MANCHESTER 1824
The University of Manchester

The University of Manchester Research

Amplifying Data Curation Efforts to Improve the Quality of Life Science Data

DOI:
10.2218/ijdc.v12i1.495

Document Version
Final published version

Link to publication record in Manchester Research Explorer

Citation for published version (APA):
Alqasab, M., Embury, S., & Sampaio, S. (2017). Amplifying Data Curation Efforts to Improve the Quality of Life Science Data. *International Journal of Digital Curation*, *12*, 1-12. Article 1. https://doi.org/10.2218/ijdc.v12i1.495

Published in:
International Journal of Digital Curation

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

OPEN ACCESS

Download date:29. Nov. 2023

IDCC17 | *Research Paper*

Amplifying Data Curation Efforts to Improve the Quality of Life Science Data

Mariam Alqasab
University of Manchester

Suzanne M. Embury
University of Manchester

Sandra de F. Mendes Sampaio
University of Manchester

Abstract

In the era of data science, and data-driven science, data sets are shared widely and used for many purposes unforeseen by the original creators of the data. In this context, defects in data sets can have far reaching consequences, spreading from data set to data set, and affecting the consumers of that data in ways that are hard to predict or quantify. Some form of waste is typically the result. For example, scientists using defective data to propose promising hypotheses for experimentation may waste their limited wet lab resources chasing the wrong experimental targets. Scarce drug trial resources may be used to test drugs that actually have little chance of giving a cure.

Because of this, database owners care about providing high quality data. Automated curation tools can be used to an extent to discover and correct some forms of defect. But, in some areas, human curation, performed by highly-trained domain experts, is needed to ensure that the data represents our current interpretation of reality accurately. Human curators are expensive, and there is far more curation work to be done than there are curators available to perform it. Tools and techniques are needed to enable the full value to be obtained from the curation effort current available.

In this paper, we explore one possible approach to maximising the value obtained from human curators, by automatically extracting information about data defects and corrections from the work that the curators do. This information is packaged in a source independent form, to allow it to be used by the owners of other databases (for which human curation effort is not available or is insufficient) to find out if the same defects are present in their data or not. This amplifies the efforts of the human curators, allowing their work to be applied to other sources, without requiring any additional effort or change in their processes or tool sets. We show that this approach can discover significant numbers of defects, which can also be found in other sources.

Received date | *Revision received* date | *Accepted* date

Correspondence should be addressed to Mariam Alqasab, M13 9PL, United Kingdom Email: mariam.alqasab@postgrad.manchester.ac.uk

The 12th International Digital Curation Conference takes place on 20–23 February 2017 in Edinburgh. URL: http://www.dcc.ac.uk/events/idcc17/

Copyright rests with the authors. This work is released under a Creative Commons Attribution 4.0 International Licence. For details please see http://creativecommons.org/licenses/by/4.0/

Introduction

Many database owners make their data available on the Web for others to use. While in most cases no assurance is provided on the quality of Web data, data is generally considered as trustworthy if its quality level is high. Some people deem data that does not contain errors as data of high quality, but this is often not the case, particularly when other aspects of data quality are not taken into account.

The data in a database are not fix, as the database providers will make changes if it requires. As part of keeping data in a high data quality level, databases providers use automatic and manual curation to improve their data quality. Some defects in data can be detected using automatic tools. These automatic tools can normally deal with general data defects such as completeness and uniqueness. For example, checking if all required fields contain data and are not empty. However, these automatic tools tend to be limited to systematic errors, this makes the rest of non-systematic errors to remain unsolved. The non-systematic errors require knowledge and expertise in the domain to be detected and solved. For example, in biomedical area, information about a protein need a human expert and analysis to be extracted if a new publication is published in the area.

In many domains, this expertise is not widely available and is not free. Domain experts have to spend much time to search for defects in data and fix them. Some experts may volunteer to spend some of their time to curate data in their domain of interest, in other area, owners of core community resources have had to hire full-time data curators to maintain the accuracy, concurrency, completeness and consistency of their data. For example, UniProt has a number of data curators, where are paid as full-time job to manually curate UniProt data. However, there are many smaller data sources are managed on a voluntary basis, and have no funding to employ data curation. This resulted in growing out of data for these database, overtime, with respect to the information in the curated database, even in areas where their contents overlap such as biomedical scientists may have protein data and these data changed, which cause data held by the scientist to be out-of-date.

We would like to find an automated way to package the data curation work in a form that can be cheaply and easily applied to these other databases and data consumers without requiring any extra work from the data curators. In this paper, we presents, the IQBot, an approach to solve this problem. Our aim is to find a way that can detect and extract changes in a curated database and share them. We also aim to predicate and provide the reason behind each change in order to give the choice for those who would like to apply the changes to determine whether to follow the new changes or not depends on the provided reason for the change.

Related Work

As database consumers care about using high quality data to produce correct results, this leads database providers to care more about their data quality, and try to keep their data in high quality all the time. To maintain the quality of the data, the database providers use different ways to curate their data. This makes

a number of research paid attention to data curation. The research vary between semi-automatic tools, that require interacting with human experts, and completely automatic tools for curation.

Abrams et al. provided a tool for database providers to share their data with others by storing them in a Merritt repository. Each time a database provider make changes and curate their data, they update the version stored in the repository. By updating the repository version, this will help database users to get the updated version of data. This is done by providing a graphical user interface to make it easier to the database consumers to find and download datasets, which match their need (Abrams et al., 2014).

Ravagli et al. proposed, OntoBrowser , a collaborative tool for curation of ontologies. The tool basically allow the curators to work with a single copy of data, and the data are stored in a central database to avoid redundancy (Ravagli, Pognan, and Marc, 2016). The tool also can pre-map unmapped ontology terms automatically by using fuzzy matching., which are been added by the curators, which have been added by the curators.

Some curation tools do not provide end to end automatic tool as they require human to participate in the curation process. Bunt et al. proposed a semi-automatic curation tool that basically sends e-mails to authors, who have new publication in the area, and ask them to fill in a specific form. In return, the provided information will help curators to faster the curation process, as they will have all the basic information to easily identify the data that need to be curated (Bunt et al., 2012). However, authors response to e-mails is a voluntarily task, which means that authors have the choice whether to help by filling in the form or not. We cannot guarantee that all authors, who received e-mails from the tool, will response and participate in the process.

Other research proposed different approaches for automatic data curation tools. Kumari et al. evaluated tools, which are based on ON-Demand Curation, by providing a case study to see how the presentation of the data in a database can affect the performance of the curation. They claimed that if the amount of the provided data has many details, it will increase the cost of the curation, otherwise, if the provided data is very little, then their will be potential of invalid curation. The authors also suggested user interface guidelines to add more efficiency to the tools. (Kumari, Achmiz, and Kennedy, 2016).

Some research tried to find ways to minimize the time curators spend to curate data by reusing and sharing curation efforts with others. Orchard et al. proposed the International Molecular Exchange (IMEx) which provides interaction between all participated protein databases to share their curation efforts (Orchard et al., 2012). This is done by assigning a journal to each partner database to curate data depends on the publication of the assigned journal. The curators need to follow specific curation rules in order to make the curation work applicable to other databases. Another research, called MIntAct project, also focused on sharing curation efforts between databases by providing a common curation platform (Orchard et al., 2013). Although IMEx and MIntAct are sharing curation efforts with other databases, the sharing of curation efforts is limited between the databases linked to the tool.

The literature were focused on providing different tools to curate data to

improve data quality. The tools used different approaches to curate data whether by interacting with human to participate in the curation process, reusing and sharing curation efforts. However, when sharing curation efforts, curation work is shared only between the databases participated on their frame of work. However, our aim focus more on producing a way that works as a bridge between original data sources and consumers of the data. A way that detects changes in data and provides information regarding the detected changes automatically without requiring curators for extra efforts.

Extracting Defects from Changes

To achieve our aim to maximising the use of the curation work made to a curated database, we will tackle a component, called IQBot, to extract defects in data, find the corrections of the defects, and infer the reason for the change. In this section, we will focus on describing how the IQBot works in general. In later section, we will explain with example how the IQBot differs if it is applied to a specific curated database.

The general algorithm of the IQBot

The main job of the IQBot is to find defects in data and drive corrections of the defects, and provide the reason behind the defects corrections. Besides, the IQBot does not focus only on the current changes occurred to data. But, it also extract the full history of changes for data during its life time. However, before we start explaining the general algorithm of the IQBot, four things need to be clarified about the curated database, that will be used:

1. As the IQBot job is to extract changes in data by comparing the content of two consecutive versions of the data. To do that the curated database needs to have at least two versions of data to be able to find changes between versions. In case if the chosen curated database does not provide previous versions of the data, the user of the IQBot can alternatively take snapshot of the curated database periodically in order to have multiple versions of the database, then use these versions with the IQBot.

   It should be noted that it is not part of the IQBot job to check whether the assigned curated database has multiple versions or not. This type of information can be provided by the publisher of the curated database and it is outside the scoop of this paper.

2. Defining the way of accessing the curated database, as not all databases follow the same way of accessing the data. A number of database providers provide automatic access to their data through the web such as providing a specific API, and they normally provide documentation in how to use it to access their database. However, other only allow you to download the database to access their data, in this case, you need to manually define the way of accessing each version of the database.

3. The curators make a number of changes to different data. In this case, we

need to determine which data we would like to find changes for. The user of the IQBot can freely choose which data to be extracted according to their preference.

4. It is important to define the way of extracting the data selected in the previous point, because different databases use different structures of presenting their data such as text files or relational tables. For example, UniProt database provides their data in a text file format, where each line contains a specific protein information, but Mouse Genome Informatics database provides its data in a table structure.

The four points mentioned above have to be answered about the curated database, before it can be monitored by the IQBot.

Extracting changes using IQBot

Algorithm 1 shows the pseudocode of the IQBot and how it works in general. To start with, the IDs for all records in a database need to be fed to the IQBot to access the records data. The IQBot then deals with each record individually to find changes in data. Mainly, it compares data from two consecutive versions of the record. If the data remains the same in both versions, then no change is detected and the process will continue to check the rest of the record versions until it find a change or reach the last version of the record. Otherwise, if a change is found, then the IQBot will move to the next step, which is finding the reason for the change.

Algorithm 1 The general algorithm of the IQBot

```
getRecordsID()
for each protein ID do
    read record data
    data1 = extract data
    while not reached last version of the record do
        read previous record version data
        data2 = extract data
        if data1 != data2 then
            type = findChaneType(data1, data2)
            reason = findTheReasonForTheChange(type, data1, data2)
            store(ID, data1, data2, reason)
            swap(data1, data2)
        end if
    end while
end for
```

Before going into details in the next step, it should be cleared that when the IQBot monitors a curated database for the first time, it will extract changes in data for all the available versions of the records. The full history is extracted to give the opportunity for the curated database consumers to track-back all changes done for the data. This will help the database consumers, who have very old version of data, to be able to know about all the forms the data changes for. As mentioned previously, extracting the full history is conducted only when running the IQBot

for the first time. Later, it will work if a new release of the curated database is available.

It should be noted that extracting the full history of the changes can be hard, as not all curated databases provide accessibility for their previous releases of the database. As mentioned previously, we can have a snapshot of every new release of the curated database in order to be monitored and compared to find changes later.

Finding the Reason for the Change

When the IQBot detect a change in data, then the reason behind the change needs to be assigned. However, the extracted changes need to be first classified. To classify the changes, we divided the changes into a number of types, which are varied between simple, partial and complete changes. We came up with these types by observing a number of detected changes. Our observation was made to a collection of 1499 changes detected by the IQBot (more details about the changes can be found in page 10). We refer to syntactic errors such as spelling mistakes, changing punctuations and changing letter case, as a simple change type. Normally, the reason for conducting this type of change is done according to issues related to the curated database.

However, the reasons behind partial and complete change types are domain specific reasons that need more analysis and knowledge in the domain to be specified. The reasons for the changes vary depend on the domain of the curated database, as the reasons cannot be generalized to cover different changes in different databases. To make the IQBot able to identify the reason for the change, the user of the IQBot needs to plug-in the reasons for the change.

The UniProt Protein Names IQBot

This section describes how the IQBot works to monitor a specific curated database. Currently, we tested the IQBot with UniProt 1, which is a Universal Protein resource contains a collection of protein sequences. UniProt is considered as one of the popular sources of biomedical data for scientist. It contains around 66 millions proteins entries, that vary between manual and automatic curated entries. UniProt hired a group of curators, who are experts in the biomedical domain to manually curate their data. The curators job is to read and analyse all the publication in the area, and apply their knowledge in order to detect and fix defects in data, to provide high data quality for the consumers. UniProt is used as the monitored curated database for the following reasons:

- UniProt data is manually curated by domain experts, as our aim is to focus on reusing curation efforts done by human experts.
- The database is curated in a regular base (every 4 weeks), this will allow us to test our IQBot more often.
- All previous versions of data can be accessed, this helps in finding the full history of changes. As we aim to provide all changes made to the data during

1 http://www.uniprot.org/

Table 1. ECO evidence codes used in UniProt entries and their meaning (UniProt, 2015).

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.

its lifetime.

In page 4, we explained four points that need to be known about the curated database to be monitored. As we will use UniProt to test the IQBot, we found the answers to the four points as follows:

1. As the IQBot needs to grant access to all entry versions we found that UniProt provides access to all its previous releases. This will allow the IQBot to extract the full history of changes done to data.

2. UniProt provides an entry for each protein, the entry contains all related data about a protein. Each protein entry has a unique id, known as accession number, and in order to access protein entry data we use the protein URI. The URI consists of two parts: the first part is fixed and is presented as "http://www.uniprot.org/uniprot/", and the second part contains of the protein accession number and the entry version number. For example, "http://www.uniprot.org/uniprot/P42645.txt?version=117". As each protein has at least one entry version, and a new entry version is produced if the entry data is changed.

3. UniProt provides a number of information about proteins such as protein name, publication, organism classification and others. Each protein has a unique entry that contains all its related information. As mentioned previously,the IQBot will extract changes in data and provide the full history of changes if it is applicable. Currently, we work with one type of changes to see how the IQBot works. The data we tried to find changes in is protein name, because protein name is very important to scientists, who are working with proteins, and not following the latest changes in protein name will cause problems and quality issues in their data.

Protein name change for several reasons, which will be explained later in this section. However, changing protein name is not a random work of curation, as it requires efforts from the curators as they need to use their knowledge and expertise in the domain to correctly identify the new protein name.

4. UniProt has a specific structure to present data in a protein entry. Protein data is displayed in text format, where each line represents a specific data type. A line has an initial of two letters, and each initial refers to the type of the line content. For example, the protein name can be found in line, which has the initial "DE". In our code,we extracted the protein name by accessing the protein entry and reading the content of the line with "DE" initial.

The process of comparing changes and assigning the change type remains the same as mentioned in page 4. When IQBot monitor UniProt, it follows similar steps as in algorithm 1. For each protein, the protein entry content is read to extract the current protein name. For the rest of the protein entry previous versions, the steps of extracting protein name are repeated. After that, both extracted names are compared. If the names are the same, then steps in lines 6-8 are repeated until a change is detected or last entry version is reached. However, if the names are different, then the type and reason for the change are assigned.

However, the reasons for the changes are domain specific as mentioned previously, so the reasons need to specified first. From our observation to a number of entries where name change is detected and according to UniProt Evidence (UniProt, 2015), it can be noticed that before September 2015, UniProt curators did not provide any information about the change made to data and why it was made. However, from September 2015 curators started to add Evidence Code Ontology (ECO) 2 when they make any changes to protein entries.

ECO evidence code is mainly designed to give description of the reason behind the change in biomedical research. The ECO evidence code has a collection of different terms that covers both types of curation the manual and automatic, as the terms distinguish whether the change happened in manual or automatic curation. Each ECO evidence code refers to a specific meaning. For example, if an entry contains the code ECO:0000305, it means that the reason for the change is based on the scientific knowledge of the curator, and the curators made their own analysis depends on their domain knowledge.

As mentioned previously, protein entries released on and after September 2015 include ECO evidence code, when change is made to a protein entry. However, all previous releases remain the same without any ECO evidence code. We summarize the reasons for the name changes into two periods of time, before and from September 2015 as follows:

From September 2015, already defined reasons for the change can be found in protein entry in which a name change is detected. The curators will add ECO evidence code next to the change they make to indicate the reason for the change. Currently, UniProt is using ten different ECO evidence code seven for manual curation and three for automatic curation as shown in table 1.

2 www.evidenceontology.org

Before September 2015, for name changes in this preiod of time, we cannot count on the ECO evidence code to identify the reason for the change, as it is not provided. As a result, we investigated protein entries, where name change is detected, and other resources to be able to identify the reason for the change. We found five reasons for name change from observing a collection of 1499 changes in protein names, which were extracted in the previous section. The reasons are ordered from the most common reason to the lest, as follows:

1. Protein entries are stored whether in TrEMBL or SwissProt database. TrEMBL database contains protein entries which are automatically curated, where SwissProt database has the manually curated entries.

   In our sample set of 1499 protein names, we found that the majority of name changes are made when an entry is moved from TrEMBL to SwissProt, database. That means, the protein entry first receives a manual curation. Normally, in the first manual curation the curator reviews all the information in an entry and fix errors including changing protein name if requires. In our code, we automatically check if the protein name changed due to first manual curation by comparing the name of the database, where the entry is stored, for both entry versions the one where name change is detected and the previous version. If the databases are different, then it means that the entry is curated manually for the first time.

   We also generated association rules using information collected from protein entries, where name change is detected. Basically, we monitored which changes occurred to entry data besides name change. We analyzed the resulted association rules and we found that a name change happened when a protein entry is curated manually for the first time by a curator.

2. There are a number of cases where protein name changed due to merging protein entries. Merging protein entries means combining a number of protein entries into one entry. In manual curation, curators are required to follow the database curation guidelines, and one of these guidelines is to merge all entries, which have the same gene name or same species into one entry. This resulted in changing the protein name (UniProt, 2014). We checked this by comparing the value of the accession number in both versions the one where the name changed is detected and the previous one, then if the recent version has more than in the previous entry accession numbers, then it means a merge is made.

3. Curators manually reviewed the literature in proteins, and when they find new publication related to a protein they make changes to that protein entry if requires such as changing protein name. In our code, we look at publication section in the protein entry version where name change is detected. If it differs than the publication section in the previous entry version, then it means that protein name change due to new publication in the area.

4. It should be noticed that most of the cases we came a cross when some parts of the name were removed, and the removed part was added as

flags to the protein entry. After investigating, we realized that this change occurred due to changes in the UniProt naming guidelines 3. The guidelines does not allow some terminologies to be used in protein names. For example, if the protein name contains the word "Putative", then the word should be removed from the protein name and added to a section in the entry called flags.

5. In rare situations, nothing from the mentioned reasons can be found in the entry data, where name change is found. From reviewing (UniProt, 2014) it can be said that the curators made changes to protein entries based on their knowledge, expertise and analysis of the publication in the domain. So they change protein entries data based on their domain knowledge.

Experimental Results

We claimed that IQBot can extract defects in data and find corrections for the defects. Also, provide the reasons behind the detected changes. When we used IQBot to monitor UniProt, it detected 1499 changes in protein name for 249 proteins.

We would like to draw attention to IQBot, and how it would help other databases to improve the quality of their data. As the available data in the database might be out-of-date and need to be updated. This is done by looking at the current and previous protein names, and check if a database is using the updated protein name or the old name.

In order to do this, we accessed a collection of databases, which deal with proteins, to check the availability of protein names. We used DBGET 4, which is an on-line tool that grant access to a collection of biomedical databases. Using DBGET will save time and efforts as we do not need to be familiar with the way of accessing each database individually, because DBGET provides the same way to access all provided databases. We implemented a java program that access the biomedical databases in DBGET automatically and check if each database has old or updated protein names. We searched in eight databases, which are: KEGG BRITE, GO, KEGG GENES, KEGG DGENES, KEGG ORTHOLOGY, KEGG MGENES, NCBI-Gene and KEGG ENZYME.

Algorithm 2 shows the pseudocode of finding the previous and current protein names in databases. For each database, we check the availability of both protein names (current and previous). If the database contains both protein names or only the current protein name, then it will be considered as an updated database. But, if the database has only the previous protein name, then the database will be considered as out of date database.

Figure 1 displays the results of checking the appearance of the last pair of protein names. The old refers to the previous protein name and the new refers to the current protein name. Number one means that the database contains protein

3 www.uniprot.org/docs/nameprot
4 http://www.genome.jp/dbget/

Algorithm 2 Pseudocode to access different databases to search in DBGET.

```
queryEachDB(db, currentName)
for result: results do
    if result.contain(currentName) then
        count DB as updated
    else
        queryEachDB(db, previousName)
        for result: results do
            if result.has(previousName) AND !result.has(currentName) then
                count DB as not updated
            end if
        end for
    end if
end for
```

names whether old or new names. However, number zero means that the database does not contain any protein name. Figure 1 shows that only one database, KEGG GENES, contains the most updated protein name. However, four databases have out-of-date protein names.

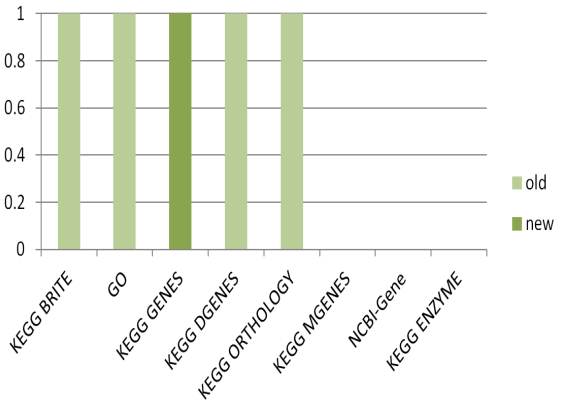

Figure 1. The results of checking the availability of old and new protein names in databases.

Conclusion and Future Work

In this paper, we explained how we would like to amplify the use of the curation efforts made to a curated database. Currently, we tested our IQBot using UniProt database. The IQBot approved that it can identify changes made by curators and it can find the reason behind the change. The are several extensions to this work, which are as follows:

ECO evidence code We mentioned previously that ECO evidence code contains a huge collection of evidence terms and uses by other biomedical databases, so we can link our model to the whole ontology of ECO evidence code. By doing

this, we aim to give the opportunity to all biomedical databases, which use ECO evidence code, to use the IQBot with minimal changes to the code. As they need only to specify which data to be extracted and how it is accessed.

Publishing the results After changes in the curated database can be automatically extracted and found the reasons behind them, it is time to publish the results in a way which make them applicable to other sources to use. Making the results available on the web will help users and owners of databases that are still using out of date data to raise the quality of their data by updating them according to the IQ-Bot results.

Creating a model for curation We also will test the IQ-Bot against different databases, we would like to try it with other non-biomedical databases to allow us to check the generality of our model.

References

Abrams, S., Cruse, P., Strasser, C., Willet, P., Boushey, G., Kochi, J., … Rizk-Jackson, A. (2014). Datashare: empowering researcher data curation. *International Journal of Digital Curation*, *9*(1), 110–118.

Bunt, S. M., Grumbling, G. B., Field, H. I., Marygold, S. J., Brown, N. H., Millburn, G. H., Consortium, F., et al. (2012). Directly e-mailing authors of newly published papers encourages community curation. *Database*, *2012*, bas024.

Kumari, P., Achmiz, S., & Kennedy, O. (2016). Communicating data quality in on-demand curation. *arXiv preprint arXiv:1606.02250*.

Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., …, Del-Toro, N., et al. (2013). The mintact project—intact as a common curation platform for 11 molecular interaction databases. *Nucleic acids research*, gkt1115.

Orchard, S., Kerrien, S., Abbani, S., Aranda, B., Bhate, J., Bidwell, S., …, Cesareni, G., et al. (2012). Protein interaction data curation: the international molecular exchange (imex) consortium. *Nature methods*, *9*(4), 345–350.

Ravagli, C., Pognan, F., & Marc, P. (2016). Ontobrowser: a collaborative tool for curation of ontologies by subject matter experts. *Bioinformatics*, btw579.

UniProt. (2014). Uniprot manual curation sop. http://www.uniprot.org/docs/sop_manual_curation.pdf. [Online; accessed 30-July-2016].

UniProt. (2015). Evidences. http://www.uniprot.org/help/evidences. [Online; accessed 23-May-2016].