RESEARCH

Stitcher: An entity resolution framework for comprehensive data integration of approved drugs

Dac-Trung Nguyen*, Noel Southall, Ivan Grishagin, Daniel Katzel, Tyler Peryea†, Ajit Jadhav and Ewy Mathé

*Correspondence:
nguyenda@mail.nih.gov
Division of Pre-clinical Innovation, National Center for Advancing Translational Sciences (NCATS), National Institutes of Health, USA
Full list of author information is available at the end of the article
†Current address: Office of Health Informatics, Office of Chief Scientist, Food and Drug Administration (FDA), USA

Abstract

As biomedical data continues to grow at an unprecedented rate, the need to provide an integrated biomedical knowledgebase for drug discovery remains a major challenge. One of the limiting factors in any data integration effort is *entity resolution* (ER), the ability to determine entities from different data sources with shared partial (and perhaps even inconsistent) identities are equivalent. For many entity types with well-defined nomenclature (e.g., gene, protein, cell line, etc.), ER amounts to a simple lookup of identifiers. For drug entity type, however, ER is rather challenging due to ambiguities in how it is defined and represented. Herein we report on our recent effort to develop an ER framework, STITCHER, for drug data integration. Using *active moiety* as the defining concept for drug entity, we develop a set of equivalence relations (i.e., *stitch keys*) in STITCHER specifically for ER of drug entities. We demonstrate the utility of our approach through the development of a comprehensive resource of drugs that have either been marketed or approved in the United States for human use.

Keywords: entity resolution; drug database; data integration

Introduction

As the volume of biological data continues to grow at an unprecedented rate, data de-duplication—also commonly known as record linkage or *entity resolution*—is proportionally playing a prominent role in data integration. From the construction of training data for machine learning to building knowledge graphs as epistemological frameworks for artificial intelligence, proper entity resolution (ER) is essential in creating ground-truth data and turning data into knowledge. The core challenge of ER is in establishing *equivalence* between entities. For well-defined entity types (e.g., gene, tissue, cell line), this is often determined solely based on established identifiers and nomenclature; for other entity types (e.g., drug, disease, phenotype), however, equivalence is not as well-established due to conceptual ambiguities in how entities are defined and represented. Take disease as an example; the discrepancy between the theoretical concept of "disease entity" from its clinical nosology [1] is what makes disease ER extremely challenging.

Drug is another entity type that is also challenging for ER due to ambiguities in its definitions and representations. While the word is included within the name of the organization, the U.S. Food and Drug Administration (FDA) does not have a straightforward definition of the word "drug." The Federal Food Drug and Cosmetic Act (FD&C Act) and FDA regulations define the term drug, in part, by reference to its intended use, as "articles intended for use in the diagnosis, cure, mitigation, treatment, or prevention of disease" and "articles (other than food) intended to affect the structure or any function of the body of man or other animals" [2]. More practically, the agency defines "drug substance" and "drug product," respectively, as the physical ingredients found in marketed products. Others use the word "drug" to sometimes refer to "drug substances" and sometimes to "drug products" as convenient, and this causes a great deal of semantic confusion within drug data found on the internet. The National Library of Medicine produces a semantic product, RxNorm, that provides a variety of precise semantic types for ingredients, tradenames, dose forms, semantic clinical drug components, semantic clinical drug forms, and semantic clinical drugs which facilitate working with drug data, but its terminology is unfortunately limited to commonly used prescription drugs, "clinically significant ingredients," and adoption of this complex semantic scheme is limited [3].

There is a third definition of the word "drug" that is commonly used in the literature and used by the FDA when it refers to an active moiety and a new molecular entity. In this case, ingredients whose pharmacological effect occurs through the same molecular entity are considered the same drug. This holds for different salt forms such as SUMATRIPTAN SUCCINATE and SUMATRIPTAN HEMISULFATE, but it also holds for prodrugs and their metabolized active forms such as BRINCIDOFOVIR and CIDOFOVIR [4]. The FDA defines *active moiety* as follows:

> An active moiety is a molecule or ion, excluding those appended portions of the molecule that cause the drug to be an ester, salt (including a salt with hydrogen or coordination bonds), or other noncovalent derivative (such as a complex, chelate, or clathrate) of the molecule, responsible for the physiological or pharmacological action of the drug substance [5].

Under the Food and Drug Administration Amendments Act of 2007, all newly introduced active moieties must first be reviewed by an advisory committee before the FDA can approve these products. We adopt this defintion throughout the paper.

As in other information domains, the names used to refer to drug substances and products are particularly problematic because their definitions change as a function of location or jurisdiction, time and context. FDA and other national regulators of medicines have collaborated to produce ISO 11238 [6] which endeavors to define an information scheme for the unambiguous identification of all ingredients found in medicinal products, and FDA uses an implementation of ISO 11238 as the backbone of its information systems within the agency [7]. While this facilitates data exchange within the FDA and with other national authorities, the task still remains to be able to map other, external data sources into this rigorously-defined scheme using whatever names and data are at hand.

Herein we report on our recent data integration effort to build a comprehensive resource of drugs that have either been marketed or approved in the United States for human use. Such a resource is not only instrumental for drug repurposing but also serves as a valuable tool to further our understanding of the mechanistic properties of molecular targets [8, 9]. To the best of our knowledge, INXIGHT DRUGS is currently the most comprehensive resource of its kind. In the remainder of this paper, we discuss the development of STITCHER, an ER framework that we have developed to address the challenges of drug data integration.

Overview

As a motivating example, consider the data shown in Table 1, which illustrates a common scenario where we would like to integrate data from multiple sources such that each data source might contain only partial (and possibly conflicting) information regarding the identities of the entities. There are four possible attributes that can be used to assert equivalence for the entities in this table, with the attribute `Structure` indicates whether the entity is associated with a chemical structure and that it possibly contains errors (e.g., missing or incorrect stereo assignments). Based on the definition of "drug" as defined above, we would like to determine the number of unique drugs that are there in the table. Examining the table, we can immediately make the case for three equivalence classes $\{A_4, B_2\}$, $\{A_3, C_2\}$, and $\{A_2, C_3\}$. We also know that A_3 is an active moiety of A_4 per our definition; through transitive closure, we now have the following equivalence classes: $\{A_4, B_2, A_3, C_2\}$ and $\{A_2, C_3\}$. (We should note here this active moiety relationship is rather trivial in that it can be inferred algorithmically, whereas active moiety relationships that involve metal complexes and non-trivial metabolites are likely to require manual curations.) Continue to perform transitive closure on the shared attributes in order, we eventually arrive at the final three equivalence classes $\{A_1, A_2, B_1, C_3, D_1, D_3\}$, $\{C_1\}$, and $\{A_3, A_4, B_2, C_2, D_2, E_1\}$, which correspond to three distinct isomers (*S*-, *R*-, and racemic, respectively) of the drug OMEPRAZOLE. (Note that the *R*-isomer is not an approved drug.)

The goal of ER, as shown by the above example, is to partition entities within a given dataset into disjoint sets such that those within the same set are considered equivalent. To achive this, STITCHER first "stitches" together entities with shared identity attributes (i.e., *stitch keys*). Next, transitive closure is performed based on heuristics that we have developed in assigning priorities to the attributes. And finally equivalence classes are efficiently derived through standard union-find algorithm [10]. This is the essence of STITCHER.

Preliminary concepts

The conceptual data model underlying STITCHER is a *multigraph*. Within this multigraph, a node can either be a *stitch node* or *data node*. Each data node represents the "raw" entity as ingested from the data source; its corresponding stitch node is a *standardized* representation that is used for *stitching*. An edge between two stitch nodes can either be a *stitch key* (undirected) or *relationship* (directed). A unique *stitch value* is associated with each stitch key such that it forms a clique. Figure 1 shows an instance of a connected component of a stitch multigraph with overlapping cliques.

A connected component in the stitch multigraph represents the basic unit of work for ER. While the majority of connected components are of reasonable sizes (e.g., 20 to 50 stitch nodes), the real challenges center around effective strategies for handling very large connected components—or also commonly known as *hairballs* [11]. For example, the current version of the INXIGHT DRUGS resource has a hairball close to 30,000 stitch nodes spanning across 15 data sources. Developing strategies to untangle large hairballs is the primary challenge for STITCHER.

Equivalence classes in a connected component are explicitly represented as *sgroup nodes* in the stitch multigraph. Entities that share a common *sgroup* node are considered equivalent. There can be multiple instances of sgroup nodes for a given stitch node, with each instance perhaps reflects a specific algorithmic strategy or version. Figure 1 shows that there is only one equivalence class as determined by the underlying ER algorithm for the given connected component.

Stitch keys

Stitch key is a core concept in STITCHER. It defines how entities are matched, which, in turn, determines how cliques and connected components are formed. By virtue of its importance, the stitch key should reflect the true identity of the entity as much as possible. Depending on the entity type, the stitch key can be generic (e.g., synonym) or very specific (e.g., molecular hash key). For drug entity type, STITCHER relies on the following stitch keys for each entity:

`N_Name`. This is the most generic stitch key available. Stitch values associated with this stitch key can be any established names or nomenclature; e.g., tradenames, INN (International Nonproprietary Names), USAN (United States Adopted Names), IUPAC (International Union of Pure and Applied Chemistry).

`I_UNII`, `I_CAS`, `I_CID`, `I_CODE`. These stitch keys represent (i) unique identifiers assigned to the entity by a well-known registrar (e.g., the U.S. Food and Drug Administration in the case of UNII) or (ii) internal company code. `I_UNII`, `I_CAS`, and `I_CID` are specific to drug (or substance in general) entity type, whereas `I_CODE` can be used for any type of identifiers. The decision to use specific stitch keys over generic ones ultimately rests on the strategies used for ER.

`H_LyChI_L5`, `H_LyChI_L4`, `H_LyChI_L3`. For the small molecule class of drugs, perhaps more important than any identifiers is the underlying chemical structure definition. These stitch keys are hash values derived from the molecular structure at different resolutions [12]. Section discusses in detail how these derived stitch values are generated.

`R_activeMoiety`. Technically not a stitch key, the active moiety relationship between two drugs provides a strong evidence of equivalence. While this relationship can be inferred directly from the chemical structures (e.g., freebase and salt forms, with and without esters), there is some level of curation needed to handle structures with metal complexes and metabolites.

Table 2 shows examples of stitch keys and stitch values for the drug entity IMATINIB MESYLATE. In this example, the `R_activeMoiety` relationship specifies the UNII of the freebase form (IMATINIB) of IMATINIB MESYLATE.

Methods

In general, data integration with STITCHER consists of four basic steps executed in order: *ingestion*, *stitching*, *entity resolution*, and *entity normalization*. With the exception of *entity resolution*, all other steps—as they are currently implemented in STITCHER—are generic and can be applied to a wide range of entity types.

Data ingestion

Stitcher is capable of ingesting data in a wide variety of sources and formats. Semantic formats such as OWL, RDF, and Turtle are supported as are JSON, delimiter separated text, and custom formats. For non-semantic format, a separate configuration file is required to map properties to stitch keys.

An important step in data ingestion is the standardization and validation of stitch values. For `N_Name` stitch key, the standardization procedure is simply to convert the input string to uppercase; no validation is performed. For `I_UNII` and `I_CAS` stitch keys, no standardization is required, and validation is a simple checksum calculation to ensure the stitch value is proper. Depending on the input format, Stitcher also provides basic utilities (e.g., regular expression) to help with data transformation during ingestion.

Perhaps the most unique feature of Stitcher is its ability to incorporate knowledge of chemical structures into ER. Whereas traditional approaches rely on names and identifiers to determine equivalence substances, Stitcher goes a step further and utilizes the underlying chemical structures to infer equivalence. This is particularly relevant when the drug is a mixture, prodrug, or active moiety with complex excipient (or derivative thereof). As an example, consider the drug entity Imatinib Mesylate and its active ingredient Imatinib. Here, it is obvious that the two entities cannot be matched by name alone. Instead, having structural information by way of molecular hash keys for each molecular component allows us to determine equivalence from the common active moiety Imatinib between the two entities. This trivial example might suggest that, instead of comparing names exactly, we find the longest common substring of the names. The approach would certainly work in this example, but to make it work in general would require very specialized parsing rules and dictionaries.

For data sources with chemical structures, the most computationally demanding step in data ingestion is the generation of molecular hash keys. Hash keys are generated for each component of a chemical structure in three different structural levels: `L5`, `L4`, and `L3`, which correspond to stitch keys `H_LYCHI_L5`, `H_LYCHI_L4`, and `H_LYCHI_L3`, respectively. Level `L5` is the most specific; it represents the chemical structure as-is, i.e., without structure normalization and standardization. With the exception of the relation `R_activeMoiety`, a match at this level has higher priority over other stitch keys. The next level `L4` represents the structure after normalization and standardization per the LyChI software package [12]. A match at this level implies that two structures are equivalent in terms of stereochemistry, resonance, and tautomerism. And the last level `L3` is the same as `L4` but without stereochemistry. A match at this level is considered weak and does not constitute equivalence without other significant supporting evidence. The purpose for `L3` is in anticipation of incorrect or missing stereo information, which is one of the most common type of errors associated with chemical structures. For each hash key, a suffix `-M`, `-S`, or `-N` is also assigned to designate the molecular component as either a *metal*, *salt*, or *neither*, respectively. Table 2 illustrates all three representations for the drug Imatinib Mesylate. Note that the cardinality for `L5` is always one, whereas for `L4` and `L3` the cardinality is equal to the number of non-hydrate molecular components. (Hydrate components are removed prior to processing.)

Data stitching

Stitching is the process by which the stitch multigraph is incrementally constructed as data is ingested. Algorithm 1 describes the basic stitching algorithm of STITCHER. This algorithm is applied to each data source, and upon its completion produced a stitch multigraph such that any stitch value that spans N stitch nodes is an induced clique, i.e., a complete subgraph of N nodes and $\frac{N(N-1)}{2}$ edges. Overlapping induced cliques form the basis for the proposed ER approach discussed in the next section. As a side-effect, the stitching algorithm also utilizes the union-find algorithm [10] to efficiently track connected components.

Entity resolution

After all data sources have been stitched together, the next step is to partition the stitch multigraph into disjoint entity sets. Formally, this step is known as *entity resolution* (ER) and is the only step within STITCHER that is specific to the drug entity type. This is to be expected: Given that ER is about adjudicating the splitting and merging of entities, a reasonable amount of knowledge of the entity type is required for the adjudication to be effective. For a given connected component, the iterative process of assigning equivalence labels to stitch nodes is known as *untangling*. Algorithm 2 gives a high level outline of the untangling process.

At the core of the algorithm is the implied priorities associated with the stitch keys. The relation `R_activeMoiety` has the highest priority as it is manually generated and is a special relation that only available to the G-SRS data source. As an example, consider the entities *acetylsalicylic acid* (or also commonly known as *aspirin*) and *ethyl acetylsalicylate* shown in Figure 2. While the two entities have nothing in common, in G-SRS *acetylsalicylic acid* is annotated as being an *active moiety* of *ethyl acetylsalicylate*. Further examination of the structural differences shows that only an *ester* separates the two entities; this falls well within what the FDA considers as equivalent drugs. This example also highlights a quagmire: Computationally, there is nothing to prevent us from imputing *active moiety* relationships through efficient (sub-) graph isomorphisms. This, however, is a very tempted trap that we have thus far resisted due to other forms of *active moiety* relationships—e.g., metabolites and metals—that would require considerable investment of effort.

The next priority is the stitch key `I_UNII`. As UNII is the primary identifier for the G-SRS data source, any data source that provides mapping based on this identifier implies that the data source has sufficient knowledge of G-SRS (i.e., guilt by association). For entities that can be represented by chemical structures, the stitch key `H_LyChI_L4` has the next level of priority. The complexity required for two entities to have the same stitch values means that the entities are less likely to match by errors. The rest of the stitch keys (i.e., `N_Name`, `I_CAS`, `I_CID`) all have the lowest priority.

At the completion of Algorithm 2, the disjoint set data structure U contains all equivalence entity classes such that each class is represented by an *sgroup* node in the stitch multigraph. The sgroup nodes are the *resolved entities*.

Entity normalization

The last step in the data integration pipeline is to decide how the resolved entities are defined. This step is referred to as *entity normalization* and its goals are to have

(i) clear and consistent strategies for merging properties and (ii) conflict resolution (semantic as well as self consistency). While this step can be quite trivial if the properties are mutually exclusive across all data sources, to address this in a general setting will require considerable efforts in terms of understanding the data source and its metadata. Within the context of the current work, we resort to a simple strategy: When merging properties, we preferentially use those that come from G-SRS with a basic consistency constraint that a synonym is never associated with more than one resolved entity. It also helps that many of the data sources in Table 5 have mutually exclusive properties; e.g., the property "drug approval year" in the Drugs@FDA data source is not a property of G-SRS.

Results

With Stitcher serving as the data integration framework, we set out to build a comprehensive resource, InXight Drugs, on drugs that have either been marketed or approved for human use in the United States. Our starting point is the public G-SRS data source from the FDA [14]. This data source is well-curated and contains over 100K substances across six different classes: chemical, structurally diverse, protein, mixture, polymer, and nucleic acid. As a data source derived from the FDA's internal substance registry system [7], the G-SRS data source naturally forms the basis of our data integration effort. Using this data source as the "seed" from which other data sources can map onto has the following benefits:

- Since the G-SRS data source implements the ISO 11238 standard [6] for defining medicinal substances, it serves as an ideal starting point for what constitutes a "drug."
- The data is a public version of the internal substance registry within the FDA; as such, it is well-curated and up-to-date.

Furthermore, by establishing a reference data source for data integration, we have finer control over the following:

Data quality. A reference data source is typically selected such that it is of high quality. Here, we can also impose other data quality constraints (e.g., no synonyms can span multiple entities) to guide ER.

Data resolution. ER is particularly challenging when data integration involves ontologies. A reference data source can serve as the anchor ontology from which other ontologies can be mapped. As with data quality, we can also impose any additional semantic constraints; e.g., an equivalence class cannot have more than one active moiety.

Data curation. Generating ground-truth data is more manageable with a single data source than across multiple data sources. This is particularly important due to the iterative feedback between data curation and data integration.

While the G-SRS data provides rigous definitions for substances, it lacks other information such as approval status, year, and juridiction, indication, patent, publication, and other uses. The complete list of data sources currently used by Stitcher is shown in Table 5.

Availability

The InXight Drugs resource is available at `https://drugs.ncats.io`. Stitcher and the data integration pipeline developed for the InXight Drugs resource are available in source form at `https://github.com/ncats/stitcher`. The stitch multigraph built with data sources listed in Table 5 is available as a Neo4j database [13] at `https://stitcher.ncats.io/browser`. This database currently contains 192,413 stitch nodes and 11,948,470 edges (relationships and stitch keys). Tables 3 and 4 give a breakdown of the stitch keys and values, respectively, in the stitch multigraph. All figures and examples used throughout this paper have been generated directly from this database.

Case studies

ASPIRIN is a versatile drug that can be used alone or in combination with other drugs. Shown in Figure 3 is the induced subgraph of the much larger ASPIRIN connected component that forms the ASPIRIN entity. This example demonstrates STITCHER's ability to tease out only the relevant stitch nodes for which ASPIRIN is likely to be the active moiety for the underlying substance.

LEVOMETHADYL and its derivative LEVACETYLMETHADOL are often considered as two separate drugs. This is readily apparent in Figure 4, which shows that there are two distinct "clusters" in the stitch multigraph. If ER is based on graph metrics (e.g., betweenness centrality), it is likely that this connected component will yield two drugs instead of one. Here, the priority of the stitch key allows the two clusters to be merged to indicate that there is only one drug.

Figure 5 shows the connected component for BENOXAPROFEN, a nonsteroidal antiinflammatory drug approved in 1982. This drug is a racemic mixture. The density of this connected component is a reflection of the lack of specified stereocenter that caused many spurious stitch keys. STITCHER is able to disambiguate the connected component into three distinct entities that represent the mixture, R-, and S-isomer.

Discussion

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Acknowledgements

We thank our colleagues, Mark Williams and Tyler Beck, for their valuable proof-reading of early drafts of the manuscript. We also thank our colleague, Tongan Zhao, for his help in developing a prototype curation user interface for STITCHER. We are particularly grateful to Alexey Zakharov and Tim Sheils for their constant encouragement and support.

References

1. Hucklenbroich, P.: "Disease Entity" as the Key Theoretical Concept of Medicine. Journal of Medicine and Philosophy **39**, 609–633 (2014)
2. FDA: Human drugs. [https://www.fda.gov/industry/regulated-products/human-drugs; accessed January, 2020] (2020). `https://www.fda.gov/industry/regulated-products/human-drugs`
3. NLM: RxNorm: Prescription for Electronic Drug Information Exchange. [https://www.nlm.nih.gov/research/umls/rxnorm/RxNorm.pdf; accessed January, 2020] (2005). `https://www.nlm.nih.gov/research/umls/rxnorm/RxNorm.pdf`
4. FDA: New drugs at FDA. [https://www.fda.gov/drugs/development-approval-process-drugs; accessed January, 2020] (2020). `https://www.fda.gov/drugs/development-approval-process-drugs`
5. FDA: DRUGS FOR HUMAN USE. [https://www.govinfo.gov/content/pkg/CFR-2012-title21-vol5/pdf/CFR-2012-title21-vol5-chapI-subchapD.pdf; accessed January, 2020] (2012). `https://www.govinfo.gov/content/pkg/CFR-2012-title21-vol5/pdf/CFR-2012-title21-vol5-chapI-subchapD.pdf`
6. ISO: Health informatics—Identification of medicinal products—Data elements and structures for the unique identification and exchange of regulated information on substances. [https://www.iso.org/standard/69697.html; accessed January, 2020] (2018). `https://www.iso.org/standard/69697.html`
7. FDA: FDA's Global Substance Registration System. [https://www.fda.gov/industry/fda-resources-data-standards/fdas-global-substance-registration-system; accessed January, 2020] (2019). `https://www.fda.gov/industry/fda-resources-data-standards/fdas-global-substance-registration-system`

8. Huang, R., Southall, N., Wang, Y., Yasgar, A., Shinn, P., Jadhav, A., Nguyen, D.-T., Austin, C.P.: The NCGC Pharmaceutical Collection: A Comprehensive Resource of Clinically Approved Drugs Enabling Repurposing and Chemical Genomics. Science Translational Medicine **3**, 80–16 (2011)
9. Huang, R., Zhu, H., Shinn, P., Ngan, D., Ye, L., Thakur, A., Grewal, G., Zhao, T., Southall, N., Hall, M.D., Simeonov, A., Austin, C.P.: The NCATS Pharmaceutical Collection: a 10-year update. Drug Discovery Today **24**, 2341–2349 (2019). doi:10.1016/j.drudis.2019.09.019
10. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, USA, ??? (2001)
11. Croset, S., Rupp, J., Romacker, M.: Flexible data integration and curation using a graph-based approach. BMC Bioinformatics **32**, 918–925 (2016). doi:10.1093/bioinformatics/btv644
12. NCATS: Layered Chemical Identifier. [https://github.com/ncats/lychi; accessed October, 2019] (2013). `https://github.com/ncats/lychi`
13. Neo4j: Neo4j database. version 3.5.13 (2020). `https://neo4j.com`
14. FDA: G-SRS substance database. [https://tripod.nih.gov/ginas; accessed January, 2020] (2019). `https://tripod.nih.gov/ginas`

Figures

Figure 1 A stitch multigraph. A connected component in the stitch multigraph with four *stitch nodes* (medium) and corresponding *data nodes* (small). Each stitch value forms a clique within this connected component. The edge labels between stitch nodes are the stitch keys. The large node is the derived entity (i.e., sgroup node) from entity resolution that establishes equivalence between the stitch nodes.

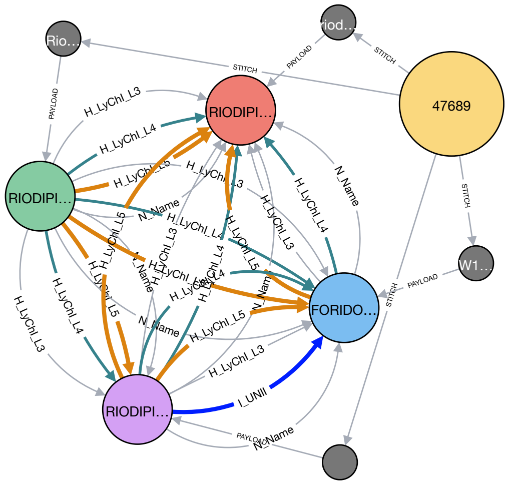
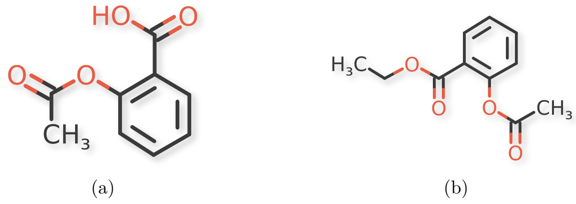


Figure 2 Chemical structures for (a) acetylsalicylic acid and (b) ethyl acetylsalicylate.



Tables

Algorithms

Figure 3 A connected component for ASPIRIN.

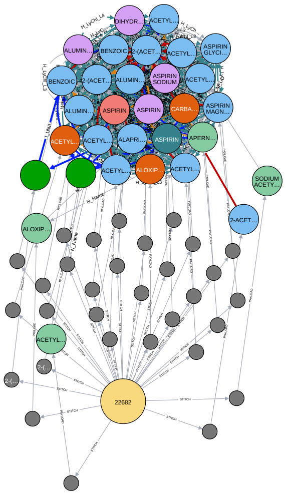


Figure 4 A connected component for LEVOMETHADYL that clearly shows two distinct clusters.

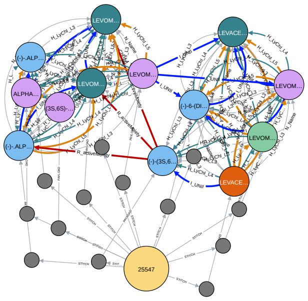


Figure 5 A dense connected component for BENOXAPROFEN that resolved to three unique entities.

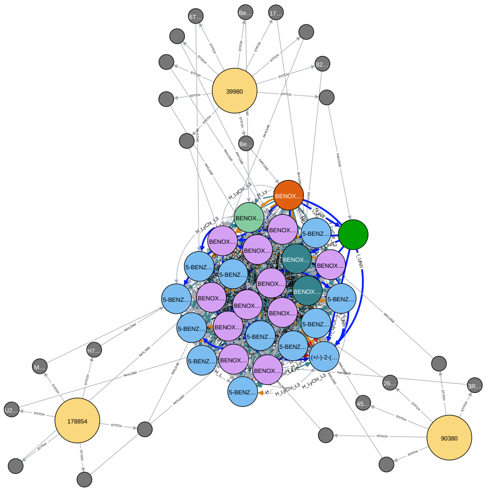


Table 1 An example of integrating data from multiple sources where each source contains only partial information.

Source	ID	Name	CAS	UNII	Structure
A	1	ESOMEPRAZOLE STRONTIUM ANHYDROUS	914613-86-8	SCC2RK476A	Correct
A	2	ESOMEPRAZOLE	217087-09-7	N3PA6559FT	Correct
A	3	OMEPRAZOLE	95382-33-5	KG60484QX9	Correct
A	4	OMEPRAZOLE SODIUM	95510-70-6	KV03YZ6QLW	Correct
B	1	Esomeprazole		C5N25H3803	Correct
B	2	Omeprazole		KV03YZ6QLW	Correct
C	1	OMEPRAZOLE, (R)-	119141-89-8	S51HU491WJ	Correct
C	2	OMEPRAZOLE	73590-58-6	KG60484QX9	Correct
C	3	ESOMEPRAZOLE	119141-88-7	N3PA6559FT	Correct
D	1	esomeprazole	161973-10-0		Correct
D	2	omeprazole	73590-58-6 95510-70-6 95382-33-5 131959-78-9 172964-80-6 161796-78-7		Incorrect
D	3	esomeprazole			None
E	1	Omeprazole	73590-58-6		None

Table 2 Stitch keys and stitch values for the drug *imatinib mesylate*.

Stitch key	Stitch value
N_Name	IMATINIB MESYLATE; GLEEVEC; GLIVEC
I_UNII	8A1O1M485B
I_CAS	220127-57-1
I_CID	5291
I_CODE	STI-571; CHEMBL941
H_LYCHI_L5	7S4GKGNQ6N3X-N
H_LYCHI_L4	VLU17BQBSGWU-N; K83X3L3XSSHK-S
H_LYCHI_L3	VL3FPUQ59CU-N; K846NBMB7T3-S
R_activeMoiety	BKJ8M8G5HI

Table 3 Distribution of edge types for the stitch multigraph.

Stitch key	Size
H_LyChI_L3	6,140,942
H_LyChI_L4	5,300,078
N_Name	177,446
H_LyChI_L5	162,160
I_UNII	139,176
R_activeMoiety	14,940
I_CAS	11,684
I_CID	2,044

Table 4 Top stitch values for each stitch key. The LyChI hash keys L3 and L4 correspond to the potassium ion (K^+).

Stitch key	Stitch value	Size
H_LyChI_L3	VUSPQLGXN18-M	1,344,440
H_LyChI_L4	VU8BQZFPPYTZ-M	1,307,592
N_Name	ROFECOXIB	72
H_LyChI_L5	9DKQLD7D29DN-N	162,160
I_UNII	UNKNOWN	210
R_activeMoiety	2M83C4R6ZB	106
I_CAS	25322-68-3	1,806
I_CID	121225712	380

Table 5 Data sources used in the current version of STITCHER.

Data source	Size
G-SRS, April 2019	105,019
Withdrawn and Shortage Drugs List Feb 2018	674
Broad Institute Drug List 2018-09-07	6,125
NCATS Pharmaceutical Collection, April 2012	14,814
Rancho BioSciences, March 2019	51,591
Pharmaceutical Manufacturing Encyclopedia (Third Edition)	2,268
DailyMed Rx, January 2019	74,850
DrugBank, December 2018	11,922
DailyMed Other, January 2019	13,393
DailyMed OTC, January 2019	79,448
DrugsFDA & Orange Book, July 2019	28,256
ClinicalTrials, December 2017	305,833
OTC Monographs, December 2018	2,713
FDA NADA and ANADAs, December 2018	554
FDA Excipients, December 2018	10,212

Algorithm 1: Entity stitching algorithm

```
Let $W$ denote the set of stitch nodes created in the data ingestion step for a given data source $D$.
Let $\langle k, v \rangle$ be the tuple of stitch key and value, respectively, defined for a stitch node $w$.
$G = (V, E)$ is the current stitch multigraph.
Find($k, v$) is a function that returns all stitch nodes in $V$ containing stitch key $k$ and stitch value $v$.
Union($w, z$) is a union-find algorithm for tracking disjoint sets (i.e., connected components).
for $w \in W$ do
    for $\langle k_i, v_i \rangle \in w$ do
        for $z \in$ Find($k_i, v_i$) do
            $E \leftarrow E \cup z \sim w$
            Union($w, z$)
        end
    end
    $V \leftarrow V \cup w$
end
```

Algorithm 2: An algorithm to untangle a connected component

```
Let $U$ be the disjoint set data structure for all entities.
Let $S$ denote the set of unlabeled entities (i.e., singletons).
$C = (V, E)$ is the connected component.
MergeNodes($U, r$) is a function that performs transitive closure on stitch nodes in $C$ which are connected by a relation $r \in E$. The results are accumulated in $U$.
MergeCliques($U, K$) is a function that takes a set of stitch keys $K$, finds overlapping cliques that span two or more stitch keys, and performs transitive closure on the entities.
MergeSingletons($U, S, K$) is a function that also takes in a set of stitch keys $K$, a set of singleton stitch node $S$, and find the best mapping to an already labeled stitch node.
MergeNodes($U$,R_activeMoiety)
MergeNodes($U$,I_UNII)
MergeNodes($U$,H_LyChI_L4)
MergeCliques($U$,N_Name,I_CAS,I_CID,H_LyChI_L4)
MergeSingletons($U, S$,N_Name,I_CAS)
```