Bioinformatics, 32(6), 2016, 918–925
doi: 10.1093/bioinformatics/btv644
Advance Access Publication Date: 10 November 2015
Original Paper

OXFORD

Databases and ontologies

Flexible data integration and curation using a graph-based approach

Samuel Croset*, Joachim Rupp and Martin Romacker

Roche Innovation Center Basel, F. Hoffmann-La Roche AG, CH-4070 Basel, Switzerland

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on June 5, 2015; revised on October 20, 2015; accepted on October 21, 2015

Abstract

Motivation: The increasing diversity of data available to the biomedical scientist holds promise for better understanding of diseases and discovery of new treatments for patients. In order to provide a complete picture of a biomedical question, data from many different origins needs to be combined into a unified representation. During this data integration process, inevitable errors and ambiguities present in the initial sources compromise the quality of the resulting data warehouse, and greatly diminish the scientific value of the content. Expensive and time-consuming manual curation is then required to improve the quality of the information. However, it becomes increasingly difficult to dedicate and optimize the resources for data integration projects as available repositories are growing both in size and in number everyday.

Results: We present a new generic methodology to identify problematic records, causing what we describe as 'data hairball' structures. The approach is graph-based and relies on two metrics traditionally used in social sciences: the graph density and the betweenness centrality. We evaluate and discuss these measures and show their relevance for flexible, optimized and automated data curation and linkage. The methodology focuses on information coherence and correctness to improve the scientific meaningfulness of data integration endeavors, such as knowledge bases and large data warehouses.

Contact: samuel.croset@roche.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

With the current quantity and diversity of data available in the biomedical domain (Lipinski *et al.*, 2014; Marx, 2013), it becomes increasingly necessary to combine the information coming from multiple sources, in a variety of formats, into a unified representation. This practice goes by the name of data integration or record linkage (Winkler, 1995). Different reasons motivate the exercise: it can be for instance the appealing possibility to query and analyze information about complementary thematics (e.g. gene–disease relationship), consolidation of some knowledge existing about one topic, or the absorption of a source into another for maintenance needs (e.g. dataset coming from company acquisitions). Recently, data integration efforts have been particularly active in the drug discovery domain, with platforms such as transMART (Szalma *et al.*, 2010) or Open PHACTS (Williams *et al.*, 2012a), focused on building a pharmacological space from public repositories. Other examples include ChemSpider (Pence and Williams, 2010), a resource providing a central hub related to chemical names and structures from various sources or Identifiers.org (Juty *et al.*, 2012), a cross-reference platform for biomedical identifiers and data connections.

Fundamentally, data integration can be seen as a problem of creating the correct links between equivalent, yet disconnected, database records. This process is sometimes called 'stitching', or 'reconciliation' (Bollacker *et al.*, 2008; Dong *et al.*, 2014). Once records are rightfully associated, it becomes possible to query across or merge them if necessary. As links are created between entries, it is

intuitive to rely on an abstract graph structure to represent the problem faced. Vertices or nodes are records or entities of interest; edges represent the associations between them. An illustration of the graph abstraction is the Semantic Web: this series of standards relies on the Resource Description Framework (RDF) and a graph structure to facilitate the interoperability and integration of independent pieces of information on the World Wide Web (Berners-Lee *et al.*, 2001). The Semantic Web also provides means to establish equivalence between segregated records (*sameAs* or *exactMatch* relations), with the use of rules or reasoners for instance. Moreover, the biomedical domain is rich in unique identifiers (e.g. chemical structure identifiers like InChIKeys) and cross-references which can be used to automatically assess equivalence between database entries: if one record references another, it might be possible to deduce that the two entities are the same, for instance.

All these strategies are automatable and reliable in theory, however complications can arise quickly if cross-references are absent, errors present in the original sources or if the data is fuzzy and ambiguous by nature, such as with drug names and chemical structures for instance (Tiikkainen *et al.*, 2013; Williams *et al.*, 2012b). The effect of erroneous information is dramatic for data integration: records can get incorrectly associated with other entries, themselves recursively linked to other records. This cascade of events leads to the creation of unwanted 'hairballs', which we define as *groups of records not equivalents and not supposed to be linked, but yet connected because of the deficient state of the data*. In such a scenario, manual intervention from expert curators is required in order to improve the quality and scientific validity of the dataset. Unfortunately, with the increasing size of biomedical databases and repositories, it becomes more likely that such errors will arise by chance, and more expensive and time consuming to correct them by hand.

In this document, we use an in-house data integration project related to the creation of a drug product terminology in order to illustrate a generic and flexible approach to identify and handle problematic and erroneous records during the integration process. The drug terminology is built from millions of database entries present in eight heterogeneous sources, including four from third-party vendors (Integrity, Cortellis, PharmaProject and AdisInsight), one developed internally and three public drug databases (DrugBank, part of ChEMBL and ChEBI). Our motivation is to build and maintain an integrated database (also called *data warehouse*) from these various sources, each of which contains a part of the desired information (see Fig. 1). The graph-based method presented allows researchers to isolate and prioritize problematic entries and flexibly adjust the

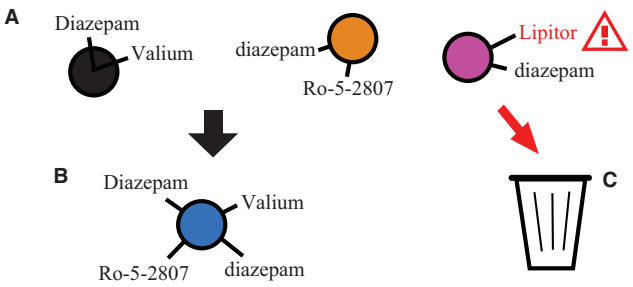

Fig. 1. The data integration challenge. (**A**) Records coming from multiple sources, represented by different colours. All these records refer to the same drug product, in this case *diazepam*, but contain only partial information about it, the synonyms. (**B**) The entries have to be integrated and merged into consolidated records. (**C**) In the process, erroneous entries have to be excluded, to improve the quality and consistency of the resulting data warehouse

curation work required over an automatic integration to maximize the quality and scientific usefulness of the data.

1.1 Limitations of previous work

The methodology presented in this manuscript differentiates itself from previous record linkage approaches by its flexibility to handle many different sources and exclude erroneous records, as well as with its capability to deal with redundant identifier types. We briefly summarize in this section the state of the art and the current limitations.

Record linkage depends on the presence of one or more common identifiers in the available datasets in order to assert equality between database entries (Hernández and Stolfo, 1998, Winkler, 2014). It can for instance be the name of a person, or in our case one of the various synonyms used to describe drug products (e.g. regulatory name, brand name). Previously published techniques can then be divided in two categories: deterministic and probabilistic approaches (Roos and Wajda, 1991). Deterministic methodologies rely on a series of custom rules to create the equivalence between the records, using techniques close to text-mining. Records are merged based on exact or similarity-based matching of the common identifiers. Such an approach is straightforward and can deal with data coming from many sources, yet it is very vulnerable to erroneous information (Wajda *et al.*, 1991). Deterministic approaches do not provide means to detect and remove mistakes in the original data and are therefore not suitable for our needs: the presence of an incorrect drug product synonym would for instance result in the mixing of two or more drug products in the same category, situation to be avoided at all costs. Probabilistic-based approaches try to link pairs of records by assigning weights to a range of identifiers and by determining a likelihood for a match to be correct (Fellegi and Sunter, 1969). These techniques have the clear limitation of handling only pairs of records, whereas we needed n data sources to be considered at once for the integration process. Moreover, these methodologies usually assumed that identifiers are independents (Wilson, 2011), which is not the case with drug names: a brand name appears for instance late in the life cycle of a product, and is always preceded by a regulatory name. Finally, none of the previous approaches are designed to exclude erroneous records, they limit themselves to determine whether there is a match or non-match between records. This feature is critical for our use-case and for data quality.

To conclude, the approach we introduce is graph-based, i.e. records are abstracted as vertices, further linked by edges, and is therefore compatible with any graph representation such as RDF or graph database for implementation. The identification of erroneous records relies on two indicators from graph theory and traditionally used in social science analysis: the graph density and the betweenness centrality. We demonstrate and evaluate their relevance in the data integration and cleaning tasks. The work presented finds its application for the automatic combination of large amounts of expensive data coming from different sources, where curation needs to be prioritized and optimized in order to generate high quality scientific content. Future use cases also include the management of datasets resulting from mergers and acquisitions, in order to integrate the new valuable information with existing internal databases and reduce the amount of maintenance work necessary.

2 Methodology

A total of eight different sources were used to build the data warehouse (i.e. the drug product terminology). The methodology for the integration of the data is decomposed in three steps, respectively

called *Link*, *Evaluate* and *Clean*. Depending on the desired outcome, it is also possible to add an extra final step called *Merge* in which records are merged together. Briefly, in the *Link* step the graph structures are formed by flexibly creating edges between the initial entries from the sources. The presence of particular label types was used to connect the records; for instance, if two records have a brand name in common, an edge between the two entries was created. This step results in the creation of many graphs, reflecting the associations between records. In the *Evaluate* step, metrics are computed over the graphs (i.e. betweenness centrality and density). From these measures, it is possible to isolate the problematic graphs and entries, which can be cleaned manually or automatically given a certain threshold (*Clean* step). Each of these steps is described in detail in this section.

2.1 Data sources and implementation

The creation of a comprehensive terminology about drug products requires the integration of multiple starting data sources, each providing a subset of the wanted information (see Fig. 1). A total of 373 823 drug records are to be integrated, related to 1 881 760 labels representing the possible synonyms a drug can have: laboratory code, generic name, brand name, identifier, cross-reference, etc. (see Supplementary Fig. 1 for full scope). Three of these sources are public databases: DrugBank (11 584 drug records), ChEBI (53 185 drug records) and ChEMBL (84 901 drug records, entries with synonyms only). Alongside these, we also used four private feeds from various vendors: Thomson Reuters Integrity (87 561 drug records, including only entries with a generic or brand name), Thomson Reuters Cortellis (57 410 drug records), PharmaProjects (64 206 drug records) and AdisInsight (32 633 drug records). The last data source is an internal repository containing the drugs developed within the company alongside their synonyms and laboratory codes (5 312 drug entries). This starting set of sources illustrates one of the data integration challenges for pharmaceutical companies: information comes with different level of privacy and confidentiality and often requires a custom in-house solution. All sources were updated the 17th of September 2015 in different formats (XML, RDF, SQL) and loaded in a Java web application built using the Play! framework 1.2.7 using MongoDB as back-end store.

For our use-case, each drug entry is abstracted as a vertex (also called *node*) and has a list of one or more labels related to it. These labels will in turn be used to create the links between the records. As a result, a graph structure is created, over which it is possible to derive various measures.

2.2 Graph measures

Connected data can be abstracted as graph structures, and a data integration problem can be seen as correctly creating undirected links or edges between pairs of records or vertices. The links are created following a certain condition, for instance given the presence of a cross-reference, or using labels shared by records. One of the main benefits of the graph abstraction is the possibility to re-use the mathematical tools and descriptors of graph theory. We wanted to identify problematic records coming from S complementary yet different sources. To perform this task, we used the graph density as a theoretical measure of deviation against the perfect case (i.e. a *complete graph*, discussed later). Once problematic graphs, or 'hairballs', are identified, it is possible to isolate and clean ambiguous records using the betweenness centrality, or measure of the relative influence of a vertex within the graph based on its connectivity. This section introduces these various graph descriptors and serves as a theoretical reference.

2.2.1 Complete graphs and data sources

A complete graph is an undirected graph in which every pair of distinct vertices are connected by an edge (Wikipedia, 2014a). Given a number V of vertices, the associated completed graph is denoted K_N. Its number E_k of edges can be calculated as follows:

$$E_k = \frac{V \times (V-1)}{2} \tag{1}$$

Complete graphs are interesting for data integration purposes; they represent a perfect agreement between records coming from different sources. For example, consider the integration of three databases, each containing some complementary and partially overlapping information, such as three resources describing drugs; one expects the entries present in one of these databases to be also equivalent to some of the records in the other resources. For instance, the entry related to the diazepam in ChEBI (CHEBI:49575) is equivalent to the entry describing the diazepam in DrugBank (DB00829) and so forth, based on the synonyms they have in common. If links were drawn between equivalent records, one would therefore expect to find triangle-shaped graphs where the equivalent records from each three databases were connected to each other. The presence of the triangle-shaped graphs gives confidence in the integration: one node from each resource is reciprocally linked to a corresponding node from each other resource, forming a densely connected network, representing an ideal data integration situation. In the case of three databases being integrated, the triangle shape corresponds to the complete graph K_3 of three vertices (see Fig. 2A).

From this specific example, it is possible to generalize the approach considering the data integration of S sources; for such cases it is expected to find complete graphs K_S, where one record from each of the S original sources is linked to every other equivalent record from the other sources. In summary, complete graphs represent the perfect cases for data integration purposes, where records represented as vertices are tightly connected together by the edges representing an equivalence relationship, with a size depending on the number of starting sources S.

2.2.2 Density

In practice, linked records often do not form complete graph structures, but have a geometry drifting away from it, due to the only partial overlap of information (e.g. non-existing cross-references or shared labels). It is possible to quantify this deviation using a metric called *density*. The density of a graph is a number ranging from 0 to 1 and reflecting how connected a graph is in regards to its maximum potential connectivity. Given an undirected graph with E edges and V vertices, the density d is defined as:

$$d = \frac{2 \times E}{V \times (V-1)} \tag{2}$$

The density can also be formulated as the ratio between actual connections, i.e. the number of existing edges in the graph (E), and the number of maximum potential connections between vertices, called PC:

$$d = \frac{E}{PC} \tag{3}$$

PC is the same as the number of edges E_k in the complete graph, as defined in Equation 1 and leading to the equality:

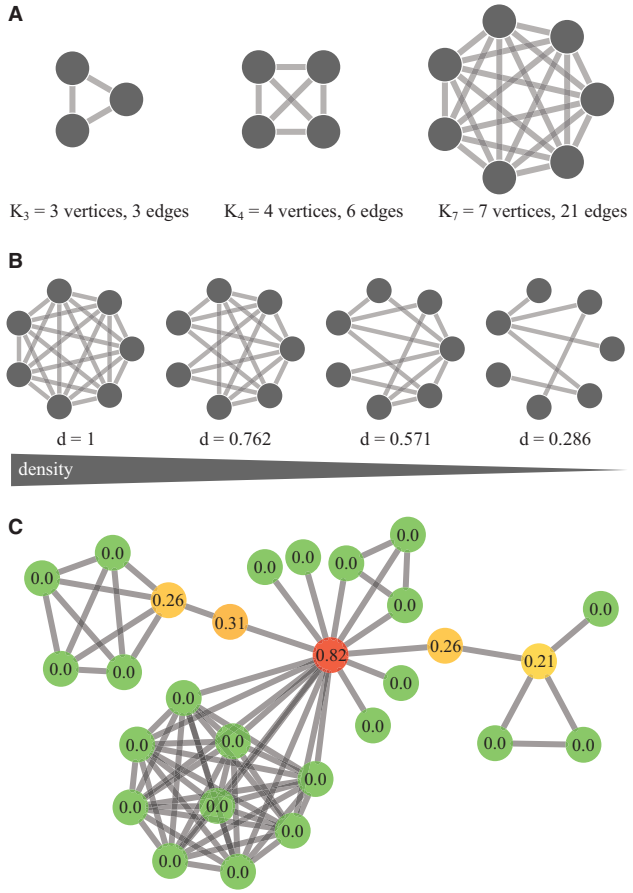

Fig. 2. Graph types and measures for data integration. (**A**) Examples of complete graph for various number of vertices. Vertices are equivalent records from different databases, linked on the basis of having synonyms in common. (**B**) Illustration of the graph density behavior. (**C**) Betweenness centrality computed for each vertex of an example graph. Vertices with low BC are in green, and high BC in red. The vertex in red connects communities of records that should be separated, and can then be excluded to improve the resolution of the information

$$d = \frac{E}{PC} = \frac{E}{E_k} = \frac{2 \times E}{V \times (V-1)} \quad (4)$$

This property is of primary interest for data integration, as the density directly reflects how far away a graph is from the ideal case of a complete graph, as outlined in the previous section. Intuitively, the graphs with the lowest densities are the ones diverging the most from an ideal scenario (i.e. perfect graph) and are therefore more suspicious. The evolution of the density for different graph structures is illustrated in Figure 2B. This indicator is used in our context to prioritize and rank the graphs from the *Link* step, in order to identify which entries are not densely linked and therefore problematic (i.e. the hairballs).

2.2.3 Betweenness centrality

The graph density helps to identify groups of problematic records, diverging from the perfect case; however, it does not provide any help to further clean problematic vertices. The betweenness centrality (BC) fulfils this task. The BC is an indicator of a vertex's centrality in a graph. Given all possible shortest paths between every pair of vertices in a graph, the betweenness centrality of a vertex, written $BC(v)$, is the fraction of the number of times a given vertex is on the shortest path:

$$BC(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \quad (5)$$

Where V is the set of vertices in a graph, $\sigma(s,t)$ is the number of shortest (s,t) paths and $\sigma(s,t|v)$ is the number of those paths passing through the vertex v other than (s,t). The measure ranges from 0 to 1, with high values reflecting a relatively high centrality of a vertex in the graph, in regards to other vertices. For data integration purposes, BC helps to identify the vertices with a prominent influence i.e. the records connecting communities or groups of records within a graph (Fig. 2C). As this measure can be potentially calculated for a very large number of vertices, likely millions, it is important to consider its computational complexity. In this regard, the computation of the BC can take $\mathcal{O}(EV)$ time for undirected graphs, where E is the number of edges and V the number of vertices (Brandes, 2001). This time is translatable as $\mathcal{O}(V^3)$ from Equation 1, assuming the worst case where all the graphs are completes. This value is below polynomial time and therefore considered as a tractable problem, which makes it in principle computationally affordable for large datasets. The BC used in combination with the graph density helps to first identify the suspicious groups and secondly to retrieve the problematic records in these hairballs. For both measures, a threshold value can be set in order to optimize the number of data points to be inspected or cleaned.

2.3 Integration process

The integration process is based on a series of steps, summarized in Figure 3. First the graph structures are created, then the graph descriptors introduced previously are computed. Then follows an iterative cleaning, based on the graph metrics and resources available; the cleaning can be automatic or manual. Each step is presented in the coming sections; the actual implementation is flexible and can vary depending on the task addressed.

2.3.1 Link

In this step, individual records or vertices are connected by edges representing an equivalence relation, i.e. a record from one source is supposedly equivalent to a record from another source. The reason behind the creation of the edge is case-specific and flexible; it can be based on a shared identifier between vertices, for instance a cross-reference, or on a series of special rules (also referred to as *deterministic approach* in the literature).

Our starting dataset did not include many cross-references; we therefore had to rely on a series of custom rules to connect the equivalent records. Our aim is to build a terminology about experimental and approved drug products; examples of use-cases involve the retrieval of all existing synonyms given a specific product or internal code. The terminology should be also suitable to search and monitor the scientific literature, among other things.

We decided to connect records with an edge if they have at least a chemical name, a generic name, a brand name, a laboratory code or a chemical structure in common. The presence of only one identical label was enough to create a link between vertices. Fuzzy matching and other string similarity-based equivalences over the label values were purposefully avoided: We are primarily interested in gathering real and unambiguous synonyms, malformed labels should be excluded by the methodology. For instance, a record containing the erroneous label 'Valuim' should be excluded and not merged with the records containing the correct spelling 'Valium'. We considered this series of rules to be specific enough for our

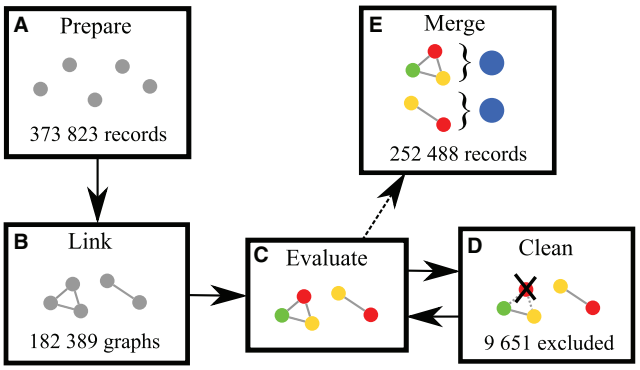

Fig. 3. Data integration process. (**A**) The records from the source databases are downloaded locally. (**B**) The graph structure is formed on top of the source records, and equivalence edges created. (**C**) The graph metrics are computed (density and betweenness centrality). (**D**) Hairballs are identified (graphs with low density) and the problematic records are excluded from the build (vertices with high betweenness centrality). (**E**) Finally, the entries present in the same graph are merged, in order to create consolidated records containing the originally segregated information (optional step)

purpose, yet it could be easily adapted for different use-cases or datasets.

Links between records are undirected in order to capture as simply as possible the fact that a pair of records are semantically identical. This relation could also be represented by a bi-directed edge, or using two directed edges starting from each vertex of the pair. However, such a representation would add unnecessary complexity to the problem as the edge directionality is not used during the computation of the graph density and BC. We outlook at the end of this document the potential use and impact of other graph metrics and representation, such as weighted edges.

2.3.2 Evaluate

The graphs created in the *Link* step vary widely in size and shape (see Section 3). Because of the fuzziness and ambiguity of the original data, some records aggregate in 'hairball-like' structures, which must be cleaned and disambiguated. The betweenness centrality and density are both helpful metrics in this regard, and they are computed in this step. We used the GraphStream library (Dutot *et al.*, 2007) to calculate the betweenness centrality (Equation (5)), the computation of the density was implemented directly in the program. We also created a series of methods to export and visualize the metrics, as illustrated in the Section 3. This step is generic and can be applied on the top of any type of graph structure.

2.3.3 Clean

Problematic and ambiguous graphs can be identified from the metrics computed in the *Evaluate* step. Depending on the time, available resources and dataset size, a threshold can be applied to flexibly isolate and classify the graphs to be further inspected. The density and the betweenness centrality are however not flawless measures: for instance, sometimes graphs with low density can be correct (false positives) or erroneous nodes can have a low betweenness centrality (false negatives). In order to find the optimal threshold values yielding the best results, we performed a receiver operating characteristic (ROC) analysis. For this purpose, both graph measures were considered as independent binary classifiers, and we manually inspected a series of cases to generate the curves. The optimal threshold values were determined using Matthews correlation coefficient (MCC). In order to find the optimal density value separating the problematic graphs from the correct ones, experts looked at 200 randomly selected graphs created in the *Link* step (see Supplementary Fig. 2). Based on this training set, the density threshold of 0.59 gave the best results at separating graphs to be inspected versus correct ones (MCC = 0.651). The optimal betweenness centrality threshold was derived from inspecting 100 random vertices coming from true problematic graphs. The betweenness centrality threshold of 0.33 gave the best results and was used to separate vertices to be excluded from vertices to be kept (MCC = 0.837, see Supplementary Fig. 3). This approach was used as we wanted most of the process to be automated; note that it is also possible to define the thresholds based on the number of cases to inspect manually or any other external parameters. The *Clean* and *Evaluate* steps can be repeated at will until a satisfactory state of the data is reached. We repeated them four times before merging the records (see Supplementary Fig. 4 for effect of cleaning steps on the dataset).

2.3.4 Merge

The final step is optional and relevant only when the records need to be merged or consolidated. The graph structure is no longer required for such scenarios, as only merged individual entries are necessary in the final data warehouse. As our work required the output to be delivered in this fashion, we merged the cleaned graphs and simplified the duplicate and redundant labels.

2.3.5 Warehouse content evaluation and availability

The suitability of the data warehouse content for scientific tasks was evaluated against a list of largest selling pharmaceutical products Q4 2013, obtained from Wikipedia the 10th of December 2014 (Wikipedia, 2014b). Given a brand name on the list, the task was to check if an entry was present in the data warehouse, and if this entry was correct or not: only one drug product should be described in the entry, and all the information about the product should be located on this entry and not spread across multiple records. We chose this list of best selling products, as corresponding entries in the warehouse are more likely to contain ambiguities and errors due to an often larger synonym set for the given products. These drug products are also more likely to be searched and referred to by the users of the warehouse, given their importance on the market, and are therefore solid evaluation points. The Wikipedia list contains a few errors and imprecisions on its own, highlighted on the Supplementary Tables 1 and 2 alongside the results. More evaluation work was performed internally, in particular a successful demonstration of superiority against existing solution (not presented). A subset of the final database containing the public sources is openly available at https://github.com/loopasam/flexible-data-integration. The

repository also contains the excluded records by the methodology, and a discussion based on some illustrative cases.

3 Results

The goal of the methodology is to integrate large amounts of heterogeneous data coming from different sources while detecting and handling errors to optimize the required curation. Equivalent records from different databases can indeed be linked based on the common labels they share (e.g. a brand name), as presented in the methodology section (*Link* step); the approach is very straightforward and assumes that entities with the same name are identical. However this strategy fails when a label is misassigned in an original record or when a name is ambiguously used to describe different entities. As a result, links between records are incorrectly created and propagated which creates hairballs, or large sets of records erroneously grouped together. For such cases, the source entries containing the misleading information, at the origin of the hairball, must be excluded. We present in this section the results and evaluation of the methodology, and characterize with examples the hairballs and their resolution towards better quality scientific content. We started with eight different sources describing 373 823 drug records and containing a total of 1 814 281 unique label entries. After the *Link* step, 182 389 graphs were created. The final database of integrated records contains 252 488 entries (i.e. the drug products) and 1 814 281 labels. In the process, 9 651 problematic records were identified and excluded from the build (2.58% of total starting number of records).

3.1 Hairballs

The methodology presented emphasizes the identification of problematic records, as they are the ones subject to manual curation, an expensive and tedious process. Data hairballs are an expected phenomenon using biomedical data (Williams *et al.*, 2012b), and vary in sizes and shapes. They are intuitively identifiable as graphs of low densities, when compared against the ideal case of a complete graph (see Section 2). In our work, we determined using an optimal density threshold of 0.59 to identify erroneous graphs using a ROC analysis (sensitivity = 0.828, specificity = 0.846). In line with this observation, the lower the density, the more likely the graph is to be problematic: We did not find for instance any non-problematic graph with a density below 0.33 (sensitivity = 0.234, specificity = 1.0). Figure 4 presents one of these hairballs, containing no less than 503 records, themselves describing dozens of different drug products. If kept in the integrated data, hairballs create further ambiguities and greatly diminish the value of the data integration exercise. For instance, it would not be possible to separately identify the different drug products contained in the hairball, with potentially important consequences if the integrated data were used for pharmacovigilance purposes. It should be noted that the older a drug is, the more likely it is to be part of a hairball, as more synonyms are used to describe it.

Fortunately, hairballs are not the most prevalent structures found in the integrated dataset. Figure 5 presents the distribution of size and density for the graphs formed during the equivalence linking; the large majority of graphs are complete or of high density (>0.95) and of small size, as expected (i.e. below the number of starting sources). The same figure also reveals the series of graphs with lower densities, namely the hairballs to be corrected. Given the large number of original records and labels, it would have been very challenging to manually identify these entries.

3.2 Error resolution

In order to increase the quality of the integrated data, hairballs have to be identified and the records causing them removed. We used the graph density to detect hairballs and the betweenness centrality to identify and exclude hairball-causing records (see Section 2). Figure 6 illustrates this process for the atorvastatin molecule (also commonly referred to/sold as Lipitor). Following a succession of cleaning steps, the records related to the molecule of interest are progressively isolated from the other drugs, until they form a stable graph with a high

Fig. 4. Illustration of data 'hairball'. The graph contains numerous records ($n = 503$) and has a low density ($d = 0.027$). The hairball contains many different unrelated entries, which would be erroneously merged together if ambiguous records were not identified and excluded. Data hairballs compromise the value and scientific quality of the data warehouse

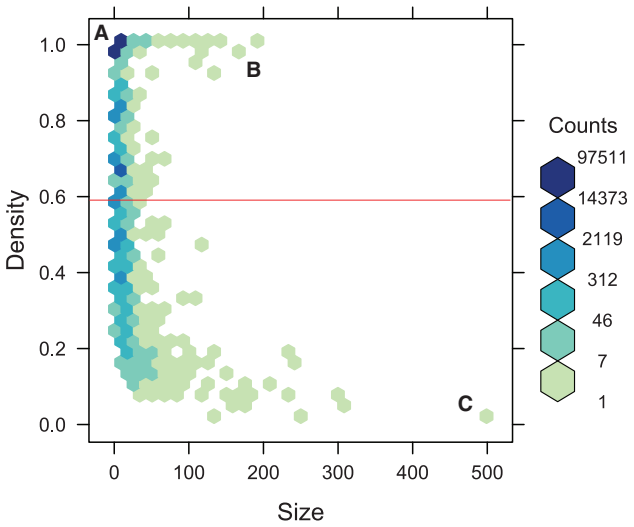

Fig. 5. Density and size of the 182 389 graphs formed in the *Link* step, before cleaning. The red line represents the density threshold under which graphs are inspected for cleaning ($d = 0.59$). (**A**) Optimal data integration, graphs have a very high density and an expected size, based on the number of starting sources. (**B**) Dense graphs, with a large size, appearing due to a large number of similar entries in the source databases, for instance related to 'gene therapy'. (**C**) Graphs of low density and large size, the hairballs, as shown on Figure 4

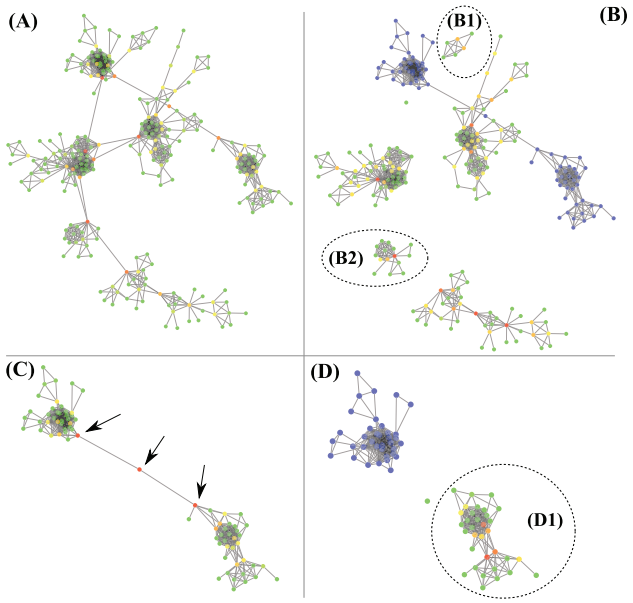

Fig. 6. Cleaning and resolution of a data hairball for the atorvastatin (Lipitor) entry. (**A**) Initial hairball of erroneously linked records, describing 12 different drug products. (**B**) Exclusion of the records with high BC after the first cleaning step, the Lipitor-related entries are still mixed with other products (blue graph, low density). Some graphs are now dense and do not need further cleaning, for instance (B1) refers to gemfibrozil and (B2) to adenosine. (**C**) The hairball is subject to a second iteration step, where more records are excluded (arrows on the graph). (**D**) The entries related to the atorvastatin are eventually well isolated (blue graph) and ready to be merged as a high quality consolidated record. The other drug product (D1) is fenofibrate. See Supplementary Material for bigger version of this figure

density. Records causing the hairballs are excluded during the process: it therefore results in a loss of information, but an improvement of the resolution and data quality. Drug product categories are well separated and isolated from one another and it becomes possible to uniquely and unambiguously refer to the atorvastatin molecule after the cleaning steps. Furthermore, given that the Lipitor is one of the best selling drugs of all time (Kidd, 2006), we expect that it will be searched and referred to by many users, stressing the importance of providing a quality entry for it in the integrated dataset.

3.3 Evaluation

The validity of the graph-based methodology presented in this document was evaluated with a ROC analysis, performed to determine the optimal density and betweenness centrality thresholds (see Section 2). For the optimal density threshold ($d = 0.59$), the method obtained a sensitivity of 0.828 and specificity of 0.846. These high values provide confidence in the approach and in the suitability of the density as a measure to identify problematic groups of linked records. We obtained a specificity of 0.951 and a sensitivity of 0.879 for the optimal betweenness centrality threshold ($BC = 0.33$). These numbers reflect here as well the capability of the betweenness centrality to be a useful metric to separate correct records from problematic ones.

We decided to further evaluate the quality of the final resource after the integration and the multiple cleaning steps by looking at the list of the hundred best-selling drugs in 2013. Given a drug brand name as found in the list, we looked at whether the drug was present in the integrated data or not, and whether the entry was correct (see Section 2 for details). We were able to find entries for all of these drugs, with an overall correctness of 94%. These numbers show the suitability of the approach for data integration and quality control, in order to produce a resource suitable to address real-world problems.

3.3.1 Error analysis

As a primary motivation for this work, we wanted to integrate data in order to generate a resource describing drug products and their various synonyms in an unambiguous way: one entry should eventually correspond to one drug product only. However, drugs are complex entities to represent and define; multiple abstractions or views can be used to categorize them, either based on the chemical structure or on the clinical usage for instance (Batchelor *et al.*, 2014). We adopted a flexible representation, reflected in the *Link* step: original entries are asserted as equivalent if they share either a chemical name or structure (e.g. InChI), but also a brand name or a laboratory code among other properties. As a result the final categories can contain multiple chemical entities, representing for instance the active ingredient and the corresponding salts, prodrug forms or isotopic variants. It should be noted that we did not consider such cases as problematic during the evaluation, they were desired features. Graphs of integrated records can also have a low density and still be correct; this is often the case with drugs sold in a variety of different forms and formulations, such as glucagon or insulin for instance. For such cases, the important number of entries in source databases referring to the drug creates an unexpectedly large graph with a low density value, appearing as a false positive to be cleaned.

A common type of excluded records is the ones referring to projects or compound series, themselves describing multiple molecules. Such records are prone to generate hairballs, by connecting the entries describing the single molecules only. For such cases, the assigned labels were not necessarily wrong, but ambiguous and undesirable for the wanted representation. We also identified errors in the source records, for instance the same laboratory codes being incorrectly used to refer to different molecules. Whenever possible, we feed this information back to the original provider for inspection and correction; then the improved version of the record can be reconsidered in a future release of the resource. Another source of error comes from some drug combinations and mixtures. Depending on the representation in the original record, combinations can be linked to entries referring to an active ingredient alone. For instance the integrated entry about the paracetamol molecule erroneously contains an original record related to the drug combination paracetamol/methionine. This record was linked to the other entries because both the chemical structure of methionine and paracetamol molecules appear independently in the source entry describing the drug combination. In conclusion, some errors still remain unrevealed by the methodology (false negatives), but based on the evaluation, the overall results confirmed the suitability of the approach to prioritize and automatically resolve ambiguous entries, in particular in the absence of cross-references or when unstructured text has to be linked to connect records. Moreover, the method helps to evaluate and control the quality of the resulting data warehouse.

4 Discussion

We present in this manuscript a generic methodology to integrate and merge data coming from different repositories into a central warehouse containing the consolidated information. As opposite to previous approaches (see Section 1), the method emphasizes error detection and is qualified as flexible, for multiple reasons: First, the

assertion of equivalence between records can be implemented as a series of steps including for instance, data cleaning and transformation, in order to best match the desired outcome (e.g. chemical structures, cross-references, free-text label). Secondly, the approach can handle *n* numbers of data sources, and is not limited to handle only pairs of records. Thirdly, the density measure helps to extract numbers regarding how confident one can be in regards to the quality of the integrated information and to determine how much curation is required on the top of the data. Finally, the betweenness centrality identifies the erroneous records, and can be used to resolve issues either automatically or manually. Thresholds for graph measures can be flexibly adjusted too, in order to optimise the work based on the time and resources available.

The abstraction of the data integration problem as a graph representation has been already introduced by previous work (Berners-Lee *et al.*, 2001; Bollacker *et al.*, 2008; Singhal, 2012), yet the use of graph measures on the top of the data to assess the quality of the integration and to detect anomalies is novel as far as our knowledge goes. From a usability perspective, we believe that the quality of the integrated data is more important than the technology or standards used; in this regard, the methodology can be implemented in a variety of frameworks, from RDF graphs to traditional relational databases, as needed by the user. Other graph indicators could be used in the future for similar tasks and to quantify different problem types coming from data integration: As an example, we decided not to assign weights to the edges of the graphs. This choice was motivated by the sparsity of the data; we estimated that records sharing one label were just as likely to be equivalent to records sharing multiple labels. Considering weights could result in the computation of different graph metrics, useful to characterize a particular type of records, assuming a rationale could be defined for the alternative indicators the same way it was done here with the density and BC.

A particularly challenging task resulting from data integration initiatives is the maintenance of the created resource, the update of records and the incorporation of new information. This can be implemented alongside the methodology proposed, in different ways: First it is possible to create incremental versions of the warehouse. With this approach, the new data sources are downloaded following a certain time period, and the integrated dataset is rebuilt from scratch each time, following the same graph-based methodology. New and corrected records are tested again, and excluded if necessary. The main drawback of this option comes from the difficulty of maintaining unique identifiers, as equivalences can vary from one release to the other. The second possibility for data update considers the current data warehouse just as any standard starting source, and tries to match and disambiguate new records following the same methodology as presented in this article. Identifiers are easier to maintain as the new version derives from the old one, but it might become challenging to keep track of the origin of the data. Removed information from starting sources, such as synonyms, can also be trickier to detect and correct.

In conclusion, the goal of data integration is to provide a complete picture of a scientific problem; we introduced here a methodology emphasizing data coherence and correctness to fulfill this greater task, where ambiguities, redundancies and errors are identified and removed. In summary, the methodology presented addresses practical concerns faced by large scale data integration exercises, prevalent nowadays in the drug discovery domain. In an ideal world, database entries and cross-references would be perfectly maintained and errors non-existent, which would enable the straightforward creation of a semantic network between entities. In practice, a costly and tedious curation step is often necessary in order to control the quality of the resource and the scientific value of its content. The presented methodology focuses on this aspect, in order to best prepare the integrated data for the derivation of scientific knowledge.

Funding

F. Hoffmann-La Roche AG.

Conflict of Interest: none declared.

References

Batchelor,C. *et al.* (2014) Scientific lenses to support multiple views over linked chemistry data. In: *The Semantic Web–ISWC 2014*. Springer. pp. 98–113.

Berners-Lee,T. *et al.* (2001) The semantic web. *Scientific American*, **284**, 28–37.

Bollacker,K. *et al.* (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, pp. 1247–1250.

Brandes,U. (2001) A faster algorithm for betweenness centrality*. *Journal of Mathematical Sociology*, **25**, 163–177.

Dong,X. *et al.* (2014) Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 601–610.

Dutot,A. *et al.* (2007) Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. In *Emergent Properties in Natural and Artificial Complex Systems. Satellite Conference within the 4th European Conference on Complex Systems (ECCS'2007)*.

Fellegi,I.P. *et al.* (1969) A theory for record linkage. *J. Am. Stat. Assoc.*, **64**, 1183–1210.

Hernández,M.A. *et al.* (1998) Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining Knowled. Discov.*, **2**, 9–37.

Juty,N.*et al.* (2012) Identifiers. org and miriam registry: community resources to provide persistent identification. *Nucleic Acids Res.*, **40**, D580–D586.

Kidd,J. (2006) Life after statin patent expiries. *Nature Reviews Drug Discovery*, **5**, 813–814.

Lipinski,C.A. *et al.* (2014) Parallel worlds of public and commercial bioactive chemistry data. *J. Med. Chem.*

Marx,V. (2013) Biology: The big challenges of big data. *Nature*, **498**, 255–260.

Pence,H.E. and Williams,A. (2010) Chemspider: an online chemical information resource. *J. Chem. Educ.*, **87**, 1123–1124.

Roos,L. and Wajda,A. (1991) Record linkage strategies. part i: Estimating information and evaluating approaches. *Methods Inform. Med.*, **30**, 117–123.

Singhal,A. (2012) Introducing the knowledge graph: things, not strings. *Official Google Blog*.

Szalma,S. *et al.* (2010) Effective knowledge management in translational medicine. *J. Trans. Med.*, **8**, 68.

Tiikkainen,P. *et al.* (2013) Estimating error rates in bioactivity databases. *J. Chem. Inform. Model.*, **53**, 2499–2505.

Wajda,A. *et al.* (1991) Record linkage strategies: Part ii. portable software and deterministic matching. *Methods Inform. Med.*, **30**, 210–214.

Wikipedia (2014a) Complete graph.

Wikipedia (2014b) List of largest selling pharmaceutical products.

Williams,A.J. *et al.* (2012a) Open phacts: semantic interoperability for drug discovery. *Drug Discov. Today*, **17**, 1188–1198.

Williams,A.J. *et al.* (2012b) Towards a gold standard: regarding quality in public domain chemistry databases and approaches to improving the situation. *Drug Discov. Today*, **17**, 685–701.

Wilson,D.R. (2011) Beyond probabilistic record linkage: Using neural networks and complex features to improve genealogical record linkage. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 9–14. IEEE.

Winkler,W.E. (1995) Matching and record linkage. *Business Survey Methods*, **1**, 355–384.

Winkler,W.E. (2014) Matching and record linkage. *Wiley Interdisciplinary Reviews: Computational Statistics*, **6**, 313–325.