RESEARCH

Curami: An Assisted Attribute Curation Tool for the BioSamples Database.

Matthew Green*

Correspondence:
hewgreen@ebi.co.uk
European Bioinformatics Institute, European Molecular Biology Laboratory (EMBL), Wellcome Trust Genome Campus, CB10 1SD Cambridge, UK
Full list of author information is available at the end of the article
*Equal contributor

Abstract

First part title: Text for this section.

Second part title: Text for this section.

Keywords: metadata; biosamples; assisted curation; data quality

Background

Architects of biological databases are forced to decide which trade offs to make in order to best balance the data consumer's downstream requirements, a data contributors input effort versus the value added benefits and the project's available resources. Whilst inexplicit validation and pliable data schema lower the barrier to initial data submission, this practice results in unstructured data that is difficult to reuse without incurring high curation costs. Many tools have been built which aim to improve data quality whilst minimising incurred financial and time costs. These tools broadly operate either pre-data submission (validation) or post-data submission (curation) [REF NEEDED]. Within these two strategies, the tools are spread on a scale between fully automated or fully manual. Often increased manual effort correlates with increased data value and diminishing throughput. Therefore, a tool should be designed to suit the requirements of the resource [1]. Thankfully, there are applications that succesfully leverage the benefits of automated data assessment and combine this with expert manual curation to measurably increase the efficiency of manual curation efforts [2, 3, 4].

The BioSamples database at the European Bioinformatics Institute (EBI) is a metadata repository for all biological samples used to generate datasets across the EBI [5, 6]. The database consists of more than 4.5 million sample records which each contain a unique BioSamples identifier. Many records also have links to the sample's derived data in external databases, relationships to other samples (derived from, child of etc.) and sample metadata expressed as key value pairs (note that keys are hereinafter referred to as attributes). As the database provides a single point of entry to metadata from a wide range of scientific domains and technologies, it allows users to create fields that are not predefined to suit their needs. Whilst this increases utility and flexibility for submitters, the dataset incurs redundancy and inconsistency. For example there are 22 different attributes that contain longitudes. Although BioSamples mitigates this problem by providing elastic search with ontology expansion, the samples returned may not necessarily be findable, accessible, interoperable and reusable, the universal targets defined as the FAIR

principles [7]. A snapshot of the dataset containing 4,790,415 sample records (taken on the 5th January 2018) was used for initial analysis. These samples contained 29,751 unique attributes used conjointly 45,275,314 times, giving a mean average of 9.45 attributes per sample record. The redundancy and inconsistency within the large number of attributes in the BioSamples database prompted the creation of the assisted-curation tool described herein.

The overarching goal was to provide a tool that facilitates the reduction of redundancy and inconsistency of the attributes in the BioSamples database, ultimately to reduce the total number of attributes in the database without loosing information. In order to achieve this, we aimed to identify pairs of attributes that were semantically similar enough to merge and to further identify the correct polarity of the merging. Although these aims differ from recent work to identify more generally topically related attributes in the Gene Expression Omnibus (GEO) [8, 9], the strategy the authors employ highlighted some key challenges to solving this problem [10]. Whilst GEO encourages data submitters to conform to the Minimum Information About a Microarray Experiment (MIAME) guidelines [11], similarly to BioSamples, the lack of a controlled vocabulary leads to the usage of different terms to represent the same concept. The authors developed a clustering methodology to group attributes that covered similar concepts and in doing so aim to separate curation activities into more manageable chunks. Unfortunately, applying clustering methodologies to data with heavily biased distributions leads to biased results which impedes semantic understanding. Furthermore as is the case of BioSamples, the majority of metadata submissions come from a few pipelines which may impose their own validation, recommendations and guidelines which lead to further bias and render frequency of term usage a less definitive predictive metric.

Methods

Source code and application documentation is available for inspection and reuse at *https://github.com/EBIBioSamples/curami*.

Overview of Data and Work Flow

Curami's modular design is to aid reuse and adaption of the code (see figure 1). Prior to running the application, initialization is required to identify attribute pairs and calculate the various features described in table [ADD TABLE OF FEATURES]. This process is independently initiated and can be ran centrally and updated on a regular basis. In the first step of initialization, each sample record is requested via the BioSamples Application Programming Interface (API) and parsed into four input files. A CSV list of unique attributes and their frequency of usage, a CSV list of sample records (including the BioSamples ID and the attributes the sample contains), a CSV list of attribute pairs that co-occur within a sample record at least once with the frequency of the co-occurrence and a JSON file with unique values and their frequency of use for each unique attribute. The second step of initialisation uses these four files to create a network using the Python package networkX [12] which is exported as a .gexf for later use by the co-occurrences analysis script. This graph contains nodes representing individual attributes and weighted edges with cco-occurrence can also be instantly visualised using Gephi [13] and explored

using various inbuilt force-directed layout algorithms [14] (see Figure [**?**]). 29,751 attributes represent 885,092,250 potential pairs, excluding self-matches. Initially, to reduce this number for feature analysis pairs with a Levenshtein distance is lower than 0.8 are removed [15]. 35,699 pairs were above this threshold (lexically matched pairs) and underwent further analysis.

There are currently three modules that extract features from the lexically matched pairs. These aim to employ orthogonal measures of similarity and their separation encourages future development of new feature extraction methods (such as ontology analysis). In addition to the lexical pairwise filtering, the lexical analysis module also categorises the type of lexical miss match

Co-occurrence Network

Co-occurrence refers to attributes that are used together within the same sample. By counting co-occurrence we can quantify a relationship between attributes and calculate a relationship weighting. A high frequency of co-occurrence may indicate that the attributes provide distinct information and therefore should not be merged. However due to the skewed distribution of attribute usage, highly popular attributes will co-occur more frequently than less popular attributes. To account for frequency of use when calculating co-occurrence weighting the following normalisation is performed.

If $C(a_1)$ is the number of occurrences of the first attribute, $C(a_2)$ the number of occurrences of the second and $C(a_t)$ the total occurrences of all attributes in the dataset. The probability of $C(a_1)$ occurrence ($P_{(a1)}$) is:

$$P_{(a1)} = \frac{C(a_1)}{C(a_t)}$$

Therefore, we can calculate the expected co-occurrence ($E[x]$) if attributes were randomly paired:

$$E[x] = P_{(a1)} \cdot P_{(a2)} \cdot P_{(t)}$$

The observed co-occurrence ($O[x]$) for each pair is transformed into a normalised weight ($W_{(a1,a2)}$) like so:

$$W_{(a1,a2)} = \frac{O[x] - E[x]}{\sum(O[x] - E[x])}$$

These weights are used to create a NetworkX co-occurrence graph. If the frequency of co-occurrence is zero the weight will also be zero, although these edges are not explicitly calculated to reduce overhead. Equally the co-occurrence JSON only contains counts greater than zero. The theoretical weight range is +1 to -1 and all weights in the graph sum to 1. As the majority of BioSamples metadata records come from relatively few sources, downstream validation biases these weightings. For example, table 3 a. shows the 5 strongest associated attributes which are all required in European Nucleotide Archive (ENA) checklists [REF]. Equally bias, table 3 b. shows the weakest attribute associations [INSERT DESCRIPTION OF THESE HERE AFTER SPEAKING TO ENA]. Nevertheless, weights are often intuitively relevant. For example, table 3 c. highlights three pairs that intuitively belong together (depth & elevation, serovar & strain and disease state & individual) and two pairs that occupy contrary domains which intuitively don't overlap (organism part & serovar and organism part & environment biome). Individual attribute nodes are connected by the weights described above in a Neo4J subgraph (not connected to

the graph layout shown in figure 1) which powers the recommendation functionality in Curami.

Attribute Recommendation

Due to the diverse nature of the BioSamples data set, slicing attributes by specialism is a very important curator requirement. Rather than predefining these slices the application allows the user to select attributes that are commonly associated with the dataset they are interested in, and uses this to suggest pairs that are relevant by looking at attributes that are closest according to the co-occurrence weighting. For example, a curator interested in marine metagenome samples could choose attributes [GIVE SOME SUGGESTIONS HERE THAT FIT WITH THE FIGURE]

Calculated Data Features

Different setting options

Actual Curation Work (including some observations of problems and potential issues)

Pair Sorting Settings

Results

Discussion

Curami is designed to be used from two perspectives. As an assisted curation application it can be used to navigate the user through their domains of interest within the dataset and provide auxiliary information to the curator which may increase their confidence and allow them to make a curation decision. From a second perspective, this decision capture is incredibly useful as a training set to later discover in which circumstances the calculated features become predictive indicators of a required curation operation. Although this work has not explored analysis of the captured features, this second utility was a major contributing factor to the design of Curami. Displaying the right pairs to the curator and the information required to make decisions will aid the curation process [REF NEEDED] but the potential of later leveraging each of these decisions through machine learning will further increase a curators impact. However, the current lack of training data dictated that as a first step, Curami had to provide immediate utility whilst generating training data for later correlation analysis.

Conclusion

Competing interests
The authors declare that they have no competing interests.

Author's contributions
Text for this section ...

Acknowledgements
Text for this section ...

References

1. Goble, C., Stevens, R., Hull, D., Wolstencroft, K., Lopez, R.: Data curation+ process curation= data integration+ science. Briefings in bioinformatics **9**(6), 506–517 (2008)
2. Salgado, D., Krallinger, M., Depaule, M., Drula, E., Tendulkar, A.V., Leitner, F., Valencia, A., Marcelle, C.: Myminer: a web application for computer-assisted biocuration and text annotation. Bioinformatics **28**(17), 2285–2287 (2012)
3. Salimi, N., Vita, R.: The biocurator: connecting and enhancing scientific data. PLoS computational biology **2**(10), 125 (2006)

4. Szostak, J., Ansari, S., Madan, S., Fluck, J., Talikka, M., Iskandar, A., De Leon, H., Hofmann-Apitius, M., Peitsch, M.C., Hoeng, J.: Construction of biological networks from unstructured information based on a semi-automated curation workflow. Database **2015** (2015)
5. Gostev, M., Faulconbridge, A., Brandizi, M., Fernandez-Banet, J., Sarkans, U., Brazma, A., Parkinson, H.: The biosample database (biosd) at the european bioinformatics institute. Nucleic acids research **40**(D1), 64–70 (2011)
6. Faulconbridge, A., Burdett, T., Brandizi, M., Gostev, M., Pereira, R., Vasant, D., Sarkans, U., Brazma, A., Parkinson, H.: Updates to biosamples database at european bioinformatics institute. Nucleic acids research **42**(D1), 50–52 (2013)
7. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. Scientific data **3** (2016)
8. Edgar, R., Domrachev, M., Lash, A.E.: Gene expression omnibus: Ncbi gene expression and hybridization array data repository. Nucleic acids research **30**(1), 207–210 (2002)
9. Barrett, T., Wilhite, S.E., Ledoux, P., Evangelista, C., Kim, I.F., Tomashevsky, M., Marshall, K.A., Phillippy, K.H., Sherman, P.M., Holko, M., *et al.*: Ncbi geo: archive for functional genomics data setsóupdate. Nucleic acids research **41**(D1), 991–995 (2012)
10. Hu, W., Zaveri, A., Qiu, H., Dumontier, M.: Cleaning by clustering: methodology for addressing data quality issues in biomedical metadata. BMC bioinformatics **18**(1), 415 (2017)
11. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., *et al.*: Minimum information about a microarray experiment (miame)ótoward standards for microarray data. Nature genetics **29**(4), 365 (2001)
12. Hagberg, A., Schult, D., Swart, P.: NetworkX. https://github.com/networkx/networkx Accessed 13.4.18
13. Bastian, M., Heymann, S., Jacomy, M., *et al.*: Gephi: an open source software for exploring and manipulating networks. Icwsm **8**, 361–362 (2009)
14. Jacomy, M., Venturini, T., Heymann, S., Bastian, M.: Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. PloS one **9**(6), 98679 (2014)
15. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, pp. 707–710 (1966)
16. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and experience **21**(11), 1129–1164 (1991)
17. Newman, M.E.: Modularity and community structure in networks. Proceedings of the national academy of sciences **103**(23), 8577–8582 (2006)
18. Jaccard, P.: Distribution of the alpine flora in the dranseís basin and some neighbouring regions. Bulletin de la Societe vaudoise des Sciences Naturelles **37**, 241–272 (1901)

Figures

Figure 1 Data and Work Flow Overview of the modular design of Curami showing separation of modules and general dataflow. Initialization and analysis modules 1-5 should be executed in series to pull information from the BioSamples API to generate input files, pair lexically similar attributes, extract pairwise and individual features and store the results in the Neo4J database. The Data Analysis Store has three node types User, Pair and Attribute with corresponding relationships as shown. Edges *r1* represent curation decisions made by the user about a pair of attributes and can be one of four types; merge, merge with reversed polarity, don't merge and skip. Edged *r2* has only one type, 'pair contains', and indicated which individual attributes are in the pair. The Curation Interface allows curators to view the attribute pairs sorted by applying one of the available settings described in section . Curation decisions are then stored in the Neo4J database which can then be directly converted into curation objects, a format that can be directly submitted to the BioSamples API.

Figure 2 Gephi snapshot Snapshot from the interactive network layout tool Gephi [13]. Each attribute is a node and the node size represents the approximate frequency of attribute usage (size is binned rather than scaled). The Fruchterman-Reingold layout [16] uses cooccurance weighting captured on the edges which are represented as grey lines between nodes. Colour assignment was done using the inbuilt Newman's modularity function [17] to loosely highlight node clusters. Some specific attributes are labelled along with some large clusters that show many attributes related to 'human metagenome' samples and a commonly used 'medical screen'.

Tables

Table 1 A table of analysis features calculated for each attribute pair.

Feature	Description
Levenshtein Distance	[15]. N.B. pairs that score ¡ 0.8 are not included.
Attribute 1 frequency	The frequency of usage of attribute 1. This is used to predict the polarity of the merge. This assumes that the most popular attribute is most likely to be correct.
Attribute 2 frequency	The frequency of usage of attribute 2.
Discrepancy Type	The result of several lexical tests (see table 2)
Camel Case	Boolean indicating lexical detection of potential camel case usage.
Edge Weight	placeholder
Jaccard Coefficient [18]	Calculated using the co-occurrence NetworkX graph. Quantifies the overlap of usage between attributes which may not directly co-occur (have a weighted edge in the graph).
Break Number	placeholder
Degree	placeholder
Edge Total	placeholder
Value Match Type	Can be either numeric, date or string match if more than 90% of the values adhere to one data type. Alternatively, the field will show a mixed match or no match.
Order of Magnitude	placeholder
Jaro Score	placeholder

Table 2 These tests aim to catagorize the lexical difference between the lexically similar attributes. ‡ are strong candidates for automated curation.

Lexical Catagory	Description of Difference Test
Number Discrepancy	A numerical character differentiates the attributes.
Case Discrepancy	If case stripping improves the Levenshtein score, case is identified as at least one differential factor.
Space Discrepancy	Indicates if a space is at least one part of the discrepancy.
Only Space Discrepancy‡	If a space difference is the only difference between the attributes.
Specials Discrepancy	If special characters are partially responsible for the difference.
Just Specials Discrepancy‡	If special characters are the only difference between the attributes.
Word Number Discrepancy	If the number of tokens in the attributes differ.
Stop Word Discrepancy	If the difference is due to the presence of stopwords (from the NLTK corpus imported from nltk.corpus import stopwords).
Dictionary Matching	Triggered if there is a difference in spelling between the attribute's tokens (so ignoring equally misspelled tokens) that fail a dictionary test against enchants US dictionary after stripping numbers, case and special characters.
Lemmatisation Matching	If lemmatisation or stemming produces an identical match.
S discrepancy‡	If an additional 's' character is the only difference between attributes.

Table 3 a. Most strongly associated and b. weakly associated pairs ranked by co-occurrence weighting. c. Highlights pairs that demonstrate intuitive validation of the weighted ranking.

a. Strongest attribute associations

Rank	Attribute Name	Second Attribute Name	$W_{(a1,a2)}$
1	package	synonym	6.94×10^{-4}
2	model	synonym	6.94×10^{-4}
3	colection date	geographic location	4.45×10^{-4}
4	geographic location	package	4.43×10^{-4}
5	geographic location	model	4.43×10^{-4}

b. Weakest attribute associations

Rank	Attribute Name	Second Attribute Name	$W_{(a1,a2)}$
710293	Sample source name	synonym	-2.54×10^{-4}
710294	Sample_source_name	model	-2.87×10^{-4}
710295	description	model	-4.06×10^{-4}
710296	description	package	-4.07×10^{-4}
710297	Sample_source_name	synonym	-4.17×10^{-4}

c. Highlighted pairs

Rank	Attribute Name	Second Attribute Name	$W_{(a1,a2)}$
107	depth	elevation	8.72×10^{-5}
251	serovar	strain	5.90×10^{-5}
363	disease state	individual	4.84×10^{-5}
710117	organism part	serovar	-2.38×10^{-5}
710223	environment biome	organism part	-5.24×10^{-5}