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 which 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 simple identifier lookups. 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 semantic concept for drug entity, we develop a set of equivalence relations we call *stitch keys* specifically for the ER of drug entities. We demonstrate the utility of our approach through the development of INXIGHT DRUGS (https://drugs.ncats.io), a new online resource that aggregates curated, reliable drug development data from multiple public sources, all in one place. To the best of our knowledge, this resource is currently the most comprehensive of its kind. The source code for STITCHER is available at https://github.com/ncats/stitcher.

Keywords: entity resolution; drug database; data integration

Introduction

As the volume of biological data continues to grow at an unprecedented rate [1], *entity resolution* (ER)—also commonly known as *data de-duplication* or *record linkage* [2]—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 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 a "disease entity" and its nosological classification [3] 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. Ironically, even the U.S. Food and Drug Administration (FDA) does not have a straightforward definition of the term "drug." The

Federal Food Drug and Cosmetic Act (FD&C Act) and the FDA regulations define "drugs," in part, by reference to the 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" [4]. More practically, the agency defines "drug substance" and "drug product," respectively, as a physical ingredient and a marketed product that contains said ingredient. Others use the word "drug" as a convenient shorthand to refer to both a "drug substance" and a "drug product", 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 [5].

The third definition of the word "drug" is commonly used in the literature and 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 to be the same drug: different salt forms (e.g., Sumatriptan Succinate and Sumatriptan Hemisulfate), prodrugs and their metabolized active forms (e.g., Brincidofovir and Cidofovir), etc. [6]. 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 [7].

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 definition 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. The FDA and other national regulators of medicines have collaborated to produce the ISO 11238 standard [8] which endeavors to define an information scheme for the unambiguous identification of all ingredients found in medicinal products, and the FDA uses an implementation of the ISO 11238 as the backbone of its information systems within the agency [9]. This facilitates data exchange within the FDA and with other national authorities. Nevertheless, the ability to map other, external data sources into this rigorously-defined scheme using whatever names and data are at hand remains a largely unresolved challenge.

Entity resolution (also known as *record linkage* in the literature) is the problem of determining which entities with partially shared attributes are equivalent across data sources. This is a fundamental challenge in data integration, and one that has

been an active area of research since the early days of computing [10]. Within the biomedical community, ER has been particularly instrumental in the analysis of electronic health records (EHRs) [11]. In the context of drug discovery, however, ER has received little attention; to our knowledge, the work by Croset et al. toward a drug product terminology [12] is the only recent effort directly addressing ER. The authors utilized graph density and betweenness centrality metrics to merge and identify problematic entities for each connected component within the link graph. However, due to the lack of available data and code, we were unable to comparatively evaluate their approach.

Herein we report on our effort to develop STITCHER, a robust ER framework for drug data integration. By leveraging the semantic concept of *active moiety* and other well-defined attributes (e.g., molecular hash keys), we reduce the ER problem to that of establishing equivalence relations over a prioritized set of attributes we call *stitch keys*. We demonstrate the utility of STITCHER by having built a new online resource, INXIGHT DRUGS (`https://drugs.ncats.io`), that aggregates curated, reliable drug development data from multiple public sources, all in one place. This open-access and easy-to-use online portal lists, as accurately as possible, all prescription and over-the-counter drug substances approved in the U.S. as well as those withdrawn from the U.S. market, drugs marketed globally, and investigational interventions. INXIGHT DRUGS is designed to help researchers obtain the data needed to repurpose or advance drugs to address unmet medical needs. To the best of our knowledge, INXIGHT DRUGS currently is the most comprehensive resource of its kind.

Preliminaries

As a motivating example, consider the data shown in Table 1. In this scenario, we would like to integrate data from multiple sources such that each data source might contain only partial (and possibly conflicting) information about the identities of the entities. There are four possible attributes that can be used to assert equivalence between the entities in this table. The `Structure` attribute indicates whether the entity is associated with a chemical structure and if it contains errors (e.g., missing or incorrect stereo assignments). Based on the definition of "drug" as *active moiety*, we would like to determine the number of unique drugs in the table [i]. Upon initial examination, we can immediately make the case for three equivalence classes $\{A_4, B_2\}$, $\{A_3, C_2\}$, and $\{A_2, C_3\}$. Further, we also know that A_3 is an active moiety of A_4 per our definition. Through a transitive closure, we now have the following equivalence classes: $\{A_4, B_2, A_3, C_2\}$ and $\{A_2, C_3\}$. Having performed all transitive closures 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 illustrated by this example, is to partition entities within a given dataset into disjoint sets such that those within the same set are considered

[i] 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.

equivalent. To achieve this, STITCHER first "stitches" together entities with shared identity attributes (*stitch keys*). Next, a transitive closure is performed based on heuristics that we have developed in assigning priorities to the attributes. Finally, the equivalence classes are efficiently derived through standard union-find algorithm [13]. This is the essence of STITCHER.

Core concepts

The conceptual data model underlying STITCHER is a *multigraph*. Within this multigraph, a node can either be a *data node* or a *stitch node*. Each data node represents a "raw" entity as ingested from the data source. A data node is connected to a corresponding stitch node, which contains its 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. A set of nodes connected via stitch keys with the same stitch value 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. The majority of connected components the ER produces are of reasonable sizes (e.g., 10 to 50 stitch nodes), and the real challenges center around effective strategies for the handling of very large connected components commonly known as *hairballs* [12]. For example, the current version of the INXIGHT DRUGS resource has a hairball containing nearly 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, where each instance perhaps reflects a specific algorithmic strategy or version. Figure 1 shows an example of a connected component with only one equivalence class (sgroup) as determined by the underlying ER algorithm.

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, a stitch key should reflect the true identity of the corresponding entity as specifically as possible. Depending on the entity type, its stitch key can be generic (e.g., synonym) or very specific (e.g., molecular hash key). For the "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: tradenames, INN (International Nonproprietary Names), USAN (United States Adopted Names), IUPAC (International Union of Pure and Applied Chemistry), etc.

`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 codes. `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 small molecules, their underlying chemical structure is, perhaps, more important than any other identifier. These stitch keys are hash values derived from the structure of a small molecule at different levels of detail [14]. We discuss the specifics of how these derived stitch values are generated in the next section.

`R_activeMoiety`. While technically not a stitch key, the active moiety relationship between two drugs provides a strong evidence of equivalence. Usually, this relationship can be inferred directly from the chemical structures (e.g., freebase and salt forms, esters, etc.). Structures with metal complexes and metabolites are a noteworthy exception, as in most such cases active moieties have to be assigned manually.

Table 2 shows examples of stitch keys and stitch values for the drug entity IMATINIB MESYLATE. In this example, the value listed for the `R_activeMoiety` stitch key is the UNII of IMATINIB, the freebase form of IMATINIB MESYLATE.

Methods

In general, data integration with STITCHER consists of four basic steps applied in order: *ingestion*, *stitching*, *entity resolution*, and *entity normalization*. With the exception of *entity resolution*, all other steps—as 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 from a wide variety of sources and in multiple formats, such as JSON, semantic formats (OWL, RDF, and Turtle), delimiter separated text, and even custom formats. For non-semantic formats, a separate configuration file is required to map data attributes 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., support for regular expressions) to help with data transformation during ingestion.

Perhaps the most unique feature of STITCHER is its ability to incorporate chemical structures into the ER process. 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 a "drug" is a mixture, prodrug, or active moiety with complex excipient (or derivative thereof). For example, consider the drug entity IMATINIB MESYLATE and its active ingredient IMATINIB. It is obvious that the two entities cannot be matched by the 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.

Notably, this trivial example might suggest that, instead of comparing names, one could find the longest common substring of the names. The approach would certainly work in this specific case, but consider a counter-example: OSELTAMIVIR ACID is an active moiety of the prodrug OSELTAMIVIR. Thus substring comparison makes for a viable general-purpose approach only upon implementation of highly specialized parsing rules and rich manually curated dictionaries.

For data sources with chemical structures, the most computationally demanding step in the data ingestion is the generation of molecular hash keys. Hash keys are generated for each component of a chemical structure at 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 `R_activeMoiety` relationship, a match at this level has the highest priority. The next level `L4` represents the structure after normalization and standardization per the LyChI software package [14]. A match at this level implies that structures being compared are equivalent in terms of stereochemistry, resonance, and tautomerism. Finally, as incorrect or missing stereo information is one of the most common type of errors associated with chemical structures, the last level `L3` removes stereochemistry from consideration entirely. A match at this level is thus considered weak and does not constitute equivalence without other supporting evidence. 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 multigraph is incrementally constructed as data are 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—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. The stitching algorithm also utilizes a union-find algorithm [13] 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 disjointed entity sets. Formally, only this step constitutes the *entity resolution* (ER), and it is the only step within STITCHER that is entity type specific. This is to be expected: given that ER essentially is a process of the entity identification, a reasonable amount of knowledge of the entity type is required for the adjudication to be sufficiently accurate. 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.

The aforementioned implied priorities associated with the stitch keys are at the core of the algorithm. The `R_activeMoiety` key has the highest priority as it is a unique manually generated relationship that is only available from G-SRS, , an FDA resource derived from their Substance Registration System. While it is possible for this relationship to be automatically inferred for specific cases (e.g., salt form and freebase), we currently rely on G-SRS to provide this specific semantic annotation. 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 no attributes in common, we know for a fact that *acetylsalicylic acid* is an *active moiety* of *ethyl acetylsalicylate*. Further examination of the structural differences shows that only an *ester* separates the two entities, and this falls well within the FDA definition of equivalent drugs that we adopted. This example also highlights a predicament: computationally, there is nothing to prevent us from attempting to impute *active moiety* relationships through efficient (sub-) graph isomorphisms. This very tempting proposition, however, is a trap (that we have thus far avoided), as other forms of *active moiety* relationships—e.g., metabolites, metals—are not amenable to such imputation, and attempting it would require considerable effort.

The next most important stitch key is `I_UNII`. UNII, Unique Ingredient Identifier, is the primary identifier developed and issued by the FDA. UNII is used as the primary identifier in the aforementioned G-SRS data source. Thus any other data source that provides mapping based on a UNII is likely to have "sufficient" knowledge of G-SRS ("guilt by association").

For entities that can be represented by chemical structures, the stitch key `H_LyChI_L4` has the next level of priority. We postulate that the `L4` hash describes a structure in a sufficiently detailed manner, entities with the same `L4` stitch value are not likely to match if the source data contains errors.

All other stitch keys (`N_Name`, `I_CAS`, `I_CID`, etc.) have the lowest priority.

At the completion of Algorithm 2, the disjointed set data structure U contains all equivalence entity classes, where each class is represented by an *sgroup* node in the stitch multigraph. The sgroup nodes are the *resolved entities*—in the context of this work, individual drugs.

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 attributes and (ii) conflict resolution (semantic as well as self-consistency). While this step can be quite trivial if the attributes are mutually exclusive across all data sources, addressing this in a general setting will require considerable effort in terms of understanding of the data sources and the concomitant metadata. Here, a common strategy is to preferentially choose attributes based on the perceived quality of the data sources. Revisit an example provided in Table 1. The attributes for the *normalized* entity that corresponds to the equivalence class $\{A_1, A_2, B_1, C_3, D_1, D_3\}$ can simply be the same as those of A_2 because we have reasons to believe that data source A is of higher quality than other data sources.

Results

With Stitcher serving as the data integration framework, we set out to build a comprehensive resource, InXight Drugs, listing, as accurately as possible, all prescription and over-the-counter drugs approved for marketing in the U.S., withdrawn from the U.S. market, marketed globally, and even investigational interventions. 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 [15, 16]. To the best of our knowledge, InXight Drugs is currently the most comprehensive resource of its kind.

Our starting point and the core/reference dataset is the aforementioned G-SRS data source source [17], as it is (i) well-curated, (ii) public, and (iii) naturally authoritative due to having been derived from the FDA's internal Substance Registration System [9]. Furthermore, G-SRS contains over 100K substances rigorously defined according to the ISO 11238 standard and spanning six different classes: chemical, polymer, protein, nucleic acid, mixture, and structurally diverse.

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 to 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 provide rigorous definitions for substances, they lack other information such as approval status, year, jurisdiction, indications, patents, publications, etc. The complete list of data sources currently used by Stitcher is shown in Table 5.

Availability

Stitcher and the data integration pipeline developed for the InXight Drugs resource are available as a repository at `https://github.com/ncats/stitcher`.

The InXight Drugs resource is available at `https://drugs.ncats.io`. The corresponding stitch multigraph built with the data sources listed in Table 5 is available as a Neo4j database 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. The complete list of sgroup nodes (i.e., equivalence classes) is available for browsing at `https://stitcher.ncats.io/app/stitches/latest`. 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 apparent from 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, not two.

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 reflects the lack of a 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

Data integration remains a major challenge for drug discovery as biomedical research continues to generate data at an unprecedented rate.

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. Searls, D.: Data integration: challenges for drug discovery **4**, 45–58 (2005)
2. Fellegi, I., Sunter, A.: A theory for record linkage. Journal of the American Statistical Assocation **64**(328), 1183–1210 (1969)
3. Hucklenbroich, P.: "Disease Entity" as the Key Theoretical Concept of Medicine. Journal of Medicine and Philosophy **39**, 609–633 (2014)
4. FDA: Human drugs. Accessed January, 2020 (2020). https://www.fda.gov/industry/regulated-products/human-drugs
5. NLM: RxNorm: Prescription for Electronic Drug Information Exchange. Accessed January, 2020 (2005). https://www.nlm.nih.gov/research/umls/rxnorm/RxNorm.pdf
6. FDA: New drugs at FDA. Accessed January, 2020 (2020). https://www.fda.gov/drugs/development-approval-process-drugs
7. FDA: Drugs for Human Use. Accessed January, 2020 (2012). https://www.govinfo.gov/content/pkg/CFR-2012-title21-vol5/pdf/CFR-2012-title21-vol5-chapI-subchapD.pdf
8. ISO: Health informatics—Identification of medicinal products—Data elements and structures for the unique identification and exchange of regulated information on substances. Accessed January, 2020. https://www.iso.org/standard/69697.html
9. FDA: FDA's Global Substance Registration System. Accessed January, 2020 (2019). https://www.fda.gov/industry/fda-resources-data-standards/fdas-global-substance-registration-system
10. Newcombe, H.B., Kennedy, J.M., Axford, S.J., James, A.P.: Automatic linkage of vital records. Science **130**, 954–959 (1959)
11. Karr, A., Taylor, M., West, S., Setoguchi, S., Kou, T., Gerhard, T., Horton, D.: Comparing record linkage software programs and algorithms using real-world data **14**(9) (2019). doi:10.1371/journal.pone.0221459
12. 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
13. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, USA, ??? (2001)
14. NCATS: Layered Chemical Identifier. Accessed October, 2019 (2013). https://github.com/ncats/lychi

15. 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)
16. 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
17. FDA: G-SRS substance database. 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 largest node is an entity called "sgroup node" derived from the entity resolution process that establishes equivalence between the stitch nodes.

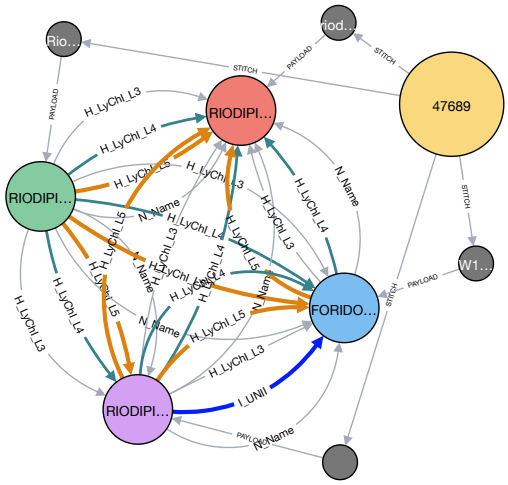
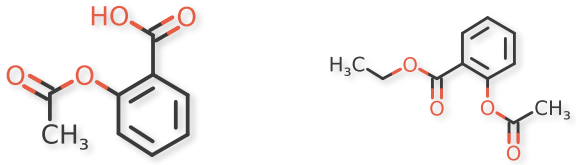


Figure 2 Chemical structures for (a) acetylsalicylic acid and (b) ethyl acetylsalicylate.



(a) (b)

Tables
Algorithms

Figure 3 A connected component for ASPIRIN.

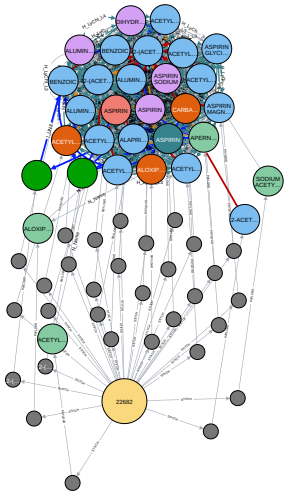


Figure 4 A connected component for LEVOMETHADYL that clearly shows two distinct clusters.

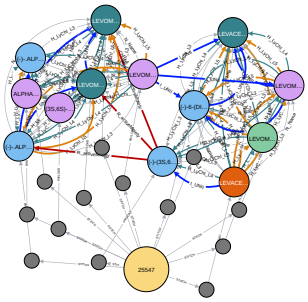


Figure 5 A dense connected component for BENOXAPROFEN that resolved to three unique entities.

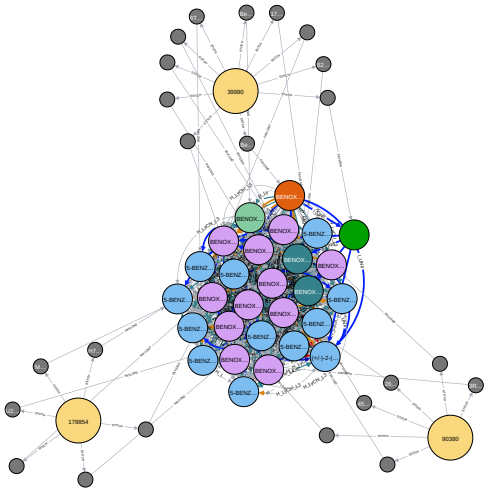


Table 1 An example of integrating data from multiple sources where each source contains only partial information.

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, July 2020	74,850
DrugBank, July 2020	11,922
DailyMed Other, July 2020	13,393
DailyMed OTC, July 2020	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 ∈ W do
    for ⟨k_i, v_i⟩ ∈ w do
        for z ∈ Find(k_i, v_i) do
            E ← E ∪ z ∼ w
            Union(w, z)
        end
    end
    V ← V ∪ 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)
```