ebi.ac.uk/spot/zooma

[10] www.nlm.nih.gov/research/umls

In other words, the aim is to maximise the benefit of the curation process based on the work that has already been done.

In the case of the literature-based curation component, the curation teams should consider issues beyond the standard curation process of selecting and curating the literature. The team should focus on other aspects of the literature that are beyond the scope of ordinary curation. For example, before an article is published, it goes through many stages, one of which is peer review. Therefore, it would be helpful to have an approach that uses reviewer feedback (or other aspects of the literature life-cycle) to enhance the curation process by improving the understanding of its nature.

Processes in the data-based curation component focus on extracting patterns from the curation work made to the curated data source. This may help curation teams to better understand how the curation process is performed over time, and the factors that affect it. This also helps implement tools more productively. In other words, curation tool providers may consider adding a feature which takes the curation practice results and translates it into a form that visualises and summarises findings of the curation practices.

In the quality assurance component, the quality of the curation process is ensured by looking through curation processes made by the community (whether manual or automatic). This may help identify the type of quality issues that appear in a community and aid in dealing with these quality issues before they become a critical problem. In addition, approaches based on an audit trail can be used to support the goal of quality assurance in the community. This can be done by analysing the curator's behaviour while they are curating data, which may help other curators to acknowledge different responses to certain curation tasks, and their concerns while curating.

In the case of the data management component, the curation teams should consider approaches that cover quality issues in data across its life-cycle, from creation to archiving. Also, the curation teams need to use approaches to generate a report periodically about the type of problems in data.

The previous levels covered many aspects of using annotations, such as requesting new terms if not available and mapping between annotation terms. At this level, the focus here is on reusing the annotation terms used while curating data to streamline the process by, for example, using an approach that gives suggestions for the annotation terms that can be used when curators annotate data. This could help curators to quickly select the suitable annotation term, rather than browsing for the right annotation terms.

Lastly, the primary community concern at this level is to extract knowledge from the curators about their actions (i.e., the curation efforts made) to improve and enhance the curation process. This leads the community teams to provide documentation for the results of the curation process. The data in this documentation can then be analysed.

3.5 Using a Maturity Model

This section illustrates how our proposed maturity model might be used in practice by giving an example. In this example, a community that has only recently started to curate its data wishes to make improvements. They will use BioC-MM to identify possible "quick wins" for improvement, based on their current practices.

The community needs to carry out the following steps:

1. Identify the current maturity level of the community curation process against each dimension in the model.

2. Identify the dimensions where improvement is most needed and select the desired maturity level of each one. The desired maturity level should be close to the current level for this exercise. The assumption behind the use of maturity models is that there is no point in trying to jump to quickly from, for example, level 2 to level 5.

3. For each dimension where improvement is needed, use the descriptions of the levels between the current level and the target level to plan a series of staged enhancements.

Consider a simple example of a community that wishes to use BioC-MM to improve its processes. Assume that this community uses a tool downloaded from elsewhere to extract new publications from the literature every week, and that it can semi-automatically detect and obtain data from the abstract using a bespoke tool that it has developed. The community uses a basic collaboration platform to curate the full text of new publications. However, the repository data is still edited manually, and no audit trail information is gathered (apart from notes kept informally by curators).

Based on the description of the community mentioned above, this community is at level 1 for dimension 1, at level 2 for dimension 2, at level 3 for dimension 3, at level 3 for dimension 4, and at level 1 for dimension 5. The curators feel they are spending too long searching through new publications to find the ones they need to pay attention to and are beginning to struggle with the lack of any formal audit trail, as errors introduced by inexperienced curators are hard to detect and correct. So, the goal is set to reach level 3 in dimension 2 and level 2 or 3 in dimension 5. Interest is also expressed in making data changes easier, so a target of level 2 is set for dimension 1.

After deciding the target maturity levels, it is time to go through each dimension which is below its target in order to improve it. Dimension 1 should be moved from manually editing repository data to semi-automatic editing. If no existing tool can be found, then a bespoke tool will need to be created. The team might decide that this is not cost-effective for them at present. To reach level 3 in dimension 3, the community needs to find a tool that can extract relevant information from the abstracts of papers. They find a suitable text mining tool but need to put some effort into configuring it to work with their preferred annotation. The team has access to text mining expertise, and decides to go ahead with this improvement.

The last dimension to be improved is dimension 5. The team decides to jump two levels here, since they realise that they can adapt an audit trail model from another closely-related community, and also make use of tools provided by that community. The maturity model has helped them to make informed and defensible decisions about how to obtain the most improvement value from the available resources.

3.6 Conclusion

To conclude, this chapter has covered the idea of providing the biomedical communities with a model to assess the maturity of their curation process through the use of BIOC-MM. The chapter describes BIOC-MM and its related maturity levels and components. Its design was based on the literature of biocuration in the last five years and on details from the curation practices in use for five other biomedical databases. BIOC-MM summarises common curation procedures from the reviewed work in a form that can be applied to all biomedical communities. Each level of BIOC-MM aims to provide a list of approaches to adopt. It should be noted that the mentioned approaches are the most recent approaches used in biocuration. BIOC-MM opens a wider vision of biocuration for communities.

-	Level 1: Ad-hoc Curation	Level 2: Standardised Curation	Level 3: Curation at scale	Level 4: Collaborative Curation	Level 5: Analytical Curation
Literature-Based Curation	Ad-hoc, manual procedures used for ranking, selecting and curating publications.	Uses agreed-upon documented approaches for selecting publications for curation and for extracting basic annotations from the text.	Organises and prioritises the large literature to be curated. Uses automation and data sharing platforms to amplify the curation work that can be done by the available curators.	Uses collaborative curation platforms. Deskills curation through the use of tools	Uses reviewers' feedback on publications in curation. Tools consider curation needs earlier in the publication life cycle.
Data-Based Curation	Uses ad hoc, manual procedures for identifying problems in data and fixing them.	Uses agreed-upon, documented procedures for finding and fixing problems in data.	Uses approaches for detecting errors and suggesting corrections, and the curators need to authorise the process.	Uses approaches for notifying curators if related resources have been updated.	Uses the results of curation to find patterns in curation.

Quality Assurance	Ad hoc, sporadic attempts to fix problems with curation processes.	Providing an audit trail for curation. A standard curation process is followed by all curators. Guidelines for the curation process and other related issues are documented.	Standard data representations are followed. Automated curation for data that cannot be manually curated. Clear, documented guidelines for using annotation terms.	Sharing best practice curation processes with other communities and adopting beneficial ideas.	Automatic gathering of training data from the work of existing curators. Tools to identify good curation practices from audit trails.
Data Management	Ad-hoc, data is collected by the team. Provides access to the latest version of data curated.	Uses agreed-upon approaches for collecting data, querying curated data, and checking basic quality aspects.	1. Uses approaches for allowing researchers to submit data. 2. Archives versions of data and provides access to previous versions of the data.	1. Link data to other external biomedical data. 2. Provides programmatic access to current and previous database version.	Uses automatic tools that diagnose quality problems in data, fix them and generate reports about them.
Computational Annotation	No attempt to use computational annotations or agreed terminologies.	Uses agreed-upon annotation terms within the community.	Allows users to request additional of new annotation terms. Uses a catalogue of annotations.	Uses approaches for mapping to different annotation terms.	Uses approaches for suggesting annotation terms.

Community Building	Ad-hoc procedures used for curation, with no attempt to standardise throughout the community.	Mechanisms for users to give feedback, report errors and supply requirements. Clear, documented guidelines for submitters of data, etc.	Standardises the data submission to the community, so data comes in partially curated. Proactively gathers priorities on curation from the community.	A shared and collaborative environment is provided between related biomedical communities.	1. Documents the results of the curation process and analyse them to gestate knowledge that can improve the curation process. 2. Tracks usage of data to identify curation priorities automatically.

Table 3.1: BIOC-MM, the Maturity Model for Curation of Biomedical Databases.

Chapter 4

Evaluating BIOC-MM

You only live once, but if you do it right, once is enough.

Mae West

The previous chapter discussed the creation of BIOC-MM, a maturity model for biocuration. The first version of BIOC-MM was build without formally contact with domain experts, as we focused on surveying the literature on biocuration.

Tarhan and Turetken argue that researchers have mostly focused on building maturity models, and have paid less attention to evaluating the created models [TTR16]. However, evaluating a maturity model is an important step in order to ensure that the model satisfies the goal it was designed for. The main questions that we would like to answer in this chapter are:

- How closely does the model fit the reality?
- Is the level of abstraction of the model appropriate for the task it aims to support?
- Are there any key omissions?
- Is everything that is included actually needed?
- Do the levels provide a useful guide for improvement and modelling of the current state of reality?

To answer these questions, we carried out the evaluation process using a range of methods, including contacting human experts in the domain to get their opinions and suggestions about the proposed maturity model.

The chapter begins by reviewing the literature of maturity models (Section 4.1). We then discuss how we evaluated BIOC-MM (Section 4.2). We followed two approaches for our evaluation. The first approach focuses on evaluating whether BIOC-MM contains the main practices of biocuration (Section 4.2.1). The second approach focus on evaluating the model from the perspective of experts in biocuration (Section 4.2.2). Section 4.3 presents the results of evaluating BIOC-MM.

4.1 Evaluating a Maturity Model

As mentioned earlier (chapter 3), several researchers have proposed methods for developing maturity models, such as the Institute of Internal Auditors [The13] and Bruin et al. [DBFKR05]. After development, a maturity model needs to be evaluated to check whether ot not it meets the criteria for which it is designed. There have been a number of attempts made to find ways to evaluate maturity models, such as asking domain experts for their evaluation, or distributing surveys. In the rest of this section, we examine the existing literature on evaluating maturity models.

Some researchers have focused on using different approaches to evaluate maturity models. Lee, Gwanhoo and Kwak evaluated their proposed maturity model by interviewing experts in the domain of the model [LK12]. The authors also carried out a focus group, which discussed the content of the maturity model. Zaabar et al. carried out two methods to evaluate their maturity model; semi-structured interviews and surveys [ZBP17].

Other researchers have focused on proposing evaluation approaches for maturity models. Salah et al. have introduced a template for expert reviewers to evaluate maturity models [SPC14]. By reviewing the results of Helgesson et al., the authors found that there is a lack of clear, detailed instruction showing how to evaluate maturity models. This resulted in their proposing an evaluation form that contains several questions for the assessment of maturity models. Two types of questions are included: *Likert* scales, and *open-ended*. The *Likert* scales questions cover three categories: maturity levels, process and practice, and the maturity model's understandability and usefulness. These questions are followed by ten *open-ended* questions that ask for ways to modify and improve the model, such as: Would you add any maturity levels? If so, please explain what and why?. The template has not been tested, but the authors mention that they are planning to contact domain experts in maturity models to continue their evaluation of the template.

Other researchers have focused on reviewing existing work on maturity model evaluation, such as that of Helgesson et al., who summarise ways to evaluate maturity models. The authors carried out a systematic review for evaluating maturity models [HHW12]. The study is based on 59 articles extracted from two publications databases: INSPEC and COMPENDEX, which are both provided by Elsevier Engineering Information Inc. The following keywords were used to search for articles: evaluation, maturity model, and CMM. Searching the two databases resulted in 1722 articles, but after removing unrelated and duplicated articles, 59 articles remained. The authors then sorted the 59 articles into three types based on the evaluation methods used: Type 1, the *off-line* evaluation process, where the evaluation task involves people who participated in the model creation (in this type, they need to evaluate the clarity and coherency of maturity model components and procedures); Type 2, the *expert* evaluation process, where the evaluation task involves people who are experts on the model area reviewing the model and give feedback to enhance it (in this type, the experts have to evaluate the content of maturity models and assess whether this reflects the model domain; and Type 3, which involves evaluating the model in a real-world community. The maturity model developers can perform these three types of evaluation in the order they prefer.

While reviewing the literature on maturity models, we found that the main focus was on developing maturity models and validating the process of creating these models. However, less research focused on evaluating maturity models themselves.

4.2 Evaluating BIOC-MM

As mentioned earlier, BIOC-MM was initially created based on a review of the literature on biocuration. The problem with the literature is that it does not cover all aspects of the biocuration process, focusing on literature-based curation. While creating the matrix structure of the model, we faced difficulties in filling in the cells. Some of the cells were empty after we added the information from the literature. We used our own limited expertise to fill these gaps. However, with our level of expertise, we cannot ensure that the proposed matrix structure covers the full scope of biocuration activities. Therefore, we do not know if some basic components are missing (i.e., missing columns) or if important activities are omitted from the columns we have. Furthermore, we do not know if the levels we have imposed on the activities are meaningful or useful to potential users of the model. We followed the available procedures in the literature to evaluate the model. It should be noted that BIOC-MM is the first maturity model which serves biocuration. This makes it hard to evaluate the model by comparing it to other existing maturity models, as it the only one currently available. We followed two evaluation methods, the *off-line* and *expert* evaluation (as proposed by Helgesson et al. [HHW12]), to evaluate the model and produce the final version. We chose these methods because we wanted to evaluate the model from two different perspectives. In the following subsections, we present how we performed each evaluation method and the outcome of both (see the *off-line* (Section 4.2.1) and *expert* (Section 4.2.2).

4.2.1 Off-line Evaluation

We wanted to perform a preliminary evaluation before asking experts to evaluate the model, because we do not have access to unlimited time from experts. We aimed, therefore, to improve some parts of the model so that experts can then focus their efforts on other problems in the model.

To perform the *off-line* evaluation, we checked a number of aspects of the maturity model. Initially, we revised the model components and checked whether or not they covered the main procedures of biocuration. We then checked whether activities and practises were coherent.

In the earliest version of the model, as shown in appendix B, we set the goal of the maturity level as showing how the biocuration process move from manual to automatic curation. This was set because most of the reviewed literature dealt with automating the biocuration process. For the same reason, the model components focused on curating based on the literature.

We started by evaluating whether or not the model covered the full cycle of the curation process. The model components focused on two aspects; literature curation and documenting the curation results. We added other components to complete the cycle of the curation process. Changing the model components lead us to change the cell contents as well as the maturity level aims. After these changes, we checked the content of each column and row to ensure that they met the component and level aims. The output of this evaluation method is shown in Appendix C.

4.2.2 Expert Evaluation

We needed people who were working as, or have worked as, curators to evaluate BIOC-MM. It was important to have curators' opinions, as they are experts at finding defects in data and correcting them. Besides, curators' feedback is vital for ensuring BIOC-MM meets its intended goals. According to Helgesson et al., *expert evaluation* "is conducted by involving practitioners, who are the experts on the type of process that is intended to be improved by the maturity model, but who have not been involved in the actual development of the maturity model" [HHW12, p. 439]. The *expert evaluation* is our focus in this section. We undertook two different expert evaluations. First, we sought unstructured feedback. Second, we ran a semi-structured interview with a domain expert. The rest of this section explains how both evaluation methods work.

4.2.3 The Unstructured Feedback

In the first stage of expert evaluation of BIOC-MM, we used unstructured feedback. The aim of this evaluation was to test the model before the semi-structured interviews. At this stage, we wanted the experts to point out only obvious errors. To achieve this, we set up a cheaper, more informal type of feedback gathering. We presented the maturity model in a poster, which resulted from the *off-line* evaluation (refer to Appendix C for BIOC-MM). The poster format was a working form of the model that explicitly invited feedback. To get feedback, we provided the poster in three locations: the University of Manchester E-Science lab, the Evolution and Genomic Sciences department, and the International Conference of Biomedical Ontologies (ICBO 2017). Each location targeted a different type of expert:

- Bioinformatics professionals, who support the owners of and users of curated data (including curators).
- Scientists, who rely on curated data sources to do their own work.
- A mixture of curators and consumers of curated data.

We received feedback from four people at ICBO 2017, and we did not receive any feedback from the other locations. The comments focused on adding or modifying the components. We received a number of comments suggesting that we consider computational annotation as a stand-alone component rather than including it within another component. Other comments suggested separating the tasks in the data-based component into two different components, as the tasks seem to have two diverse aspects. The comments we received resulted in some changes in the model; it changed from how it appears in Appendix B to the version that appears in Table 3.1.

4.2.4 The Semi-Structured Interview

The second expert evaluation was carried out through a semi-structured interview. Our aim with this stage of evaluation was to get experts' opinions of BIOC-MM, and establish whether it contains all the main activities of biocuration. We also aimed to test the understandability and usefulness of the model. To achieve the evaluation aims, we conducted a semi-structured interview. We chose this type of interview because we anticipated that we might need to adjust the questions based on the interviewee's answers. The interview was expected to take no longer than an hour. As mentioned earlier, Salah et al. have proposed a survey template to evaluate maturity models [SPC14]. We adjusted some of the proposed questions to meet the requirements of our evaluation. The questions we used can be found in Appendix A. The interview questions focus on assessing three aspects:

1. Whether or not BIOC-MM reflects the main aspects of the biocuration process.
2. Whether the model needs to be modified or enhanced.
3. The experts' opinions on whether the model would be useful for biocuration communities.

When interviewing an expert, we began by explaining BIOC-MM (i.e., What is it? What are the components? What are the maturity levels? What can we use the model for?) We then discussed the model and the interview questions. We took notes to record the interviewees' answers.

Although we invited ten experts in biocuration to participate in validating BIOC-MM, we only got a response from one person. We interviewed a senior scientific research manager who worked as a biocurator. At the beginning of the interview, we explained BIOC-MM to the interviewee. According to the interviewee's answers, BIOC-MM covers all the aspects of the biocuration process. However, a couple of refinements were proposed to improve the model. When the interviewee was looking through the model components, he focused on the *computational annotation* component at two levels. At level 3, the component allows users to request annotation terms. The interviewee suggested not to make the request related to annotation, use by a community, as the community can relate term requesting to their records. In the community he worked in, they referred to it as records. In addition, he suggested modifying the annotation task at level 5 to use approaches to suggest modifications for records that related to computational annotation. The interviewee was positive regarding the usefulness of the model and specifically mentioned that BIOC-MM could be useful in terms of allowing communities to assess the maturity of their process (which relates closely to our initial goal in creating the model). Moreover, interviewee said that the usefulness of the model can be extended to cover other aspects, as it offers a good source of resources for funds. This would help a community to find sustainable funders by presenting a clear plan of the steps needed to be in the required mature level.

4.3 Conclusion

We presented two different approaches for evaluating BIOC-MM. The first evaluation involved assessing the model by the people who created it, after which we evaluated it with the help domain experts. Based on the evaluation process followed, we found that the first version of the model was missing a number of key procedures in the biocuration process. We modified the model to incorporate the missing procedures, and presents the second version of the model for expert evaluation. Based on expert feedback, we edited the model to fits the reality of biocuration.

Returning to the questions raised at the beginning of this chapter, the evaluation process enabled us to ensure that the model reflects the biocuration process. Furthermore, we included the missing procedures which were mentioned during evaluation. In addition, the model contains a clear description of each level and component provided. As part of the evaluation process, some researchers propose comparing the maturity model with other existing models. However, as we mentioned earlier, BIOC-MM is the first maturity model in biocuration, which makes it difficult for us to compare it with other similar models.

Chapter 5

Inferring Defects, Corrections from Curation Changes

Happiness doesn't result from what we get, but from what we give.

Ben Carson

Many researchers have focused on raising data quality by proposing and using approaches to detect quality problems and fix them, such as Wang et al. [WS96] and Pipino et al. [PLW02]. However, the process of raising data quality cannot be fully automated, as some quality problems require expert knowledge in order to be resolved. As an example, imagine that a new publication causes a change to be required for the annotation of a protein entry. Some protein information can be annotated using tools. However, other protein information is complex and cannot be found and fixed easily. This requires human experts in the domain of the data to do the job. These experts may volunteer their time to find defects and correct them in the area of their interest, or data source providers can hire curators. It should be noted that new information is constantly coming in so fast that dealing with it is not an easy job, even for well-resourced databases, and the process requires a great deal of time.

In the previous chapter, we looked at how the curation process can be improved to gain the most curation benefit from the resources available for curation. In this chapter, we focus on proposing a way to amplify the value of the time spent on curation and package this to be used by owners of data resources with overlapping content. Crucially, we would like to find an automatic approach to extract the curation information without the need to ask curators to make extra effort. Specifically, we do not want to ask them to do additional work to package the curation knowledge manually, or to add metadata describing it. Furthermore, we do not want them to change the toolkit they use for curation, or to use any additional tools. To achieve this, we need to answer the following questions:

1. Can we infer the defects detected in data by a curator by examining the changes made to the data by curators between stable versions of the data resource?

2. Can we also infer the corrections identified for the defects by the curators from changes made between stable versions?

In this chapter, we will answer the questions above in abstract form. In the next chapter 6, we will present more concrete answers in the context of a specific curated source. We will tackle the problem of finding modifications made to curated databases and converting these changes into more widely applicable data defects and corrections. One option would be to ask curators to identify and record such defects explicitly while carrying out their curation work. However, this means extra work and time commitment from curators, which we would like to avoid. Although data curators have the expertise to spot semantic data defects, our aim in this thesis is to maximise the usefulness of curation efforts with the least amount of additional input from curators. Another option would be to ask database providers to periodically notify all their users about the changes made by curators. However, this will mean an extra expense for database providers, which we also would like to avoid in order to minimise costs.

The chapter begins with some examples to highlight the problems that we focused on in the chapter (Section 5.1). We also provide some definitions related to defects and give a brief overview of the literature regarding data defects (Section 5.2). We then explain how to achieve maximisation of curation efforts through the use of a third-party component, IQBot (Section 5.3). As part of IQBot's job is to extract changes between data sources versions, we explain the meaning of the concepts related to changes (Section 5.4). We then explain how to infer defects and defect corrections from a monitored data source (Section 5.5). Furthermore, we discuss extraction of the reason behind the change detected in the monitored data source (Section 5.6). Finally, we conclude with a way to infer defects (Section 5.7).

5.1 Motivating Example

There are many examples which demonstrate the need to find a mechanism to infer defects and corrections. Embury et al. gave an example of how a curator curates a protein entry in UniProt[1] [EJSE14]. In this example, the curator looked at a protein entry with the code ”P322169”, and spotted a defect in data related to the number of pages where protein information is published. Then, the curator changed it to the correct page number.

Another example is that many biomedical databases provide Gene Ontology[2] (GO) terms in their data. GO terms offer a collection of gene vocabularies and concepts to aid in defining the relationship between these concepts [BCA$^+$00]. Each GO term has a unique meaning, such as the term ”GO:0005829” which means that ”the part of the cytoplasm that does not contain organelles but which does contain other particulate matter, such as protein complexes”[3]. When the source ontology changes, users of GO terms need to apply these changes to their data. Unfortunately, the source ontology does not provide a service to notify its users about such changes in terms. Therefore, it is the responsibility of data curators to find the GO terms in their data and replace them with new versions.

Curation is not limited to changing data because of data was entered incorrectly. Curators need to search for changes in their original sources and check their own data to see if they can find the presence of defects. Other examples can be found in Embury et al. [EJSE14].

5.2 Background and Literature Survey

In this section, we present definitions of some information quality concepts related to this chapter. Information quality refers to errors in data that affect its quality, such as as a data defect. To overcome a defect in data, we need a data correction. Data correction is the act of removing *defects* in data. In simple terms, a data defect is where an error is found, while data correction is the way to fix it.

In terms of information quality, data defects and their corrections are categorised to fit into one of the quality dimensions. There are many information quality dimensions which cover different aspects of defect correction, such as completeness, consistency,

[1] www.uniprot.org

[2] www.geneontology.org

[3] www.ebi.ac.uk/QuickGO/term/GO:0005829

accuracy, precision, validity, and timeliness.

Much research has been conducted into the management of defects in data. Dallachiesa et al., for example, propose NADEEF, a commodity data cleaning system [DEE$^+$13]. The system allows users to choose the quality rules to apply to data and add algorithms to solve quality problems. The users' input will be taken into account in the *core* component, which is responsible for compiling the input to find defects in data and fix them accordingly. Greets et al. propose LLUNATIC, a data cleaning framework [GMPS13] which focuses on solving three quality problems: missing semantics, missing repair algorithm, and main memory implementations and scalability. To overcome these problems, the authors use equality-generating dependencies (EGDS) to produce a language that concerns about dependencies in constraints. They also build a semantic which focuses on repairing cell groups rather than an individual cell, and propose an algorithm which works on the solution. Fan et al. suggest a data cleaning method which focuses on monitoring and enhancing data [FLM$^+$12]. The method focuses on targeting input tuples, and the authors have developed a graph-based algorithm to divide the tuples by reigns. The authors suggested that using reigns helps to improve data correctness. In a further study, Fan et al. propose a framework based on repairing and matching [FMTY14]. The designed algorithm was built based on matching rules and integrity constraints.

5.3 Overview of IQBot Architecture

Our key aim is to find a way to benefit from curation efforts without requiring additional work from curators. Embury et al. propose the idea of IQBot, a mechanism that monitors a curated database to find defects and defect corrections and prepares them for sharing [EJSE14]. The authors provide IQBot architecture, but this has not yet been implemented by them. This section explains how IQBot works with other components to perform its job. Figure 5.1 gives a general view of this approach. The left side of the figure shows the curated databases that we will monitor for data defects. An IQBot monitors the curated databases to find defects and corrections. Then, the results of the process are moved to a Defect Corrections Server to be stored in the Defect Store. The Defect Corrections Server is responsible for controlling the access to the Defect Store, where the changes found in the monitored database are stored. The IQBot user can use one or more Defect Corrections Servers based on the data collected from the curated databases. Then, the changes found can be packaged and published in a more applicable format.

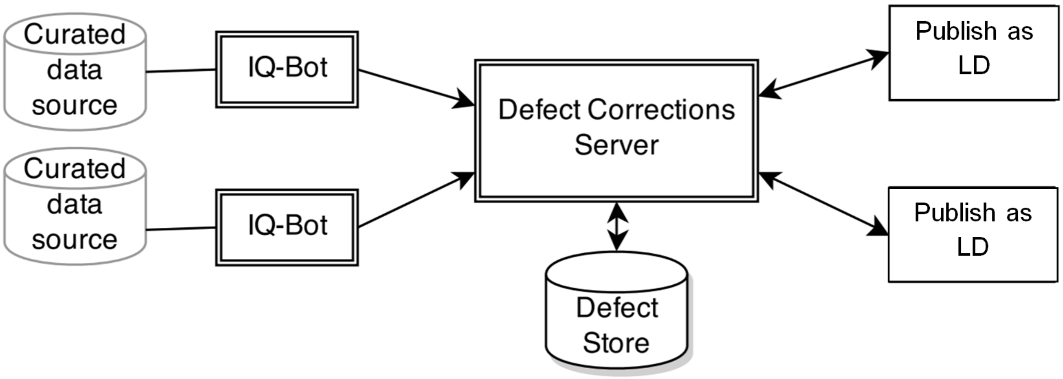


Figure 5.1: The IQBot architecture as proposed in [EJSE14].

The process behind the architecture flow is complex for several reasons. First, it is not easy to monitor a database, as not all databases have the same procedure for accessing their data. Second, not all databases use the same data representation, which is required to define the way of extracting and retrieving data. Third, packaging defects and defect corrections to suit many databases is complicated, and so we are considering providing data as linked open data.

Our focus in this thesis is the IQBot component and how it monitors a curated database to detect defects and corrections in data. An IQBot component follows five steps. First, the IQBot monitors the curated database to extract changes in data. To do this, we need to extract the data from two consecutive versions of the curated database, so that we can find changes that occur from one version of data to the next. This will assist us in keeping records of the history of changes for later use. The data extracted from both versions of the curated database is then compared. If the data are different in both versions, the IQBot will move to the third step, where it marks one version of the data as a defect and the next version as a defect correction. However, if the data are the same, the IQBot will go back to the first step and move on to the next data record. In the fourth step, the IQBot determines the relationship between the detected defect and correction by identifying the type of change. The last step involves finding out why the data has changed. The reason for the change is calculated based on information extracted in the previous steps. We believe that providing the reason for change will help IQBot users to make an informed choice as to whether to apply the change to their data or not, based on the information about the reason for the change.

5.4 Extracting Deltas from Monitored Sources

Here, we consider changes made to a curated database during the process of curating it. We refer to these changes as *delta*. Delta is a Greek letter, which, in the context of databases, means changes in which data records are updated from one version to another. IQ-complete delta in the first version are either left entirely uncorrected in the second version, or are entirely corrected in the second version. In this thesis, we not only look at finding changes in data, but also focus on inferring changes in the form of defects and defect corrections.

The method for obtaining delta information varies depending on whether a resource is versioned or not. Embury et al. discuss the issue of versioning in linked data resources [EJSE14]. The authors mention that some database owners provide access to older versions; alternatively, versioning tools can be used, such as Apache Marmotta [4]. However, in cases where database owners do not provide older versions, an IQBot can record regular snapshots of the database. An IQBot can then use these snapshots to compare data versions and extract IQ-complete delta, which is discussed in the next section.

5.5 Inferring Defects from IQ-Complete Deltas

In the remainder of this chapter, we describe the work we have done to convert the IQBot concept into a working prototype. The algorithm 1 shows an abstract view of the process IQBot follows to perform its job. IQBot starts by establishing whether or not a curated source has been monitored before. If it is being monitored for the first time, IQBot will start from the first version of the data source. Otherwise, it will start with the mocst recent version which has not yet been monitored. For the selected version, IQBot retrieves all record ID. It then compares data records from two consecutive versions of the data to extract changes. When comparing data, IQBot checks if the record ID is newly added or deleted from the previous version. If the ID record exists in both versions, IQBot compares the values of a specific attribute to check if the values are different (it follows the same process if it is monitoring more than one attribute). Then, if the values are not the same, IQBot sets the attribute value in $version_n$ as a defect and sets the attribute value in $version_{n+1}$ as a correction. When IQBot detects a defect and its correction, it will send them to a function to find the

[4] www.marmotta.apache.org

relationship between them, which we refer to as the type of change. After finding the type of change, IQBot identifies the reason behind the change and adds all the extracted data to a list of defects. The process of finding defects and their related information is repeated for all record ID. Finally, IQBot sets the version number of the data source as monitored.

Algorithm 1 The general algorithm of the IQBot

```
Require:
    input: “ver” the version of the data set to find defects
    input: “ver+1” the version of the data set to find corrections
    output: a list of defects and defect corrections
if the source is monitored for the first time then
    ver = the first version of the monitored source
else
    ver = the last monitored version of the data source
end if
while there are unmonitored versions do
    recID = get all records’ ID
    for each recID do
        if the record ID is not found in ver-n then
            The record is newly added
        else
            if the record ID is not found in ver-n+1 then
                the record is deleted
            end if
        end if
        if data record changed in ver-n+1 compare to ver-n then
            defect = data in ver-n
            correction = data in ver-n+1
            type = findDefectType(defect, data in ver-n+1)
            reason = findTheReasonForTheChange(type, defect, correction)
            add the tuple to list of defects(recID, defect, correction, type, reason)
        end if
    end for
    add the version number to a list of monitored versions
end while
```

When we tried to produce a generic algorithm to infer all defects, we faced issues covering all information quality dimensions. Because information quality is a complex and rich concept, it takes many forms. Information quality dimensions are used to get a handle on these forms and to propose solutions that cover the major cases, even if a general solution that works for all cannot be found.

Since it is difficult to find a generic algorithm to detect all defects, we focused on producing an algorithm which loops through all record updates in the table under consideration. So, in our proposed algorithm, we take each record version pair, call a function specific to each of the dimension covered, and see what defects come back.

To keep things simple, the algorithm should keep track of all the defects that are returned, and combine them into a single list of defects to be published for that entry. Each defect will be packaged in a way that describes the type of defect, so there is no problem with lumping all the defects from different dimensions into a big bag together. The remainder of the chapter presents the process of how to find the type and reason for the changes.

5.5.1 Finding Types of Defects

As mentioned previously, a tuple containing the inferred defect and its correction is extracted from the monitored database. The next step is to determine the type of defect detected in the previous phase. We selected four information quality dimensions: accuracy, precision, completeness, and consistency. We considered these four information quality dimensions to categories the type of defect because most of the defects we focus on fit into these dimensions. However, the right dimension cannot be identified without looking into the text value of the defect and its correction in order to compare and find the relationship in between them.

The following sections present each selected quality dimension in more detail. For each dimension we provide the definition, followed by examples. We also present an algorithm which shows the process of identifying the defect type at an abstract level.

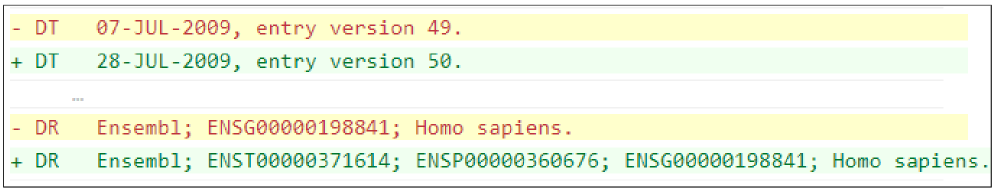
5.5.1.1 Completeness

Completeness is defined as “the state or condition of having all the necessary or appropriate parts” [5]. In simple terms, the completeness dimension, in the context of information quality, means the action of adding missing values. Changes to data that fill in missing values can point to the prior existence of completeness defects. Identifying completeness defects is simpler than other types, as it does not require analysis of the values of the defect and correction. A simple manifestation of a completeness defect and correction occurs when an attribute has a null value in $version_n$ but has a

[5] https://en.oxforddictionaries.com/definition/completeness

value in the following version. As an example of completeness, imagine that important information, such as a patient's age, is missing from their patient record. This is considered a defect where the value is missing. To correct this defect, the patient's age should be added.

We should mention that there is another type of incompleteness where a missing value is a multivalued attribute. For example, imagine IQBot is monitoring changes in an attribute, and the attribute is a list. When IQBot compares the two versions of the list, it finds that one or more elements of the list are missing. Figure 5.2 shows the "Q96EK9" protein entry in version 49 compared with its form in version 50. It can be noticed from the figure that the DR line was not previously present, and has been added to the protein entry.



```
- DT   07-JUL-2009, entry version 49.
+ DT   28-JUL-2009, entry version 50.
     ...
- DR   Ensembl; ENSG00000198841; Homo sapiens.
+ DR   Ensembl; ENST00000371614; ENSP00000360676; ENSG00000198841; Homo sapiens.
```

Figure 5.2: Q96EK9 protein entry version 49 in compare to version 50 - completeness example.

When IQBot extracts a paired defect and correction, it sends the pair to function to find the defect type. The algorithm 2 shows the part of the type function related to completeness. The algorithm checks whether the defect contains a null value(S) and whether the correction contains the missing value(s).

Algorithm 2 The completeness algorithm

```
Require:
    input: defect and its correction
    output: the type of defect
if (defect = null AND $correction \neq null$) OR (multivalued defect has null value(s)
    AND multivalued correction has value(s)) then
    set the type of defect as completeness
end if
```

5.5.1.2 Accuracy

Accuracy is "the degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard" [6]. In term of data quality, accuracy

[6] www.oxforddictionaries.com/definition/accuracy

is the degree to which data is correctly allocated. An example of a data accuracy issue is when a user enters an incorrect spelling for the city name in an address. Another example is when the user enters the current date instead of entering a date of birth.

Accuracy covers situations where data are wrong, either partially or completely. By partially wrong, we mean cases where a minor change occurs to data, such as fixing spelling mistakes and changing data that is partially incorrect (for example, entering '2.02' instead of '2.20'). By completely wrong, we mean cases when the defect is entered entirely incorrectly (for example, entering the family name instead of the first name).

To identify defects with accuracy problems, IQBot sends a pair of defect and correction to find defect type function (algorithm 1). As shown in the algorithm 3, the defect and correction should not contain a null value. When comparing the defect to its correction, the algorithm compares character-by-character. If some or all characters are changed, this can be considered as inaccuracy.

Algorithm 3 The accuracy algorithm

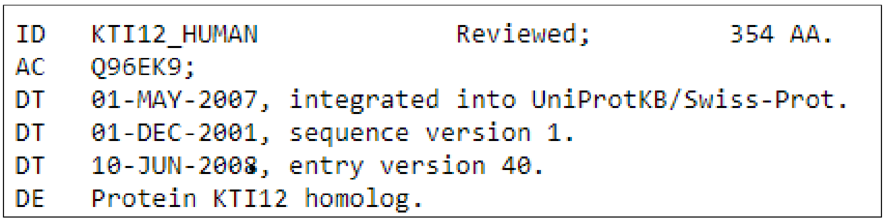
```
Require:
    input: defect and its correction
    output: the type of defect
if $defect \neq null$ AND $correction \neq null$ then
    Compare each character in defect with each character in correction
    if number of characters changed $> 0$ then
        set the type of defect as accuracy
    end if
end if
```

5.5.1.3 Consistency

Consistency is when contradictions appear in data, normally in the same version of the data. In other words, consistency is the act of ensuring data uniformity. For example, if the current age entered in a 2019 record is 30, and the year of birth is 1982, this presents an inconsistency problem. IQBot focuses on comparing two versions of the data, and therefore ensuring consistency here involves checking for inconsistencies not in the data values but in other aspects, such as ensuring that data records follow the same data format. For instance, imagine that a data source uses an abbreviation to present some data, and then later changes the abbreviation to the full term. Another example is changing the letter case from upper to lower case and *viceversa*, or changing the use of

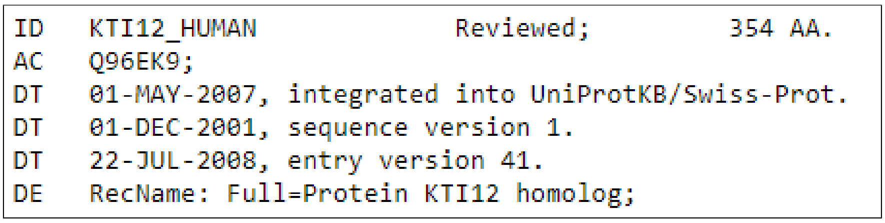
Greek letters to another numeric format. The reason for making these changes may be to follow the standard issued by the consortium of the curated source, which manages the data records. We classify these types of defects as a consistency issue.

Figures 5.3 and 5.4, for example, show two different versions of the protein entry with the code "Q96EK9": versions 40 and 41. In these two versions, the data in the DE line has the same protein name "Protein KTI12 homolog". However, the difference between the two versions is that the community started to use a specific term in front of the protein name, which appears in version 41 as "RecName". This is an example of changing the schema of the data. We consider this type of defect as a consistency problem, as the value remains the same in both versions, but the difference relates to standardising the data in the records.



```
ID   KTI12_HUMAN             Reviewed;         354 AA.
AC   Q96EK9;
DT   01-MAY-2007, integrated into UniProtKB/Swiss-Prot.
DT   01-DEC-2001, sequence version 1.
DT   10-JUN-2008, entry version 40.
DE   Protein KTI12 homolog.
```

Figure 5.3: Q96EK9 protein entry - version 40

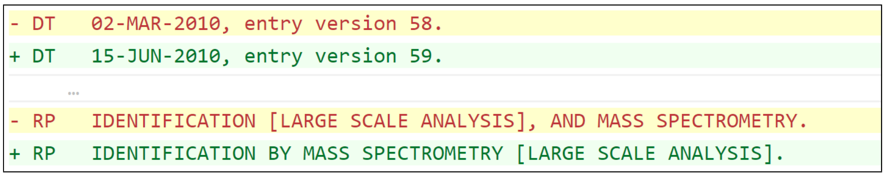


```
ID   KTI12_HUMAN             Reviewed;         354 AA.
AC   Q96EK9;
DT   01-MAY-2007, integrated into UniProtKB/Swiss-Prot.
DT   01-DEC-2001, sequence version 1.
DT   22-JUL-2008, entry version 41.
DE   RecName: Full=Protein KTI12 homolog;
```

Figure 5.4: Q96EK9 protein entry - version 41

For the same protein entry, Q96EK9, Figure 5.5 shows how the order of words can change; in this example, we consider reordering the words to be a consistency defect. Similar types of changes, which focus on adding modifications and editing the text rather than changing the actual value, are considered to be a consistency problem.

The algorithm 4 gives an overview of how to identify consistency defects. First, it ensures that defect and correction values are not null. Then, it checks two conditions. First, it checks that the defect value does not reflect the correct value when it compares to other values in the same record (as with the age-related example mentioned earlier in this section). Second, it checks if the actual value of the defect and correction are the same. Then, the defect between them is related to consortium convention (with the



```
- DT   02-MAR-2010, entry version 58.
+ DT   15-JUN-2010, entry version 59.
     ...
- RP   IDENTIFICATION [LARGE SCALE ANALYSIS], AND MASS SPECTROMETRY.
+ RP   IDENTIFICATION BY MASS SPECTROMETRY [LARGE SCALE ANALYSIS].
```

Figure 5.5: Q96EK9 protein entry - version 58 in compare to version 59.

changing letter case example mentioned earlier). If one of these conditions satisfied, then the type of defect is set as one of consistency.

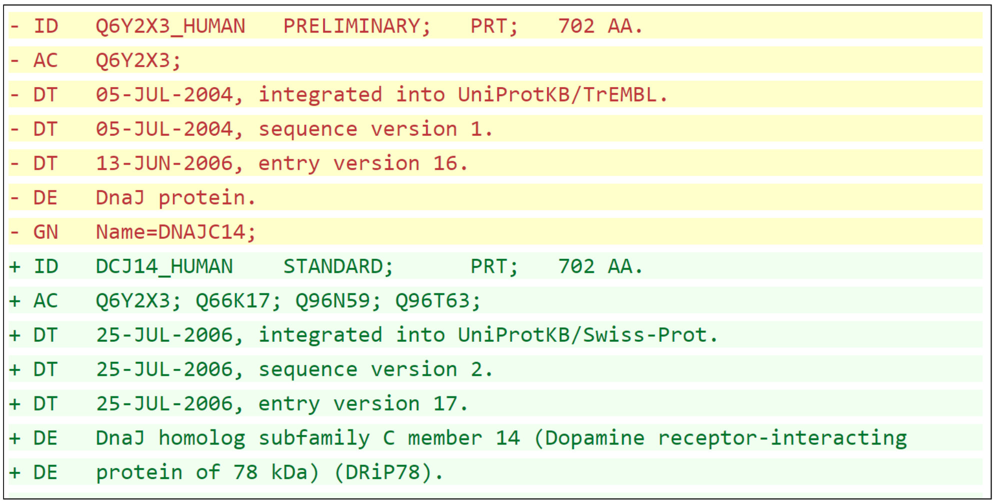
Algorithm 4 The consistency algorithm

```
Require:
    input: defect and its correction
    output: the type of defect
if $defect \neq null$ AND $correction \neq null$ then
    if defect value contradicts with values in the same record then
        set the type of defect as consistency
    else
        if defect value = correction value then
            set the type of defect as consistency
        end if
    end if
end if
```

5.5.1.4 Precision

The definition of precision is “the quality, condition, or fact of being exact and accurate” [7]. Precision, in the context of IQBot, is not related to whether or not data is presented incorrectly. Instead, it is concerned with providing data in more depth. In simple terms, post-correction data is more precise than data before editing. For example, imagine that a data source uses ontology terms. The ontology terms can have multiple levels, so after referring to the top level term, we may then edit it to a more specific sub-level term. In this case, we refer to this action as precision. Figure 5.6, for example, shows part of the changes made between versions 16 and 17 of the protein entry with accession “Q6Y2X3”. Looking at the lines beginning with “DE”, in version 16, the value stored is “DnaJ protein” while in version 17 it is modified to “DnaJ homolog subfamily C member 14”.

[7] www.oxforddictionaries.com/definition/precision



```
- ID   Q6Y2X3_HUMAN   PRELIMINARY;   PRT;   702 AA.
- AC   Q6Y2X3;
- DT   05-JUL-2004, integrated into UniProtKB/TrEMBL.
- DT   05-JUL-2004, sequence version 1.
- DT   13-JUN-2006, entry version 16.
- DE   DnaJ protein.
- GN   Name=DNAJC14;
+ ID   DCJ14_HUMAN    STANDARD;      PRT;   702 AA.
+ AC   Q6Y2X3; Q66K17; Q96N59; Q96T63;
+ DT   25-JUL-2006, integrated into UniProtKB/Swiss-Prot.
+ DT   25-JUL-2006, sequence version 2.
+ DT   25-JUL-2006, entry version 17.
+ DE   DnaJ homolog subfamily C member 14 (Dopamine receptor-interacting
+ DE   protein of 78 kDa) (DRiP78).
```

Figure 5.6: Q6Y2X3 protein entry versions 16 and 17 - precision example.

As with the previous dimensions, we use the defects list to determine the type of defect. To identify precision, the algorithm 5 shows that the pair of defect and correction has no null value. After that, the algorithm checks the value of the correction and determine whether or not it is a subset of the defect.

Algorithm 5 The precision algorithm

```
Require:
    input: defect and its correction
    output: the type of defect
if $defect \neq null$ AND $correction \neq null$ then
    if correction is a subset of defect then
        set the type of defect as precision
    end if
end if
```

5.6 Determine the Reason for the Changes

As well as the primary results of defects and corrections, an IQBot can also provide some metadata relating to the detected results. This metadata consists of information about the reason behind the changes. Providing the reason for changes helps other resources to decide whether or not they want to apply the corrections to their data.

However, determining the reason for changes is quite difficult in most cases. This is because different communities follow different procedures for presenting the reason for the change. Besides, not all curated sources provide evidence referring to these reasons. In these cases, we need to use the available data in data records, where the defects and defect corrections occur, without requiring curators in the monitored source to do extra work.

5.7 Conclusion

We have attempted to maximise the value of curation efforts, allowing results to be reused by the owners of other related data sources. The owners of some databases are able to hire curators, who are experts in their domain, to curate their data and ensure a high level of data quality. The owners of other resources cannot afford this. Besides, as curators may make changes to databases on a regular basis, it can be hard for database consumers who use data from a monitored source to stay informed of the changes applied to the monitored source. In this chapter, we focused on answering the questions raised at the start of the chapter. We asked whether it was possible to infer the defects detected in data by a curator by examining the changes made to the data by curators between stable versions of the data resource. Also, we asked if we can infer the corrections identified for the defects by the curators from changes made between stable versions. To answer these questions, we introduced the idea of IQBot. We presented an algorithm which crawls data in curated source versions to extract changes. The algorithm is designed to infer defects and corrections from the extracted changes. The algorithm we provided shows a general way of doing this, but when we apply IQBot to a specific data source, we will adjust it to serve the domain of the data source.

Our approach in this chapter has been from the perspective of achieving the goals of the IQBot. We suggested that the curated data sources could adopt ways to add an audit trail while they are curating data, as this may help others who wish to make use of these efforts. In other words, applying an audit trail to data will give users of the data an understanding of the environment in which these changes have been made, and may also enhance the community.

Chapter 6

IQBot in Practice

Nothing is yours. It is to use. It is to share. If you will not share it, you cannot use it.

Ursula K. Le Guin

In the previous chapter, we described the IQBot concept and explained how an IQBot can be used to maximise the value that can be obtained from the curation work carried out on a database.

In this chapter, we want to find out whether the abstract IQBot concept presented previously can work in practice. We focus on answering the following questions: can IQBot find meaningful defects and corrections from changes to a curated database? Can the owners/users of other databases benefit from the defects we find? This chapter considers the use of the IQBot with a real-world curated database. Since the focus of this thesis is on scientific data, we chose a well-known curated biomedical data source as the basis of our work; the UniProtKB database. This chapter describes the step-by-step process of applying the IQBot concept to UniProt, and indicates the difficulties and issues faced.

The structure of this chapter is as follows: We provide background information about UniProt, and the reasons for choosing it in the context of this study (Section 6.1). Section 6.2 discusses the process of connecting an IQBot to UniProtKB. Section 6.3 explains how the IQBot extracts defects and corrections from UniProtKB. We then describe how we identified the reasons for changes (Section 6.4). In Section 6.5, we discuss the defects and corrections produced by observing IQBot in action in conjunction with UniProtKB, and show the importance of reusing and sharing curation efforts.

6.1 The Target Source: UniProt

UniProt[1], also known as the Universal Protein resource, consists of three databases: UniProt Knowledgebase (UniProtKB); UniProt Archive (UniParc); and UniProt Reference Clusters (UniRef). UniProtKB has two parts: Swiss-Prot (manually annotated by human experts in the domain); and TrEMBL (automatically annotated by tools). UniParc provides an archive of protein sequences. UniRef "provides clustered sets of sequences from the UniProt Knowledgebase and selected UniParc records to obtain complete coverage of the sequence space at several resolutions while hiding redundant sequences from view"[2].

The UniProt consortium is one of the leading communities in the biomedical field, and receives support from various organisations, including the European Bioinformatics Institute (EMBL-EBI) [3], the SIB Swiss Institute of Bioinformatics [4], and the Protein Information Resource (PIR) [5]. Our focus in this chapter is on UniProtKB data, as it satisfies the criteria needed to test IQBot with a monitored curated database:

- It contains a huge amount of data to (potentially) monitor, as it has 111,982,257 protein entries.
- It is manually curated by dedicated domain experts.
- The data is curated on a regular basis (every four weeks), which makes UniProtKB a rich source of updates.
- It grants public programmatic access to current and previous versions of data.

UniProtKB contains protein related information in the form of 'entries'. Each protein entry includes information such as the protein ID, the accession number, the protein name, and the date of creation. Protein entries are available in a .txt file format. Figure 6.1 shows an example UniProtKB protein entry. We can see that the first line of the entry starts with 'ID', and the delimiter symbol '//' indicates the end of the entry.

[1] www.uniprot.org

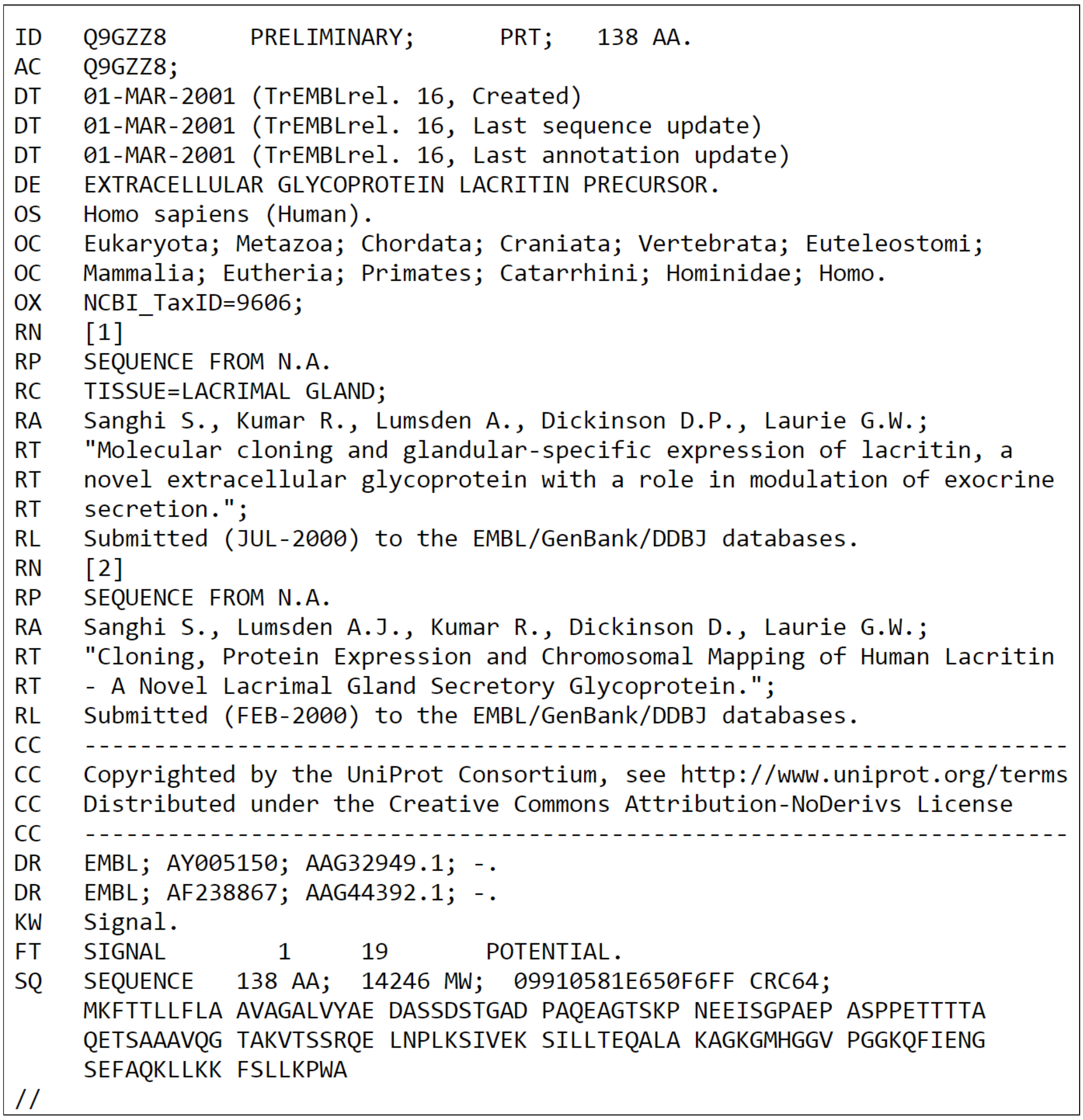
[2] www.uniprot.org/help/uniref

[3] www.ebi.ac.uk

[4] www.sib.swiss

[5] https://pir.georgetown.edu/

Each line starts with a key code, which consists of two letters referring to the type of information presented in the line. For example, the lines starting with ’AC’ contain the protein accession number, and lines starting with ’CC’ contain comments from curators. Table 6.1 presents the meaning of the line code initials used in UniProtKB entries (it should be noted that not all of this information is presented in the example entry, as some is optional).



```
ID   Q9GZZ8      PRELIMINARY;      PRT;   138 AA.
AC   Q9GZZ8;
DT   01-MAR-2001 (TrEMBLrel. 16, Created)
DT   01-MAR-2001 (TrEMBLrel. 16, Last sequence update)
DT   01-MAR-2001 (TrEMBLrel. 16, Last annotation update)
DE   EXTRACELLULAR GLYCOPROTEIN LACRITIN PRECURSOR.
OS   Homo sapiens (Human).
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
OC   Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
OX   NCBI_TaxID=9606;
RN   [1]
RP   SEQUENCE FROM N.A.
RC   TISSUE=LACRIMAL GLAND;
RA   Sanghi S., Kumar R., Lumsden A., Dickinson D.P., Laurie G.W.;
RT   "Molecular cloning and glandular-specific expression of lacritin, a
RT   novel extracellular glycoprotein with a role in modulation of exocrine
RT   secretion.";
RL   Submitted (JUL-2000) to the EMBL/GenBank/DDBJ databases.
RN   [2]
RP   SEQUENCE FROM N.A.
RA   Sanghi S., Lumsden A.J., Kumar R., Dickinson D., Laurie G.W.;
RT   "Cloning, Protein Expression and Chromosomal Mapping of Human Lacritin
RT   - A Novel Lacrimal Gland Secretory Glycoprotein.";
RL   Submitted (FEB-2000) to the EMBL/GenBank/DDBJ databases.
CC   -----------------------------------------------------------------------
CC   Copyrighted by the UniProt Consortium, see http://www.uniprot.org/terms
CC   Distributed under the Creative Commons Attribution-NoDerivs License
CC   -----------------------------------------------------------------------
DR   EMBL; AY005150; AAG32949.1; -.
DR   EMBL; AF238867; AAG44392.1; -.
KW   Signal.
FT   SIGNAL        1     19       POTENTIAL.
SQ   SEQUENCE   138 AA;  14246 MW;  09910581E650F6FF CRC64;
     MKFTTLLFLA AVAGALVYAE DASSDSTGAD PAQEAGTSKP NEEISGPAEP ASPPETTTTA
     QETSAAAVQG TAKVTSSRQE LNPLKSIVEK SILLTEQALA KAGKGMHGGV PGGKQFIENG
     SEFAQKLLKK FSLLKPWA
//
```

Figure 6.1: An example of a protein entry in UniProtKB.

Table 6.1 The meaning of line code in UniProtKB taken from [Con14a].

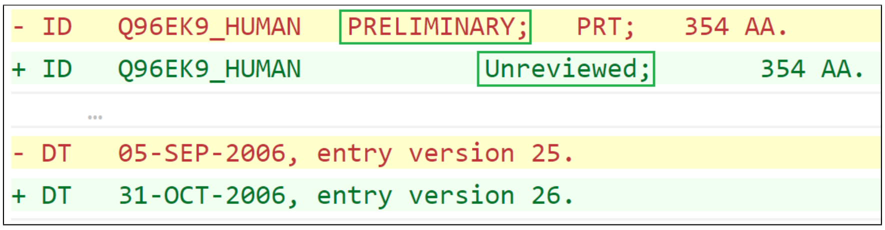
Line Code	**Content**	**Occurrence in an entry**
ID	Identification and type of database	Once; starts the entry
AC	Accession number	Once or more
DT	Date	Three times
DE	Protein name(s)	Once or more
GN	Gene name	Optional
OS	Organism species	Once or more
OG	Organelle	Optional
OC	Organism classification	Once or more
OX	Taxonomy cross-reference	Once
OH	Organism host	Optional
RN	Reference number	Once or more
RL	Reference location	Once or more
RC	Reference comment(s)	Optional
RX	Reference cross-reference(s)	Optional
RG	Reference group	Once or more (Optional if RA line)
RA	Reference authors	Once or more (Optional if RG line)
RT	Reference title	Optional
RL	Reference location	Once or more
CC	Comments or notes	Optional
DR	Database cross-references	Optional
PE	Protein existence	Once
KW	Keywords	Optional
FT	Feature table data	Once or more in Swiss-Prot, optional in TrEMBL
SQ	Sequence header	Once
(blanks)	Sequence data	Once or more
//	End of entry	Once; ends the entry

6.1.1 Difficulties in Accessing UniProt

UniProt provides an API (called 'UniProtAPI') to access its resources [6]. However, the use of UniProtAPI has some limitations for our purposes, because the function that queries for changes that occurred doing a specific time period did not retrieve any data. This led us to access protein entries via the text files accessible through the UniProt website. The website provides all proteins entry versions in .txt format.

Although protein entries have the same key codes to indicate the type of information in each line, as mentioned in Table 6.1, the structure of the data following the initial code differs over time, as the first format has been refined and extracted over time. In other words, the representation of the information in each type of line changes across source versions. For example, we found three major differences in format when extracting data from UniProtKB entries:

1. **ID line:** This line contains identification information about the protein. For UniProt, this includes information on which of the two sub-databases (TrEMBL or Swiss-Prot), the entry is stored in. The current version of the format uses the keywords UNREVIEWED and REVIEWED to refer to TrEMBL and Swiss-Prot, respectively. However, in very early entry versions, the keywords PRELIMINARY and STANDARD were used. Figure 6.2 shows an example in which the term PRELIMINARY in version 25 was replaced with UNREVIEWED in version 26. Both terms carry the same meaning. This change was made systematically across the whole database in October 2006.



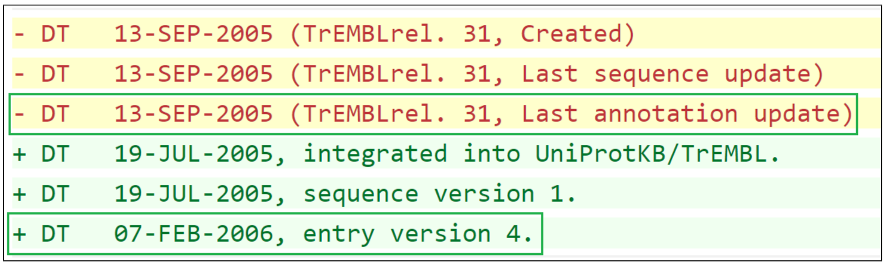
```
- ID   Q96EK9_HUMAN   PRELIMINARY;    PRT;   354 AA.
+ ID   Q96EK9_HUMAN                Unreviewed;        354 AA.
       ...
- DT   05-SEP-2006, entry version 25.
+ DT   31-OCT-2006, entry version 26.
```

Figure 6.2: Changes in representation of the PRELIMINARY to UNREVIEWED in 'Q96EK9' protein entry when comparing versions 25 and 26.

2. **DT line:** Each entry contains three pieces of information indicating changes to the entry: the date the entry was created, the date its sequence was last updated, and the date its annotation and version number were last updated. The version number is what we are concerned with here. When we reviewed the full history

[6] www.uniprot.org/help/programmatic_access

of a number of protein entries, we realised that the early versions of the protein entries contain only the release date, while later versions had the version number as well as the release date. Figure 6.3 shows a comparison between versions 3 and 4 for protein entry ’Q4P0S6’. The highlighted lines show that the version number is absent in version 3, but present in version 4.

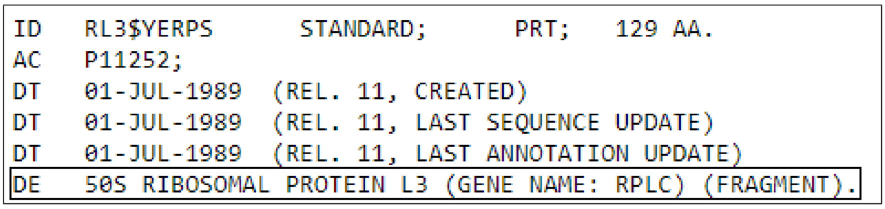


```
- DT   13-SEP-2005 (TrEMBLrel. 31, Created)
- DT   13-SEP-2005 (TrEMBLrel. 31, Last sequence update)
- DT   13-SEP-2005 (TrEMBLrel. 31, Last annotation update)
+ DT   19-JUL-2005, integrated into UniProtKB/TrEMBL.
+ DT   19-JUL-2005, sequence version 1.
+ DT   07-FEB-2006, entry version 4.
```

Figure 6.3: Changes in representation of the protein entry version number.

3. **DE line:** Lines with this header contain protein names; the primary name (main name) and secondary name(s) (other names). From observing a selection of entries, we found three different formats.

   - In the early entry versions, the primary protein name was given, followed by the other names in brackets. Figure 6.4 shows an example of presenting the protein name directly in the protein entry, in a ’DE’ line.

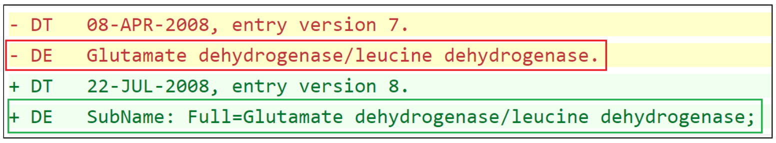


```
ID   RL3$YERPS      STANDARD;      PRT;   129 AA.
AC   P11252;
DT   01-JUL-1989  (REL. 11, CREATED)
DT   01-JUL-1989  (REL. 11, LAST SEQUENCE UPDATE)
DT   01-JUL-1989  (REL. 11, LAST ANNOTATION UPDATE)
DE   50S RIBOSOMAL PROTEIN L3 (GENE NAME: RPLC) (FRAGMENT).
```

Figure 6.4: An example showing the primary protein name in early protein entry versions.

   - The entries belonging to TrEMBL use the keyword SubName (short for ’Submitted Name’) in front of the primary protein name. The keyword ’SubName’ indicates that the protein name has been assigned by the submitter of the protein data [7]. The UniProt consortium started to use ’SubName’ in protein entries from July 2008 onwards. Figure 6.5 shows this change.

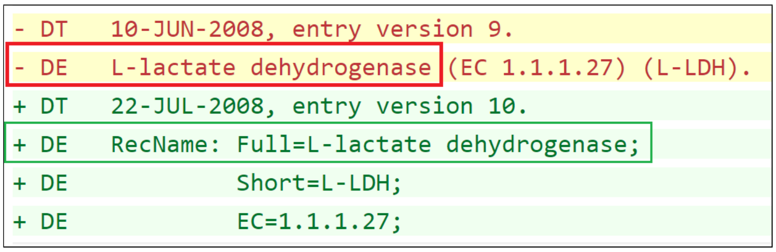
[7] www.uniprot.org/news/2008/07/22/release



```
- DT   08-APR-2008, entry version 7.
- DE   Glutamate dehydrogenase/leucine dehydrogenase.
+ DT   22-JUL-2008, entry version 8.
+ DE   SubName: Full=Glutamate dehydrogenase/leucine dehydrogenase;
```

Figure 6.5: Charging TrEMBL conventions on representing a protein names.

- Entries in Swiss-Prot use the keyword ’RecName’ (shorten for ’Recommended Name’) to indicate the primary protein name, and the keyword ’AltName’ (Alternative Name) for the secondary names. Normally, each entry has only one recommended name. However, when merging entries, all their recommended names will be included in the entry in a subsection, with the first considered to be the primary name. Figure 6.6 shows changes to protein entry ’A8AXK9’, with the keyword ’RecName’ first appearing in version 10.



```
- DT   10-JUN-2008, entry version 9.
- DE   L-lactate dehydrogenase (EC 1.1.1.27) (L-LDH).
+ DT   22-JUL-2008, entry version 10.
+ DE   RecName: Full=L-lactate dehydrogenase;
+ DE            Short=L-LDH;
+ DE            EC=1.1.1.27;
```

Figure 6.6: Appearance of the ’RecName’ keyword in a Swiss-Prot entry.

The three points mentioned above are the key issues to be considered when accessing entries. Some protein entries may have a combination of old and new data formats in the protein entry versions.

6.2 UniProt IQBot Architecture

In Chapter 5, we showed a general architecture for IQBot. In this chapter, we focus on finding changes in protein names by monitoring UniProtKB. More specifically, the aim is to track defects in protein names and find the associated corrections. In addition, IQBot extracts meta-data about the detected defect corrections to help in identifying the reason for the changes. All this should be carried out automatically without human interaction.

Algorithm 1 (mentioned in the previous chapter) outlines the basic steps that must be followed to get the desired results. As we mentioned previously, the curated database selected for the trial is UniProtKB. As shown in Figure 6.7, the UniProtKB-customised IQBot component performs three main steps: collecting changes made to protein names; determining the type of change for each change extracted in the previous step; and discovering the reason for each change. More details on how these three steps work are given in the following sections.

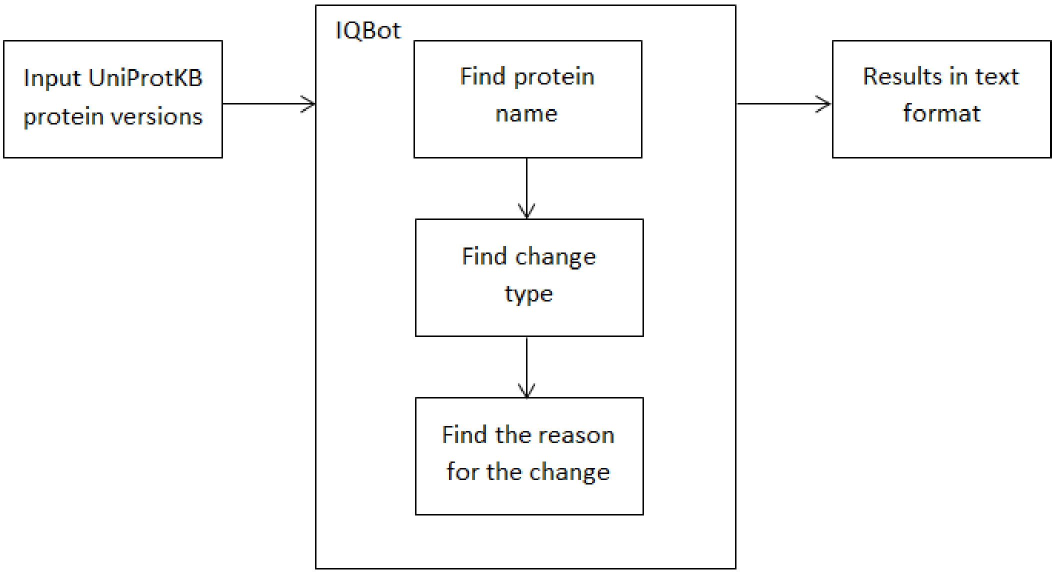

Input UniProtKB protein versions
IQBot
Find protein name
Find change type
Find the reason for the change
Results in text format

Figure 6.7: UniProt IQBot architecture.

6.3 Detecting Defects in UniProt Versions

This section explores the strategy of reusing curation efforts to infer the defects fixed between database releases. We will focus on extracting modifications occurring in one attribute; the protein name. Knowing the changes made to protein names, and the up-to-date protein names, is essential for those who are working in the field, as referring to incorrect protein names can cause problems. Detecting changes in protein names is achieved in two steps. First, we compare two consecutive versions of each protein entry to find out if the protein name has changed. Second, if the protein name has changed, we find the type of change made to it. These two steps will be explained in detail in the following section.

6.3.1 Extracting Pairs of Protein Names

When the IQBot is first run, all protein entries are examined to find the full history of changes made to the protein name. Then, changes made since the last monitored version are extracted. In other words, the database is checked after each new release of the curated database, looking for any change in protein name. The full history of changes to protein names will be stored for future use (As will be explained in Section 6.3.2).

Algorithm 6 presents the steps needed to extract a protein name from UniProtKB protein entry versions. Before monitoring any version of the UniProtKB, the IQBot checks whether or not the database version has been monitored. If it has not, the IQBot monitors each protein. It accesses the protein entry version through its URL and extracts the protein name from lines with the code 'DE". The process of extracting the protein name is repeated with the previous protein entry version. The IQBot removes any keywords provided with the protein name, such as 'RecName', and then compares the protein names extracted from both versions. If the protein names are different, then the protein name in version 'ver-1' is set as a defect, and the name in version 'ver' sets as a correction.

For example, protein entry 'Q12802' (Figure 6.8) shows the order in which the UniProt IQBot monitors the protein entry versions to detect changes in the primary name. Notice that during the full history of the protein, the name changes three times. The first change was detected in version 5, where the name changed from 'P47 LBC ONCOGENE' to 'LBC ONCOGENE'. Then, in version 11, the name changed to 'LBC oncogene'. Lastly, in version 47, the name changed entirely to 'A-kinase anchor protein 13'.

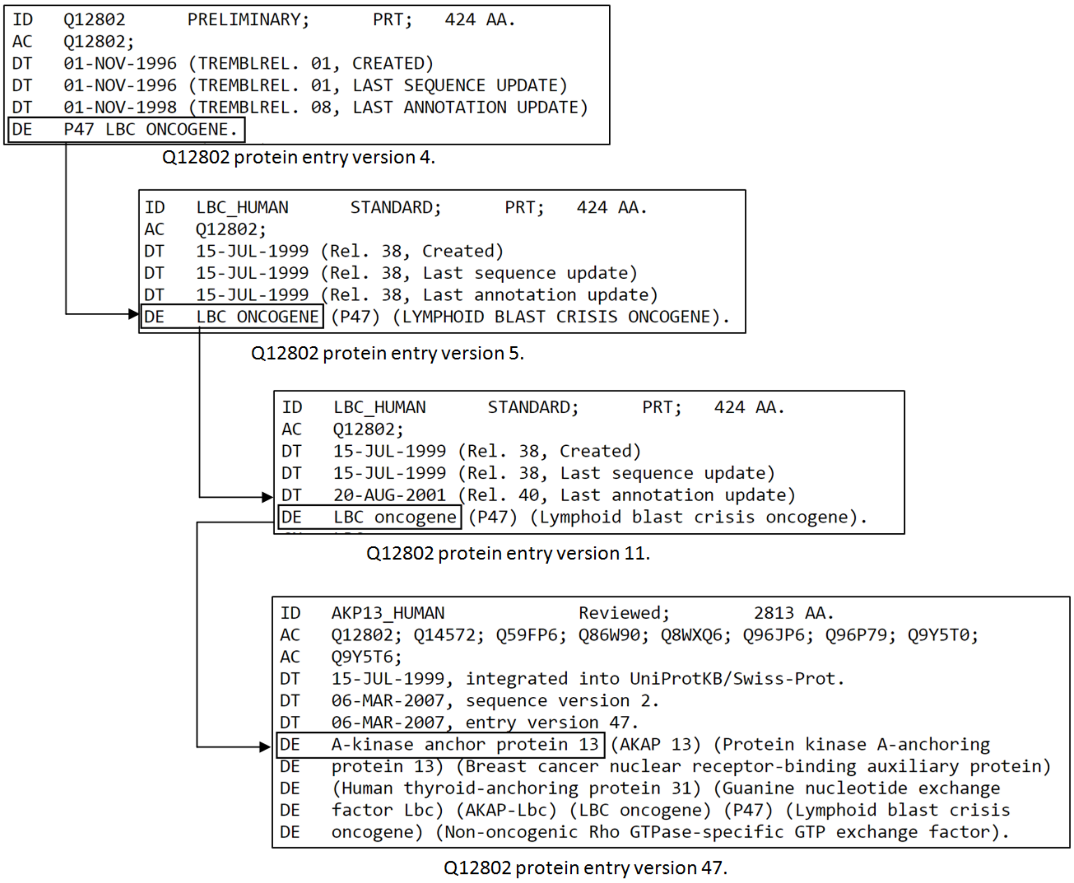
The current IQBot only provides information about the defects in, and corrections to, protein names. The changes are packaged as a collection of the protein itself, the allocated attribute (which is 'protein name'), and the detected defect and correction.

6.3.2 Finding the Type of Change

This step is responsible for assigning the type of change after the IQBot detects a defect and finds the correction in a protein name. When we set the IQBot to monitor UniProtKB for changes in protein names, we found six types of name change. The detected types varied between simple, partial and complete changes. For example, a simple change might involve changing the protein name letter case from upper to lower

Algorithm 6 The UniProtKB algorithm to detect defect correction of the IQBot

```
Require:
    ver: the version of the UniProtKB to find corrections
    ver-1: the version of the UniProtKB to find defects
    output: a list of defects and defect corrections
if UniProtKB version 'ver' is monitored by IQBot then
    go to the next version of UniProtKB
else
    while there are unmonitored versions do
        proteins = get all protein accession numbers
        for each protein do
            access protein information in ver through URL
            namever = extract protein name from 'DE' line
            access protein information in ver-1 through URL
            namever-1 = extract protein name from 'DE' line
            remove keywords from protein names
            if namever <> namever-1 then
                defect = namever-1
                correction = namever
                type = findDefectType(defect, correction)
                reason = findTheReasonForTheChange(type, defect, correction)
                add the tuple to list of defects(recID, defect, correction, type, reason)
            end if
        end for
        add the version number to a list of monitored versions
    end while
end if
```



```
ID   Q12802       PRELIMINARY;      PRT;   424 AA.
AC   Q12802;
DT   01-NOV-1996 (TREMBLREL. 01, CREATED)
DT   01-NOV-1996 (TREMBLREL. 01, LAST SEQUENCE UPDATE)
DT   01-NOV-1998 (TREMBLREL. 08, LAST ANNOTATION UPDATE)
DE   P47 LBC ONCOGENE.
```

Q12802 protein entry version 4.

```
ID   LBC_HUMAN      STANDARD;      PRT;   424 AA.
AC   Q12802;
DT   15-JUL-1999 (Rel. 38, Created)
DT   15-JUL-1999 (Rel. 38, Last sequence update)
DT   15-JUL-1999 (Rel. 38, Last annotation update)
DE   LBC ONCOGENE (P47) (LYMPHOID BLAST CRISIS ONCOGENE).
```

Q12802 protein entry version 5.

```
ID   LBC_HUMAN      STANDARD;      PRT;   424 AA.
AC   Q12802;
DT   15-JUL-1999 (Rel. 38, Created)
DT   15-JUL-1999 (Rel. 38, Last sequence update)
DT   20-AUG-2001 (Rel. 40, Last annotation update)
DE   LBC oncogene (P47) (Lymphoid blast crisis oncogene).
```

Q12802 protein entry version 11.

```
ID   AKP13_HUMAN              Reviewed;        2813 AA.
AC   Q12802; Q14572; Q59FP6; Q86W90; Q8WXQ6; Q96JP6; Q96P79; Q9Y5T0;
AC   Q9Y5T6;
DT   15-JUL-1999, integrated into UniProtKB/Swiss-Prot.
DT   06-MAR-2007, sequence version 2.
DT   06-MAR-2007, entry version 47.
DE   A-kinase anchor protein 13 (AKAP 13) (Protein kinase A-anchoring
DE   protein 13) (Breast cancer nuclear receptor-binding auxiliary protein)
DE   (Human thyroid-anchoring protein 31) (Guanine nucleotide exchange
DE   factor Lbc) (AKAP-Lbc) (LBC oncogene) (P47) (Lymphoid blast crisis
DE   oncogene) (Non-oncogenic Rho GTPase-specific GTP exchange factor).
```

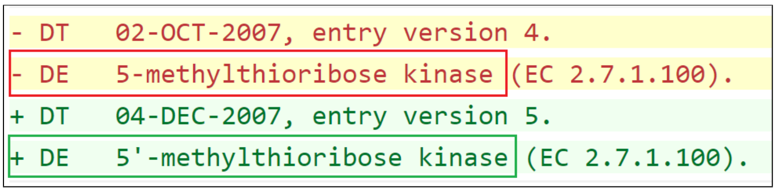
Q12802 protein entry version 47.

Figure 6.8: History of protein name change in protein entry ’Q12802’.

case. The types that we will discuss in the rest of the section fit into the four categories of change type that were mentioned in the previous chapter. We should emphasise that dividing the types is subject to understanding the relationship between the defect and its correction.

We identified six types of protein name defects by looking through the list of protein names extracted by the IQBot. The six types of name defects we have identified are as follows:

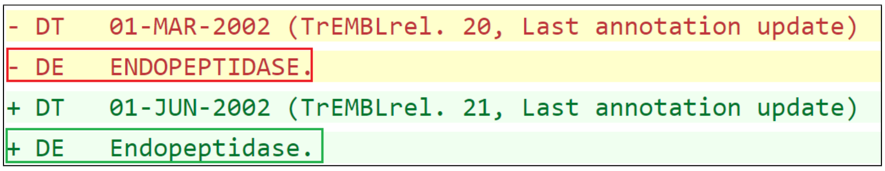
1. **Changing a special character:** In some cases, names are changed by adding or removing a special character, such as space or a comma. Figure 6.9 shows an example of adding a special character, an apostrophe, to the protein name. In version 4, the name is '5-methylthioribose kinase', while in version 5, it is '5'-methylthioribose kinase'. Adding or removing special character relates to standardising the way in which protein names are presented, and so can be seen a 'consistency' type of defect.



```
- DT   02-OCT-2007, entry version 4.
- DE   5-methylthioribose kinase (EC 2.7.1.100).
+ DT   04-DEC-2007, entry version 5.
+ DE   5'-methylthioribose kinase (EC 2.7.1.100).
```

Figure 6.9: Comparison of protein entry 'A4W7Z0' versions 4 and version 5 - example of adding a special character.

2. **Changing letter case:** Change the letter case from upper to lower case letters, and viceversa. This type of change also fits into the category of standardising data to fit community conventions, which is a 'consistency' defect type. Figure 6.10 gives an example. Here, the actual text of the protein name does not change, but the text format does.

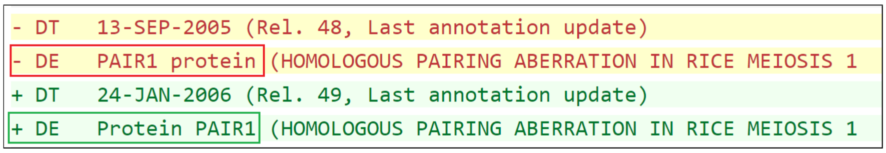


```
- DT   01-MAR-2002 (TrEMBLrel. 20, Last annotation update)
- DE   ENDOPEPTIDASE.
+ DT   01-JUN-2002 (TrEMBLrel. 21, Last annotation update)
+ DE   Endopeptidase.
```

Figure 6.10: Comparison of protein entry 'Q99W78' versions 4 and version 5 - example of changing the letter case.

3. **Misspelled name.** Misspellings can be classified as an accuracy problems. In some cases, the defect in the protein name is due to entering the protein name with an incorrect spelling.

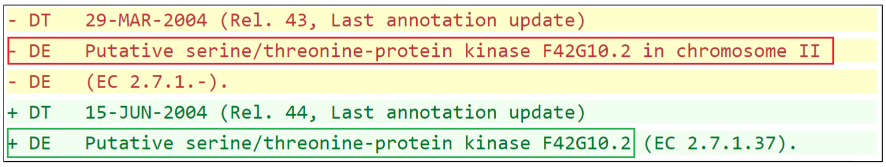
4. **Word order does not follow convention**: In some cases, we came across changes in a protein name that focused on re-arranging the order of words, rather than making a tangible difference to the protein name. Figure 6.11 exemplifies this type of defect.



```
- DT   13-SEP-2005 (Rel. 48, Last annotation update)
- DE   PAIR1 protein (HOMOLOGOUS PAIRING ABERRATION IN RICE MEIOSIS 1
+ DT   24-JAN-2006 (Rel. 49, Last annotation update)
+ DE   Protein PAIR1 (HOMOLOGOUS PAIRING ABERRATION IN RICE MEIOSIS 1
```

Figure 6.11: Comparison of protein entry 'Q75RY2' version 7 and version 8 - example of word re-ordering.

5. **Add or remove parts of the protein name:** Another set of defects focuses on partially changing the protein name by adding or removing some parts of the name. Figure 6.12 shows an instance of a protein name being removed. We consider this defect type to be an accuracy-related issue.

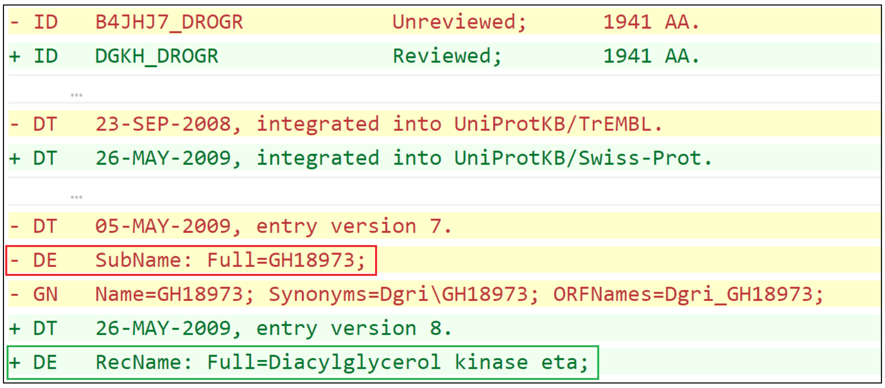


```
- DT   29-MAR-2004 (Rel. 43, Last annotation update)
- DE   Putative serine/threonine-protein kinase F42G10.2 in chromosome II
- DE   (EC 2.7.1.-).
+ DT   15-JUN-2004 (Rel. 44, Last annotation update)
+ DE   Putative serine/threonine-protein kinase F42G10.2 (EC 2.7.1.37).
```

Figure 6.12: Comparison of protein entry 'Q20347' version 25 and version 26 - example of removing some parts of the name.

6. **Incorrect name:** 'Incorrect name' is when the protein name changes completely. Figure 6.13 gives an example of such a defect being corrected. The UniProt IQBot assigns this as an 'accuracy' defect.

The types of defects listed above are assigned automatically to each pair of protein names. The IQBot contains a function called findChangeType, which takes the defect and the correction as parameters. The function has a list of conditions that matches the defect types mentioned above. After a change type is assigned to each pair of names, the reason behind each change should be determined (if applicable). However, the primary job of the IQBot is complete at this stage.



```
- ID   B4JHJ7_DROGR            Unreviewed;       1941 AA.
+ ID   DGKH_DROGR              Reviewed;         1941 AA.
    ...
- DT   23-SEP-2008, integrated into UniProtKB/TrEMBL.
+ DT   26-MAY-2009, integrated into UniProtKB/Swiss-Prot.
    ...
- DT   05-MAY-2009, entry version 7.
- DE   SubName: Full=GH18973;
- GN   Name=GH18973; Synonyms=Dgri\GH18973; ORFNames=Dgri_GH18973;
+ DT   26-MAY-2009, entry version 8.
+ DE   RecName: Full=Diacylglycerol kinase eta;
```

Figure 6.13: Comparison of protein entry 'B4JHJ7' version 7 and version 8 - example of a complete name change.

6.4 Determining the Reason for a Change

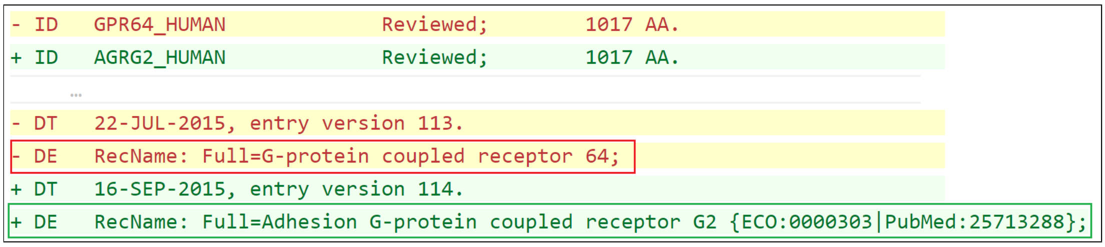
In Chapter 5, we mentioned that determining the reason for a change is not an easy task because this kind of information cannot always be found directly in the data records.

To find the reason behind every kind of protein name change, the entries where the change occurred need to be reviewed. That is, we need to compare the content of the two entries: the entry version where the change occurs, and the previous version of that entry. A collection of 1499 pairs of names were reviewed. These pairs of names are extracted in the first step of the approach. As mentioned in Section 6.1, each protein entry contains a list of information, with each piece of information represented by a specific line. Each line is compared in both versions.

Some of the reasons for changes in protein names are obvious and do not require much discussion, such as spelling mistake corrections and changes to letter case (i.e., changes related to standardisation). On the other hand, some changes need more explanation. As we are aiming to maximise the usefulness of curation efforts, we aim to use only the available information to discover the reason for the name change, without any further input from the data curators.

In Section 6.1, we mentioned that UniProtKB uses automatic and manual methods to curate entries. For the manually curated entries, curators make the changes based on an investigation of the literature in the area, or by making their assumption. In the early versions of UniProt (i.e., until 2015) curators did not provide any information on why they made changes. The challenge here is to correctly come up with a reasonable assumption for the reason behind each type of name change.

After September 2015, curators started to use the Evidence Code Ontology [8] (ECO) to give a clear reason for changes made. For example, when the code ECO:0000250 is used to annotate the protein name, this means the reason for the change is due to evidence of sequence similarity. UniProt uses a couple of ECO evidence code, Table 6.2 shows all the ECO evidence code used in UniProt at the time of writing. It should be noted that ECO evidence codes differ between manually- and automatically-curated entries. Table 6.2 shows which codes relate to manually-curated records and which relate to automatically-curated records. Figure 6.14 presents an example of an ECO evidence code provided when curating a protein name. The curator used the ECO code 'ECO:0000303—PubMed:25713288', with 'ECO:0000303' indicating that the new name is supported by evidence from a non-traceable author statement, and 'PubMed:25713288' indicating the source of information.



```
- ID   GPR64_HUMAN             Reviewed;         1017 AA.
+ ID   AGRG2_HUMAN             Reviewed;         1017 AA.
     ...
- DT   22-JUL-2015, entry version 113.
- DE   RecName: Full=G-protein coupled receptor 64;
+ DT   16-SEP-2015, entry version 114.
+ DE   RecName: Full=Adhesion G-protein coupled receptor G2 {ECO:0000303|PubMed:25713288};
```

Figure 6.14: Comparison of protein entry 'Q8IZP9' version 113 and version 114 - example of using evidence codes.

Where ECO evidence codes do not appear in protein entry versions, we need to infer the reasons for the changes by observing others properties of protein entries. In other words, for each time a change in protein name is detected, the version where a change occurred and the previous version of the same entry are compared. The comparison results in some changes in the entry that are associated with a protein name change. However, we need to distinguish whether these changes result in changing the protein name. To do this, we analysed a collection of changes to protein names, and inferred the following reasons for change:

1. The entry undergoes a manual curation for the first time. When a curator curates an entry for the first time, she or he may make many changes, including to the protein name. This information can be verified if an entry is moved from TrEMBL to Swiss-Prot when a name change is detected. This can be checked by comparing the ID line in both entry versions. As we mentioned before, the

[8] www.evidenceontology.org/Welcome.html

Table 6.2 ECO evidence codes used in UniProt entries and their meaning [Uni15].

Entry Type	ECO	Meaning
Manual	ECO:0000269	provided by experimental evidence.
Manual	ECO:0000303	non-traceable author statement.
Manual	ECO:0000250	sequence similarity evidence.
Manual	ECO:0000312	imported from another database manually.
Manual	ECO:0000305	based on scientific knowledge of the curator.
Manual	ECO:0000255	match to sequence model evidence.
Manual	ECO:0000244	a combination of experimental and computational evidence.
Automatic	ECO:0000256 ECO:0000259	match to sequence model evidence.
Automatic	ECO:0000313	imported information.
Automatic	ECO:0000213	a combination of experimental and computational evidence.

ID line contains the information related to the database where the protein entry version is stored. If the previous entry version includes the keyword 'UNREVIEWED' and the other version includes 'REVIEWED', this version marks the point at which the entry was curated manually for the first time. As we mentioned earlier in Section 6.1.1, the old versions of UniProt entries use the words 'PRELIMINARY' and 'STANDARD' in this context.

2. The protein name was changed when the protein entry was merged with other entries. As mentioned in by the UniProt Consortium [Con14b], when a number of protein entries have the same gene name in common, the curator will merge all these entries into one on the grounds that they represent independent discoveries of a single protein. Frequently, the protein name is changed as a result of merging. This data can be found in the AC line in the entry. To determine whether an entry is combined with other entries, the AC line should be compared in both versions. If the AC line in both versions gives different values, then the reason for the change will be because of merging entries.

3. In other cases, the protein name was changed due to new publications appearing that presented new data or theories about the protein's structure or function. This can be detected by checking if the publications listed in the protein entry version where the name change occurred differ from those listed in the previous protein entry version.

4. In most of the cases we came across, when some parts of the protein name were removed, the removed part was added as a Flag to the protein entry, or even moved to other sections inside the entry. After investigating, we realised that this change occurred due to changes in the UniProt naming guidelines. The revised guidelines do not allow certain kinds of terms to be used in protein names. This illustrates how some changes are related to efforts to standardise the protein names used in the community.

5. The curators' job is to apply their domain knowledge to improve the quality of the data, which leads them to make changes to it. In some rare situations, none of the above-mentioned reasons match the entry data. From reviewing [Con14b], it can be inferred that curators sometimes make changes to protein entries based on their own knowledge, expertise, and analysis of the publications in the area.

All the reasons mentioned above resulted from reviewing a collection of entries manually and automatically. Manually by checking random entries to see the changes that occurred in entries beside changes protein name. And automatically by finding the changes associated with the name change.

6.5 Evaluating

The aim of this section is to see if the IQBot concept can work in practice. Here we focus on answering two questions: Can we infer information about defects and their corrections by observing the updates expert curators make to curated data resources? Can those defects and corrections be packaged in a way that allows them to be reused by the owners of other resources in the same or similar domains?

To evaluate this, we first need to run IQBot on a real curated resource, in order to collect some inferred defects and corrections. We used UniProtKB as the curated source to be monitored by the IQBot. The IQBot extracted defects and corrections in protein names (as has been explained earlier in this chapter). We then see if we can locate other sources of protein-related data that contain these defects (but have not been yet corrected).

To find out if other databases contain the defects extracted by the IQBot, we used DBGET [9], a search tool dedicated to querying a collection of biological databases, to access data in other biomedical resources. We used DBGET because it can be

[9] www.genome.jp/dbget

Algorithm 7 Pseudocode to search different databases for the name defects in DBGET.

```
Require:
    input: a list of defects and corrections from IQBot
    input: DBs a list of databases to query
access DBGET
for each database in DBs do
    for each defect and its correction do
        if the database contains the correction then
            countDefect
        else
            if the database contains the defect then
                countDefect
            end if
        end if
    end for
    add result to the list (database, countDefect, countCorrection)
end for
```

accessed programmatically. In addition, using DBGET saves time and effort as there is no need to write different code to access each individual database. DBGET uses the same mechanism to access all provided databases. We implemented a Java programme that accessed the biomedical databases in DBGET and checked whether each database still contained any of the defects or corrections produced by IQBot. We searched eight databases: KEGG BRITE[10]; GO[11]; KEGG GENES[12]; KEGG DGENES; KEGG ORTHOLOGY[13]; KEGG MGENES[14]; NCBI-Gene[15]; and KEGG ENZYME[16].

The algorithm 7 gives the pseudocode for finding the defects and corrections in the DBGET databases. For each database, we checked the availability of both defects and corrections resulting from IQBot monitoring of UniProtKB. If the database contains both the defect and its correction, or only the correction, it is considered as not having the defect. If the database only has the defect, then the database will be considered to contain the defect. Finally, for each database, we used two counters; one for the number of defects in the database, and the second for the number of corrections in the database.

[10] www.genome.jp/kegg/brite.html

[11] www.geneontology.org

[12] www.genome.jp/kegg/genes.html

[13] www.genome.jp/kegg/ko.html

[14] www.genome.jp/mgenes

[15] www.ncbi.nlm.nih.gov/gene

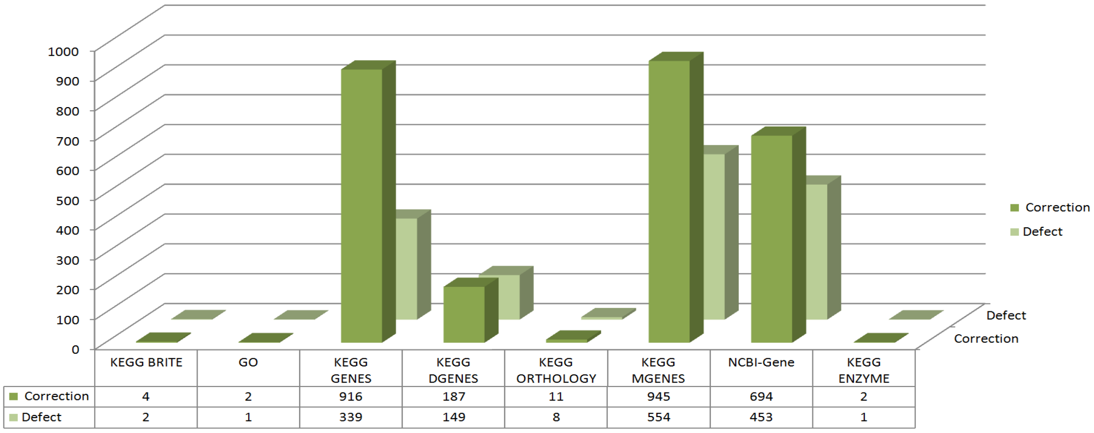
[16] www.genome.jp/kegg/annotation/enzyme.html

6.5.1 Experimental Results

We implemented a prototype of IQBot, using Java programming language, to monitor UniProtKB. Since we ran it on a personal machine, the process of extracting changes took more than 4 hours. When we ran IQBot to monitor UniProtKB, we found that the human proteins show more changes to protein names, in comparison to other types of proteins such as those of rats and zebrafish. This led us to focus on monitoring human protein entries only. We randomly chosen 249 human protein entries which had changes in protein names. The number of human protein entries was more than 249, but we limited it to the entries where protein names had changed. When IQBot monitored all versions of the 249 proteins, it found 1499 changes.

We took the 1499 changes detected by IQBot monitoring of UniProtKB and examined whether other biomedical databases had defects or corrections in their data. We used DBGet to access data from eight different databases, as shown in Figure 6.15. For each database, we checked to see if the database contained the defect or the correct protein name.

Figure 6.15 shows the numbers of times defects and corrections appear in the other databases. We can see that all the eight databases have both defects and corrections. KEGG GENS, KEGG MGENES and NCBI-Gen have the highest number of entries containing corrections. The number of defects found in these three databases was between 37% and 65%, in comparison to the number of corrections found. KEGG DGENES had 187 corrections and 149 defects which are almost similar amount of changes. The other databases had a lower number of defects and corrections in their data. In general, the figure below shows a wide variety in the number of defects and corrections found in databases. Using IQBot could limited the number of defects found in the data, and help databases to keep their data updated.



	KEGG BRITE	GO	KEGG GENES	KEGG DGENES	KEGG ORTHOLOGY	KEGG MGENES	NCBI-Gene	KEGG ENZYME
🟩 Correction	4	2	916	187	11	945	694	2
🟩 Defect	2	1	339	149	8	554	453	1

Figure 6.15: Availability of defects and corrections in DBGET sources.

6.6 Behind Extracting Defects and Corrections

As explained in this chapter, IQBot can extract defects, corrections, type of changes, and the reason for these changes. These results can be used to update data in other databases. The information gathered can also help in improving understanding of curation process, as we can see whether a change is made to meet a community convention (such as replacing upper case with lower case letters, or not using certain special characters when naming proteins). Having more information on the reason for the change aids further understanding of why a correction has been made.

6.7 Conclusion

In this chapter, we applied the IQBot (introduced in the previous chapter) to UniProtKB, a curated database. We came across a number of practical difficulties in applying the approach arising from the realities and challenges of working with real data sources. Over time, UniProtKB has used different ways to represent specific information, including changes to data value formats, identifying schemas, and the amount/-type of information stored. Any tool such as IQBot, which explores old versions of code, must cope with these.

We succeeded in creating a simple IQBot to monitor the UniProtKB database and extract defects and corrections to protein names. We were also able, in some cases, to ascertain the reason why the change happened. The challenges we faced in this trial will need be considered when monitoring other curated databases, as we may face the same kind of difficulties.

We carried out an evaluation exercise to demonstrate that the IQBot can produce results that help other data sources. When monitoring UniProtKB for defects in protein names, we found that other data sources still contain the defects, rather than the corrected protein names. This means that these other resources could use the defects inferred by IQBot to improve the quality of their data. Besides, providing the reason for changes may be useful in giving data consumers extra information to help them understand the process of data curation, and help to decide whether they need to apply the corrections or not.

Chapter 7

Conclusions

Education is the passport to the future, for tomorrow belongs to those who prepare for it today.

Malcolm X

In the research reviewed in Chapter 2, the key focus is on improving methods for curating data. The research covers aspects such as providing API, and downloading and accessing the curated data (including old and current versions).

However, there has been less attention paid to enabling connection between curated sources and users/consumers. Similarly, minimal research has been conducted in relation to the reuse of curation efforts, or into ways to help data consumers. These kind of actions might be expensive at the outset, but will help in making the connection between consumers and data source managers more accessible and user-friendly.

This thesis has tackled issues related to the concept of *data curation*. In particular, it has focused on maximising the benefit gained from curation efforts made by domain experts in the biomedical domain. The aim of the research we have conducted is to allow current and newly-formed biocuration communities to benefit from the existing curation processes and the expertise available in the domain, as well as supporting communities with limited resources who may be unable to carry out their own curation. Additionally, we aimed to reduce the redundancy of curation practices in the domain. In some cases, communities share the same data, but curate it in isolation from each other. This can result in duplication of effort. To deal with this, we have proposed several ways to help communities reuse the curation work done with well-resourced curated sources and expand this use to benefit other, less well-resourced sites.

A key issue in this thesis has been: How can we address these problems in a way that requires less human interaction? In other words, the proposed solutions have focused on using the available resources relating to data curation, without requiring further contact with data curators. This takes us back to the research questions we asked in Section 1.3.2. To answer the first question, we proposed BIOC-MM as a way to assess the maturity of the curation process. The rest of our questions revolved around the same aspect, and we proposed the IQBot as an answer to those questions. The IQBot showed that it is possible to infer defects and corrections by monitoring the changes from two consecutive versions of data in a curated source. It also provides metadata about the detected defect.

The rest of this chapter gives a summary of the thesis contributions in Section 7.1. Then, Sections 7.2 and 7.3 discuss limitations and possible future work.

7.1 Summary of Research Contributions

This section gives a summary of the main contributions made by this thesis.

7.1.1 The BIOC-MM Maturity Model

We focused on providing a way to assess the maturity of a curation process, due to the limitation of having a model that shows the workflow and procedures followed to perform biocuration. There are redundant curation procedures among communities, as many are working individually. Therefore, we aimed not only to provide a general description of biocuration, but at the same time, focus on showing all the available procedures used during curation in the biomedical domain. We aimed to give a list of the existing methods of biocuration. Providing the list may help communities to benefit from the work of others (who have already managed to deal with similar curation cases). To achieve this, we created a maturity model for biocuration, which we named BIOC-MM. BIOC-MM was built based on a review of the curation procedure for five curated biomedical databases, and on the literature relating to biocuration from the last five years. In addition to providing a general description of the biocuration process, the maturity model allows biocuration communities to assess their maturity level, and provides communities with guidance for achieving the target level of maturity.

Following the proposed maturity model, we evaluated the model using several different methods. For each type of evaluation, the model was modified based on the feedback and comments.

7.1.2 IQBot

As stated previously, one of our aims was to help data sources operating with limited resources which prevent them from affording curation. The focus here was on extracting the results of curation work on an existing data source and packaging them in a way that helps other data sources to reuse them. We solved this problem by introducing the idea of an IQBot. An IQBot extracts the changes (i.e., the defects and corrections) made between two consecutive versions of the data. The IQBot also attempts to infer the reason for the defect corrections made, where applicable.

We ran the IQBot by connecting it to a real-world curated database, UniProtKB. When we extracted data from UniProtKB and compared the extracted data to other databases (i.e., sources in the same domain as UniProtKB), we found that using IQBot can be beneficial because it ensures data in the databases is up-to-date. However, connecting the IQBot to UniProtKB raised some practical implementation challenges, such as methods of accessing the curated source data and the way in which the data is represented in the curated source.

7.2 Limitations

Model maintenance The current version of the BIOC-MM model reflects the current situation in biocuration. However, the model will need to be maintained in the future if it is to remain useful. It should be noted that the maintenance process includes adding or removing curation procedures and practices, and changing the levels and goals in the model.

Data extraction In IQBot, the method of extracting data is related to the access mechanism offered by the curated source and how it represents its data (not all data sources use the same approach). So, when connecting IQBot to a curated source, it must be adjusted to work with the data representation in the monitored source.

Determining the reason for the change There was no direct way to extract the reason for a change from UniProt data. We had to infer it from secondary resources. There were different ways to do this, depending on the data available in the data records. This is a key limitation, as the method of extracting the reason for a change needs to be determined based on the record.

7.3 Future Work

Experts curator evaluation We discussed the evaluation process of the BIOC-MM; as part of this process, we approached human experts in the domain to carry out an evaluation. However, we will need to ask experts from different groups to evaluate the model, rather than having the model evaluated by a single expert from one field.

Different evaluation method According to Helgesson et al., there are three types of approaches for evaluating a maturity model [HHW12]. This thesis applied two of these types: *offline* evaluations and *expert* evaluations. The third type of evaluation would require use of the BIOC-M by a curated community, in order to see how it works in practice and establish whether or not it needs to be modified. The third type of evaluation was not covered in this thesis because it is a time-consuming task that was outside the duration and scope of the study. It would be beneficial if the model could be evaluated in this way in the future.

Adapt maturity models in various domains From our experience with the maturity model, we found that most of the available research focused on making maturity models for business processes. However, this thesis included building a maturity model for a different domain, which is the biomedical domain, and through this experience, we would like to encourage other domains to try building maturity models (because of the benefits that can be obtained by using it).

Publishing the results Since the IQBot can extract defects, their corrections, and the reasons for these change, the next logical step is to publish the results in a form which allows them to be easily understood by owners of relevant resources. We suggest using Linked Open Data (LOD) principles to publish the results. To achieve this, Apache Marmotta [1], which is a framework for publishing Linked Data, could be a suitable platform. Making the results available on the web will help users and owners of databases that are still using out-of-date data to discover defects and remove them from their data. This will raise their data quality by updating their records to incorporate new data.

Accessing previous data versions When looking for a curated database to monitor in the evaluation of IQBot, there was a difficulty in finding a curated database

[1] http://marmotta.apache.org

which provides access to all previous versions. Not having access to the previous versions of curated databases will prevent IQBot from providing a full history of defects and corrections. To resolve this, we propose adding a feature to IQBot that allows database versions to be preserved as snapshott, which can be compared later with the new version of the database when it is released.

Bibliography

[ACS$^+$14] Stephen Abrams, Patricia Cruse, Carly Strasser, Perry Willet, Geoffrey Boushey, Julia Kochi, Megan Laurance, and Angela Rizk-Jackson. DataShare: empowering researcher data curation. *International Journal of Digital Curation*, 9(1):110–118, 2014.

[AES17] Mariam Alqasab, Suzanne M Embury, and Sandra de F Mendes Sampaio. Amplifying Data Curation Efforts to Improve the Quality of Life Science Data. *International Journal of Digital Curation*, 12(1):1–12, 2017.

[BC06] Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101, 2006.

[BCA$^+$00] David Botstein, J Ms Cherry, M Ashburner, CA Ball, JA Blake, H Butler, AP Davis, K Dolinski, SS Dwight, JT Eppig, et al. Gene Ontology: tool for the unification of biology. *Nat genet*, 25(1):25–9, 2000.

[BDD$^+$14] James P Balhoff, Wasila M Dahdul, T Alexander Dececchi, Hilmar Lapp, Paula M Mabee, and Todd J Vision. Annotation of phenotypic diversity: decoupling data curation and ontology curation using Phenex. *Journal of biomedical semantics*, 5(1):45, 2014.

[BGF$^+$12] Stephanie M Bunt, Gary B Grumbling, Helen I Field, Steven J Marygold, Nicholas H Brown, and Gillian H Millburn. Directly e-mailing authors of newly published papers encourages community curation. *Database: The Journal of Biological Databases and Curation*, 2012.

[BKP09] Jörg Becker, Ralf Knackstedt, and Jens Pöppelbuß. Developing maturity models for IT management. *Business & Information Systems Engineering*, 1(3):213–222, 2009.

[BMSN$^+$16] Peter Bourgonje, Julian Moreno-Schneider, Jan Nehring, Georg Rehm, Felix Sasaki, and Ankit Srivastava. Towards a platform for curation technologies: enriching text collections with a semantic-web layer. In *International Semantic Web Conference*, pages 65–68. Springer, 2016.

[C$^+$14] UniProt Consortium et al. Activities at the universal protein resource (UniProt). *Nucleic acids research*, 42(D1):D191–D198, 2014.

[CAB$^+$98] J Michael Cherry, Caroline Adler, Catherine Ball, Stephen A Chervitz, Selina S Dwight, Erich T Hester, Yankai Jia, Gail Juvik, TaiYun Roe, Mark Schroeder, et al. SGD: Saccharomyces genome database. *Nucleic acids research*, 26(1):73–79, 1998.

[CABO$^+$14] Andrew Chatr-Aryamontri, Bobby-Joe Breitkreutz, Rose Oughtred, Lorrie Boucher, Sven Heinicke, Daici Chen, Chris Stark, Ashton Breitkreutz, Nadine Kolas, Lara O'donnell, et al. The BioGRID interaction database: 2015 update. *Nucleic acids research*, 43(D1):D470–D478, 2014.

[CaKK$^+$08] Andrew Chatr-aryamontri, Samuel Kerrien, Jyoti Khadake, Sandra Orchard, Arnaud Ceol, Luana Licata, Luisa Castagnoli, Stefano Costa, Cathy Derow, Rachael Huntley, et al. MINT and IntAct contribute to the Second BioCreative challenge: serving the text-mining community with high quality molecular interaction data. *Genome biology*, 9(2):S5, 2008.

[CIDC$^+$13] Donald C Comeau, Rezarta Islamaj Doğan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, et al. BioC: a minimalist approach to interoperability for biomedical text processing. *Database: The Journal of Biological Databases and Curation*, 2013.

[CJAH06] Richard G Côté, Philip Jones, Rolf Apweiler, and Henning Hermjakob. The Ontology Lookup Service, a lightweight cross-platform tool for controlled vocabulary queries. *BMC bioinformatics*, 7(1):97, 2006.

[CLMO14] David Campos, Jóni Lourenço, Sérgio Matos, and José Luís Oliveira. Egas: a collaborative and interactive document curation platform. *Database: The Journal of Biological Databases and Curation*, 2014.

[CLN$^+$13] David Campos, J Lourencço, Tiago Nunes, Rui Vitorino, Pedro Domingues, Sérgio Matos, and José Luís Oliveira. Egas–collaborative biomedical annotation as a service. In *Proceedings of the Fourth BioCreative Challenge Evaluation Workshop*, volume 1, pages 254–259, 2013.

[CMB$^+$09] Carlos Cano, Thomas Monaghan, Armando Blanco, Dennis P Wall, and Leonid Peshkin. Collaborative text-annotation resource for disease-centered relation extraction from biomedical text. *Journal of biomedical informatics*, 42(5):967–977, 2009.

[CNS13] John A Clarke, Karen J Nelson, and Ian D Stoodley. The place of higher education institutions in assessing student engagement, success and retention: A maturity model to guide practice. 2013.

[Con03] FlyBase Consortium. The FlyBase database of the Drosophila genome projects and community literature. *Nucleic acids research*, 31(1):172–175, 2003.

[Con14a] UniProt Consortium. UniProt Knowledgebase User Manual. `https://web.expasy.org/docs/userman.html`, 2014. [Online; accessed 19-November-2015].

[Con14b] UniProt Consortium. UniProt Manual Curation SOP. `www.uniprot.org/docs/sop_manual_curation.pdf`, 2014. [Online; accessed 05-November-2015].

[Con16] UniProt Consortium. The UniProt Archive (UniParc). `www.uniprot.org/help/uniparc`, 2016. [Online; accessed 15-February-2016].

[Con17] UniProt Consortium. Protein naming guidelines. `www.uniprot.org/docs/nameprot`, 2017. [Online; accessed 20-January-2017].

[CRR15] Samuel Croset, Joachim Rupp, and Martin Romacker. Flexible data integration and curation using a graph-based approach. *Bioinformatics*, 32(6):918–925, 2015.

[CZT$^+$13] Rajesh Chowdhary, Jinfeng Zhang, Sin Lam Tan, Daniel E Osborne, Vladimir B Bajic, and Jun S Liu. PIMiner: a web tool for extraction of

Protein Interactions from Biomedical Literature. *International journal of data mining and bioinformatics*, 7(4):450–462, 2013.

[DBFKR05] Tonia De Bruin, Ronald Freeze, Uday Kaulkarni, and Michael Rosemann. Understanding the main phases of developing a maturity assessment model. 2005.

[DEE+13] Michele Dallachiesa, Amr Ebaid, Ahmed Eldawy, Ahmed Elmagarmid, Ihab F Ilyas, Mourad Ouzzani, and Nan Tang. NADEEF: a commodity data cleaning system. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 541–552. ACM, 2013.

[DKFB+10] Emily Doughty, Attila Kertesz-Farkas, Olivier Bodenreider, Gary Thompson, Asa Adadey, Thomas Peterson, and Maricel G Kann. Toward an automatic method for extracting cancer-and other disease-related point mutations from the biomedical literature. *Bioinformatics*, 27(3):408–415, 2010.

[DMHH+09] Karen G Dowell, Monica S McAndrews-Hill, David P Hill, Harold J Drabkin, and Judith A Blake. Integrating text mining into the MGI biocuration workflow. *Database: The Journal of Biological Databases and Curation*, 2009.

[DTW+13] Lin Dai, Ming Tian, Jiayan Wu, Jingfa Xiao, Xumin Wang, Jeffrey P Townsend, and Zhang Zhang. AuthorReward: increasing community curation in biological knowledge wikis through automated authorship quantification. *Bioinformatics*, 29(14):1837–1839, 2013.

[DWL+14] Hong-Jie Dai, Johnny Chi-Yang Wu, Wei-San Lin, Aaron James F Reyes, Shabbir Syed-Abdul, Richard Tzong-Han Tsai, Wen-Lian Hsu, et al. LiverCancerMarkerRIF: a liver cancer biomarker interactive curation system combining text mining and expert annotations. *Database: The Journal of Biological Databases and Curation*, 2014.

[EJSE14] Suzanne M Embury, Binling Jin, Sandra Sampaio, and Iliada Eleftheriou. On the Feasibility of Crawling Linked Data Sets for Reusable Defect Corrections. In *LDQ@ SEMANTICS*. Citeseer, 2014.

[FBB15] Scott E Friedman, J Benton, and Dan Bryce. Toward Automatic Ontology Curation with Similarity-Based Reasoning. 2015.

[FBSS$^+$09] Jean-Fred Fontaine, Adriano Barbosa-Silva, Martin Schaefer, Matthew R Huska, Enrique M Muro, and Miguel A Andrade-Navarro. MedlineRanker: flexible ranking of biomedical literature. *Nucleic acids research*, 37(suppl_2):W141–W146, 2009.

[FLM$^+$12] Wenfei Fan, Jianzhong Li, Shuai Ma, Nan Tang, and Wenyuan Yu. Towards certain fixes with editing rules and master data. *The VLDB Journal*, 21(2):213–238, 2012.

[FMTY14] Wenfei Fan, Shuai Ma, Nan Tang, and Wenyuan Yu. Interaction between record matching and data repairing. *Journal of Data and Information Quality (JDIQ)*, 4(4):16, 2014.

[GMPS13] Floris Geerts, Giansalvatore Mecca, Paolo Papotti, and Donatello Santoro. The LLUNATIC data-cleaning framework. *Proceedings of the VLDB Endowment*, 6(9):625–636, 2013.

[HBK$^+$12] Lynette Hirschman, Gully AP Burns, Martin Krallinger, Cecilia Arighi, K Bretonnel Cohen, Alfonso Valencia, Cathy H Wu, Andrew Chatr-Aryamontri, Karen G Dowell, Eva Huala, et al. Text mining for the biocuration workflow. *Database: The Journal of Biological Databases and Curation*, 2012.

[HHW12] Yeni Yuqin Li Helgesson, Martin Höst, and Kim Weyns. A review of methods for evaluation of maturity models for process improvement. *Journal of Software: Evolution and Process*, 24(4):436–454, 2012.

[HOO09] Kai M Hüner, Martin Ofner, and Boris Otto. Towards a maturity model for corporate data quality management. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 231–238. ACM, 2009.

[HT03] Shihong Huang and Scott Tilley. Towards a documentation maturity model. In *Proceedings of the 21st annual international conference on Documentation*, pages 93–99. ACM, 2003.

[IDKCa$^+$17] Rezarta Islamaj Doğan, Sun Kim, Andrew Chatr-aryamontri, Christie S Chang, Rose Oughtred, Jennifer Rust, W John Wilbur, Donald C Comeau, Kara Dolinski, and Mike Tyers. The BioC-BioGRID corpus: full text articles annotated for curation of protein–protein and genetic interactions. *Database: The Journal of Biological Databases and Curation*, (1):baw147, 2017.

[INN99] Jakob Iversen, Peter Axel Nielsen, and Jacob Norbjerg. Situated assessment of problems in software development. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 30(2):66–81, 1999.

[ISB] International Society for Biocuration. `www.biocuration.org`. [Online; accessed 01-May-2018].

[JRR$^+$13] Daniel G Jamieson, Phoebe M Roberts, David L Robertson, Ben Sidders, and Goran Nenadic. Cataloging the biomedical world of pain through semi-automated curation of molecular interactions. *Database: The Journal of Biological Databases and Curation*, 2013, 2013.

[Kar16] Peter D Karp. Crowd-sourcing and author submission as alternatives to professional curation. *Database: The Journal of Biological Databases and Curation*, 2016.

[KBA$^+$15] Ritu Khare, John D Burger, John S Aberdeen, David W Tresner-Kirsch, Theodore J Corrales, Lynette Hirchman, and Zhiyong Lu. Scaling drug indication curation through crowdsourcing. *Database: The Journal of Biological Databases and Curation*, 2015.

[KCCR$^+$11] Huda Khan, Brian Caruso, Jon Corson-Rikert, Dianne Dietrich, Brian Lowe, and Gail Steinhart. Datastar: Using the semantic web approach for data curation. *International Journal of Digital Curation*, 6(2):209–221, 2011.

[KKS$^+$14] Dongseop Kwon, Sun Kim, Soo-Yong Shin, Andrew Chatr-aryamontri, and W John Wilbur. Assisting manual literature curation for protein–protein interactions using BioQRator. *Database: The Journal of Biological Databases and Curation*, 2014.

[KSW$^+$14] Ingrid M Keseler, Marek Skrzypek, Deepika Weerasinghe, Albert Y Chen, Carol Fulcher, Gene-Wei Li, Kimberly C Lemmer, Katherine M Mladinich, Edmond D Chow, Gavin Sherlock, et al. Curation accuracy of model organism databases. *Database: The Journal of Biological Databases and Curation*, 2014.

[LCaM$^+$10] Florian Leitner, Andrew Chatr-aryamontri, Scott A Mardis, Arnaud Ceol, Martin Krallinger, Luana Licata, Lynette Hirschman, Gianni Cesareni, and Alfonso Valencia. The FEBS Letters/BioCreative II. 5 experiment: making biological information accessible. *Nature biotechnology*, 28(9):897, 2010.

[LHR$^+$13] Eduardo Lee, Gregg A Helt, Justin T Reese, Monica C Munoz-Torres, Chris P Childers, Robert M Buels, Lincoln Stein, Ian H Holmes, Christine G Elsik, and Suzanna E Lewis. Web Apollo: a web-based genomic annotation editing platform. *Genome biology*, 14(8):R93, 2013.

[LK12] Gwanhoo Lee and Young Hoon Kwak. An open government maturity model for social media-based public engagement. *Government information quarterly*, 29(4):492–503, 2012.

[LLH$^+$15] Weisong Liu, Stanley JF Laulederkind, G Thomas Hayman, Shur-Jen Wang, Rajni Nigam, Jennifer R Smith, Jeff De Pons, Melinda R Dwinell, and Mary Shimoyama. OntoMate: a text-mining tool aiding curation at the Rat Genome Database. *Database: The Journal of Biological Databases and Curation*, 2015.

[LLS$^+$13] Stanley JF Laulederkind, Weisong Liu, Jennifer R Smith, G Thomas Hayman, Shur-Jen Wang, Rajni Nigam, Victoria Petri, Timothy F Lowry, Jeff de Pons, Melinda R Dwinell, et al. PhenoMiner: quantitative phenotype curation at the Rat Genome Database. *Database: The Journal of Biological Databases and Curation*, 2013.

[LNZA06] Rasko Leinonen, Francesco Nardone, Weimin Zhu, and Rolf Apweiler. UniSave: the UniProtKB sequence/annotation version database. *Bioinformatics*, 22(10):1284–1285, 2006.

[LW$^+$02] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.

[McQ12] Peter McQuilton. Opportunities for text mining in the FlyBase genetic literature curation workflow. *Database: The Journal of Biological Databases and Curation*, 2012.

[MGBRS$^+$16] Peter McQuilton, Alejandra Gonzalez-Beltran, Philippe Rocca-Serra, Milo Thurston, Allyson Lister, Eamonn Maguire, and Susanna-Assunta Sansone. BioSharing: curated and crowd-sourced metadata standards, databases and data policies in the life sciences. *Database: The Journal of Biological Databases and Curation*, 2016.

[MHA$^+$15] James Myers, Margaret Hedstrom, Dharma Akmon, Sandy Payette, Beth A Plale, Inna Kouper, Scott McCaulay, Robert McDonald, Isuru Suriarachchi, Aravindh Varadharaju, et al. Towards sustainable curation and preservation: The sead project's data services approach. In *e-Science (e-Science), 2015 IEEE 11th International Conference on*, pages 485–494. IEEE, 2015.

[MM02] Stephen Marshall and Geoff Mitchell. An e-learning maturity model. In *Proceedings of the 19th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education, Auckland, New Zealand*, 2002.

[MMC09] Anja Maier, James Moultrie, and P John Clarkson. Developing maturity grids for assessing organisational capabilities: Practitioner guidance. In *4th International Conference on Management Consulting: Academy of Management*, 2009.

[MPG$^+$17] Luc Mottin, Emilie Pasche, Julien Gobeill, Valentine Rech de Laval, Anne Gleizes, Pierre-André Michel, Amos Bairoch, Pascale Gaudet, and Patrick Ruch. Triage by ranking to support the curation of protein interactions. *Database: The Journal of Biological Databases and Curation*, 2017.

[NDM$^+$13] Mariana Neves, Alexander Damaschun, Nancy Mah, Fritz Lekschas, Stefanie Seltmann, Harald Stachelscheid, Jean-Fred Fontaine, Andreas Kurtz, and Ulf Leser. Preliminary evaluation of the CellFinder literature curation pipeline for gene expression in kidney cells and anatomical parts. *Database: The Journal of Biological Databases and Curation*, 2013.

[NSW$^+$09] Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L Rubin, Margaret-Anne Storey, Christopher G Chute, et al. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic acids research*, 37(suppl_2):W170–W173, 2009.

[OAA$^+$13] Sandra Orchard, Mais Ammari, Bruno Aranda, Lionel Breuza, Leonardo Briganti, Fiona Broackes-Carter, Nancy H Campbell, Gayatri Chavali, Carol Chen, Noemi Del-Toro, et al. The MIntAct projectIntAct as a common curation platform for 11 molecular interaction databases. *Nucleic acids research*, 42(D1):D358–D363, 2013.

[OHO09] Martin H Ofner, Kai M Huener, and Boris Otto. Dealing with complexity: a method to adapt and implement a maturity model for corporate data quality management. *AMCIS 2009 Proceedings*, page 491, 2009.

[OIH17] Christian O'Reilly, Elisabetta Iavarone, and Sean L Hill. A Framework for Collaborative Curation of Neuroscientific Literature. *Frontiers in neuroinformatics*, 11:27, 2017.

[OKA$^+$12] Sandra Orchard, Samuel Kerrien, Sara Abbani, Bruno Aranda, Jignesh Bhate, Shelby Bidwell, Alan Bridge, Leonardo Briganti, Fiona SL Brinkman, Gianni Cesareni, et al. Protein interaction data curation: the International Molecular Exchange (IMEx) consortium. *Nature methods*, 9(4):345, 2012.

[OOÖ13] Martin Ofner, Boris Otto, and Hubert Österle. A maturity model for enterprise data quality management. *Enterprise Modelling and Information Systems Architectures-An International Journal: Vol. 8, Nr. 2*, 2013.

[PAM$^+$17] Sylvain Poux, Cecilia N Arighi, Michele Magrane, Alex Bateman, Chih-Hsuan Wei, Zhiyong Lu, Emmanuel Boutet, Hema Bye-A-Jee, Maria Livia Famiglietti, Bernd Roechert, et al. On expert curation and scalability: UniProtKB/Swiss-Prot as a case study. *Bioinformatics*, 33(21):3454–3460, 2017.

[PCCW93] Mark C Paulk, Bill Curtis, Mary Beth Chrissis, and Charles V Weber.

Capability maturity model, version 1.1. *IEEE software*, 10(4):18–27, 1993.

[PCM$^+$14] Monica Poelchau, Christopher Childers, Gary Moore, Vijaya Tsavatapalli, Jay Evans, Chien-Yueh Lee, Han Lin, Jun-Wei Lin, and Kevin Hackett. The i5k Workspace@ NALenabling genomic data access, visualization and curation of arthropod genomes. *Nucleic acids research*, 43(D1):D714–D719, 2014.

[PCT$^+$12] Lakshmi Pillai, Philippe Chouvarine, Catalina O Tudor, Carl J Schmidt, K Vijay-Shanker, and Fiona M McCarthy. Developing a biocuration workflow for AgBase, a non-model organism database. *Database: The Journal of Biological Databases and Curation*, 2012.

[PLW02] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[PMA$^+$14] Sylvain Poux, Michele Magrane, Cecilia N Arighi, Alan Bridge, Claire ODonovan, and Kati Laiho. Expert curation in UniProtKB: a case study on dealing with conflicting and erroneous data. *Database: The Journal of Biological Databases and Curation*, 2014.

[PR11] Jens Pöppelbuß and Maximilian Röglinger. What makes a useful maturity model? a framework of general design principles for maturity models and its demonstration in business process management. In *ECIS*, page 28, 2011.

[RBNR$^+$14] Rafal Rak, Riza Theresa Batista-Navarro, Andrew Rowley, Jacob Carter, and Sophia Ananiadou. Text-mining-assisted biocuration workflows in Argo. *Database: The Journal of Biological Databases and Curation*, 2014.

[RDS$^+$13] Fabio Rinaldi, Allan Peter Davis, Christopher Southan, Simon Clematide, Tilia Renate Ellendorff, and Gerold Schneider. ODIN: a customizable literature curation tool. In *Proceedings of the Fourth BioCreative Challenge Evaluation Workshop*, volume 1, pages 219–223, 2013.

[RFC15] Fabio Rinaldi, Lenz Furrer, and Simon Clematide. Large-scale Information Extraction for Assisted Curation of the Biomedical Literature. In *IT@ LIA@ AI* IA*, 2015.

[RHL+14] Kim M Rutherford, Midori A Harris, Antonia Lock, Stephen G Oliver, and Valerie Wood. Canto: an online tool for community literature curation. *Bioinformatics*, 30(12):1791–1792, 2014.

[RLGC+17] Fabio Rinaldi, Oscar Lithgow, Socorro Gama-Castro, Hilda Solano, Alejandra López-Fuentes, Luis José Muñiz Rascado, Cecilia Ishida-Gutiérrez, Carlos-Francisco Méndez-Cruz, and Julio Collado-Vides. Strategies towards digital and semi-automated curation in RegulonDB. *Database: The Journal of Biological Databases and Curation*, 2017.

[RPM16] Carlo Ravagli, Francois Pognan, and Philippe Marc. OntoBrowser: a collaborative tool for curation of ontologies by subject matter experts. *Bioinformatics*, 33(1):148–149, 2016.

[RRML17] KE Ravikumar, Majid Rastegar-Mojarad, and Hongfang Liu. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. *Database: The Journal of Biological Databases and Curation*, 2017.

[RW16] Maja Rey and Ulrike Wittig. Proposal for BLAHmuc 2016: Text Mining to Support Data Curation for SABIO-RK. 2016.

[SBI+13] Michael Stonebraker, Daniel Bruckner, Ihab F Ilyas, George Beskales, Mitch Cherniack, Stanley B Zdonik, Alexander Pagan, and Shan Xu. Data Curation at Scale: The Data Tamer System. In *CIDR*, 2013.

[SBR+06] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. BioGRID: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl_1):D535–D539, 2006.

[SC08] Sarah L Shreeves and Melissa H Cragin. Introduction: Institutional repositories: Current state and future. *Library Trends*, 57(2):89–97, 2008.

[SHC$^+$13] Reza M Salek, Kenneth Haug, Pablo Conesa, Janna Hastings, Mark Williams, Tejasvi Mahendraker, Eamonn Maguire, Alejandra N González-Beltrán, Philippe Rocca-Serra, Susanna-Assunta Sansone, et al. The MetaboLights repository: curation challenges in metabolomics. *Database: The Journal of Biological Databases and Curation*, page bat029, 2013.

[SLC$^+$15] Pedro Sernadela, Pedro Lopes, David Campos, Sérgio Matos, and José Luís Oliveira. A Semantic Layer for Unifying and Exploring Biomedical Document Curation Results. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 8–17. Springer, 2015.

[SO17] Pedro Sernadela and José Luís Oliveira. A semantic-based workflow for biomedical literature annotation. *Database: The Journal of Biological Databases and Curation*, 2017.

[SPC14] Dina Salah, Richard Paige, and Paul Cairns. An evaluation template for expert review of maturity models. In *International Conference on Product-Focused Software Process Improvement*, pages 318–321. Springer, 2014.

[SSL16] Ayush Singhal, Michael Simmons, and Zhiyong Lu. Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. *PLoS computational biology*, 12(11):e1005017, 2016.

[SVLB$^+$15] Dóra Szakonyi, Sofie Van Landeghem, Katja Baerenfaller, Lieven Baeyens, Jonas Blomme, Rubén Casanova-Sáez, Stefanie De Bodt, David Esteve-Bruna, Fabio Fiorani, Nathalie Gonzalez, et al. The KnownLeaf literature curation system captures knowledge about Arabidopsis leaf growth and development and facilitates integrated data mining. *Current Plant Biology*, 2:1–11, 2015.

[SZ13] Sulayman K Sowe and Koji Zettsu. The architecture and design of a community-based cloud platform for curating big data. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference on*, pages 171–178. IEEE, 2013.

[The13] The Institute of Internal Auditors. Practice guide: Selecting, using and creating maturity models: a tool for assurance and consulting engagements, 2013.

[TLL$^+$14] Manabu Torii, Gang Li, Zhiwen Li, Rose Oughtred, Francesca Diella, Irem Çelen, Cecilia N Arighi, Hongzhan Huang, K Vijay-Shanker, and Cathy H Wu. RLIMS-P: an online text-mining tool for literature-based extraction of protein phosphorylation information. *Database: The Journal of Biological Databases and Curation*, 2014.

[TLS$^+$02] Simon Twigger, Jian Lu, Mary Shimoyama, Dan Chen, Dean Pasko, Hanping Long, Jessica Ginster, Chin-Fu Chen, Rajni Nigam, Anne Kwitek, et al. Rat Genome Database (RGD): mapping disease onto the genome. *Nucleic acids research*, 30(1):125–128, 2002.

[TSVS10] Catalina O Tudor, Carl J Schmidt, and K Vijay-Shanker. eGIFT: mining gene information from the literature. *BMC bioinformatics*, 11(1):418, 2010.

[TTR16] Ayca Tarhan, Oktay Turetken, and Hajo A Reijers. Business process maturity models: A systematic literature review. *Information and Software Technology*, 75:122–134, 2016.

[Uni15] UniProt. Evidences. `www.uniprot.org/help/evidences`, 2015. [Online; accessed 23-May-2016].

[UPR$^+$14] Martin Urban, Rashmi Pant, Arathi Raghunath, Alistair G Irvine, Helder Pedro, and Kim E Hammond-Kosack. The Pathogen-Host Interactions database (PHI-base): additions and future developments. *Nucleic acids research*, 43(D1):D645–D655, 2014.

[VASM$^+$14] Kimberly Van Auken, Mary L Schaeffer, Peter McQuilton, Stanley JF Laulederkind, Donghui Li, Shur-Jen Wang, G Thomas Hayman, Susan Tweedie, Cecilia N Arighi, James Done, et al. BC4GO: a full-text corpus for the BioCreative IV GO task. *Database: The Journal of Biological Databases and Curation*, 2014.

[VJYC$^+$13] Karin Verspoor, Antonio Jimeno Yepes, Lawrence Cavedon, Tara McIntosh, Asha Herten-Crabb, Zoë Thomas, and John-Paul Plazzer.

Annotating the biomedical literature for the human variome. *Database: The Journal of Biological Databases and Curation*, 2013.

[WHL$^+$12] Chih-Hsuan Wei, Bethany R Harris, Donghui Li, Tanya Z Berardini, Eva Huala, Hung-Yu Kao, and Zhiyong Lu. Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts. *Database: The Journal of Biological Databases and Curation*, 2012.

[WKL13] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522, 2013.

[WMM$^+$13] Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, et al. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic acids research*, 42(D1):D1001–D1006, 2013.

[WOG$^+$17] Honghan Wu, Anika Oellrich, Christine Girges, Bernard de Bono, Tim JP Hubbard, and Richard JB Dobson. Automated PDF highlighting to support faster curation of literature for Parkinsons and Alzheimers disease. *Database: The Journal of Biological Databases and Curation*, 2017.

[WS96] Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

[YV13] Antonio Jimeno Yepes and Karin Verspoor. Towards automatic large-scale curation of genomic variation: improving coverage based on supplementary material. *BioLINK SIG*, pages 39–43, 2013.

[ZBNDA17] Chrysoula Zerva, Riza Batista-Navarro, Philip Day, and Sophia Ananiadou. Using uncertainty to link and rank evidence from biomedical literature for model curation. *Bioinformatics*, 33(23):3784–3792, 2017.

[ZBP17] I Zaabar, Y Beauregard, and M Paquet. Evaluation and validation of a maturity model in obsolescence management for increased performances. 2017.

Appendix A

The Interview Questions

These questions are about evaluating the BIOC-MM, Bio-curation Maturity Model, which is inspired by [SPC14].

1. Do you think BIOC-MM covers all aspects of the bio-curation process?
2. Would you update the maturity levels or components? If so please explain what and why?
3. What do you think about the BIOC-MM and its usefulness?

Appendix B

The first published version of the BIOC-MM

The table below shows the first published BIOC-MM in ICBO 2017.

Component	Level 1	Level 2	Level 3	Level 4	Level 5
Adding and editing repository data	Manually identify problems in the data records and fix them	- Define criteria to go through each data record and fix data - Adding annotations when editing data (manually)	Semi-automatic tool to detect problems in data and suggest solutions to fix problems. The curator can then go through suggestions and authorise the ideal suggestion	- Providing a catalog that link all types of annotations - Collaboration and Data Sharing providing a common curation platform to share curation efforts between databases	Completely automated way to detect and fix problems in data

Searching and choosing for new literature	Check for new publications in the literature manually	Semi-automated tool to search for literature	The tool can rank and order the extracted literature	Set the tool to work every specific period of time, and search in different sources of literature	Totally automated way to search literature and split the extracted papers by type
Reading and extracting data from the abstract	Reading and extracting data manually	Collaboration allow the authors of new publication to participate partially in the curation process	Semi-automated tool to highlight and extract	The tool can also semi-automatically find protein-protein interaction and relationship	The tool can perform its job automatically
Reading and extracting data from the full-text	Reading and extracting data manually	A tool to asses manual curation	Collaboration collaborative curation platform between communities and curators	A tool to extract data from text semi-automatically	Extend the tool, so it covers tables, figures etc. At least point out if it has something that need to be reviewed
Documenting Curation Results	Does not pay attention for documenting any results	A semi-automatic tool to help in extracting results of the curation for a specific type of data	The tool has extra feature such as specifying the period of time	The tool will display the reason	A tool to analyse the curation results

Table B.1: BIOC-MM, a Maturity Model for Curation of Biomedical Databases - first version.

Appendix C

The second version of the BIOC-MM

-	Level 1: Ad-hoc Curation	Level 2: Standardised Curation	Level 3: Curation at scale	Level 4: Collaborative Curation	Level 5: Analytical Curation
Literature-Based Curation	Ad-hoc, manual procedures used for ranking, selecting and curating publications.	Uses agreed-upon approaches for selecting publications for curation and for extracting basic annotations from the text.	Organises and prioritises the large literature to be curated. Uses automation and data sharing platforms to amplify the curation work that can be done by the available curators.	Uses collaborative curation platforms. Deskills curation through the use of tools.	Uses the reviewers' feedback on the publications and use it in curation. Tools to consider curation needs earlier in the publication life cycle.

Data-Based Curation	Uses ad hoc, manual procedures for identifying problems in data and fixing them.	Uses agreed, documented procedures for finding and fixing problems in data.	Uses agreed-pon approaches for detecting errors and suggesting corrections, and the curators need to authorise the process.	Consistent use of standard ontologies for data representation. Linking data records to other external biomedical data.	Using the results of curation process to find patterns in curation. Uses agreed on automatic tools that diagnose quality problems in data, fix them and generate reports about them.
Quality Assurance	Ad hoc, sporadic attempts to fix problems with curation processes.	Providing an audit trail for curation. A standard curation process is followed by all curators. Guidelines for the curation process and other related issues are documented.	A standard data representation is followed. Automated curation of accepted lower quality for data that cannot be manually curated.	Sharing best practice curation processes with other communities and adopting good ideas.	Automatic gathering of training data from work of existing curators. Tools to identify good curation practices from audit trails.

Community Building	Ad-hoc procedures used for curation, with no attempt to standardise throughout the community.	Mechanisms for users to give feedback, report errors and supply requirements. Clear, documented guidelines for submitters of data, etc.	Standardises the data submission to the community, so data comes in partially curated. Proactively gather priorities from community.	A shared and collaborative environment is provided between related biomedical communities.	Documents the results of curation process and provides analysis for them to improve the curation process. Tracks usage of data to identify curation priorities automatically.

Table C.1: BIOC-MM, the Maturity Model for Curation of Biomedical Databases - second version.