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

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 if 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 in Stitcher 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 a comprehensive resource of drugs that have either been marketed or approved in the United States for human use.

# 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* (ER)—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 is what makes disease ER extremely challenging.

Drug is another entity type that is also challenging for ER due to the 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.” More practically, the agency defines “drug substance” and “drug product,” respectively, as a physical ingredient found in a marketed product. Others use the word “drug” as 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 .

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 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. 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 .

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, 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. 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.

Herein we report on our recent data integration effort to build InXight Drugs, 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. 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 [tab:data-integration-example], which illustrates a common scenario. 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 that it possibly contains errors (e.g., missing or incorrect stereo assignments). Based on the definition of “drug” as an *active moiety*, we would like to determine the number of unique drugs in the table. Upon initial examination, we can immediately make the case for three equivalence classes , , and . Further, we also know that is an active moiety of per our definition. Through a transitive closure, we now have the following equivalence classes: and . (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.) Having performed all transitive closures on the shared attributes in order, we eventually arrive at the final three equivalence classes , , and , which correspond to three distinct isomers (-, -, and racemic, respectively) of the drug Omeprazole. (Note that the -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 equivalent. To achive this, Stitcher first “stitches” together entities with shared identity attributes we call *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. 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 *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 stitch node, which contains its *standardized* representation that is used for *stitching*. An edge between two stitch nodes can either be a *stitch key* (undirected) or a *relationship* (directed). A unique *stitch value* is associated with each stitch key, and a set of nodes connected via stitch keys with the same stitch value forms a *clique*. Figure [fig:graph1] 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 (ca. 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*. For example, the current version of the InXight Drugs database 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 [fig:graph1] 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 the small molecules, their underlying chemical structure is, perhaps, more important than any other identifier. These stitch keys are hash values derived from a molecular structure at different levels of detail. Section [sec:methods-ingest] discusses in detail how these derived stitch values are generated.

R\_activeMoiety. While technically not a stitch key, the active moiety relationship between two drugs provides a strong evidence of equivalence. In most cases, 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 [tab:imatinib] 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 executed 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 in a wide variety of sources and formats, such as JSON, semantic formats (OWL, RDF, and Turtle), delimiter separated text, and custom formats. For non-semantic formats, 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., 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 to make it work in general would require very 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. 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 [tab:imatinib] 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 [algo:stitching] describes the basic stitching algorithm of Stitcher. This algorithm is applied to each data source, and upon completion produces a stitch multigraph such that any stitch value that spans stitch nodes is an induced clique—a complete subgraph of nodes and 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 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 particular 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 [algo:untangle] 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. As an example, consider the entities *acetylsalicylic acid* (or also commonly known as *aspirin*) and *ethyl acetylsalicylate* shown in Figure [fig:aspirin]. 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 definition of equivalent drugs. This example also highlights a quagmire: 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 resisted), 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 by the FDA and used 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, and entities with the same L4 stitch value are not likely to match if the source data contains errors.

All other stitch keys (i.e., N\_Name, I\_CAS, I\_CID) have the lowest priority.

At the completion of Algorithm [algo:untangle], the disjointed set data structure 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 this specific example, 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 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 of a given data source and the concomitant 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 [tab:data-sources] 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, enumerating drugs that have either been marketed or approved for human use in the United States. Our starting point and the core/reference dataset is the aforementioned G-SRS 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. 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 [tab:data-sources].

## 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 data sources listed in Table [tab:data-sources] 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 [tab:stitch-keys] and [tab:stitch-values] 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 [fig:ASPIRIN] 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 [fig:LEVOMETHADYL], 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 [fig:BENOXAPROFEN] 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

# Competing interests

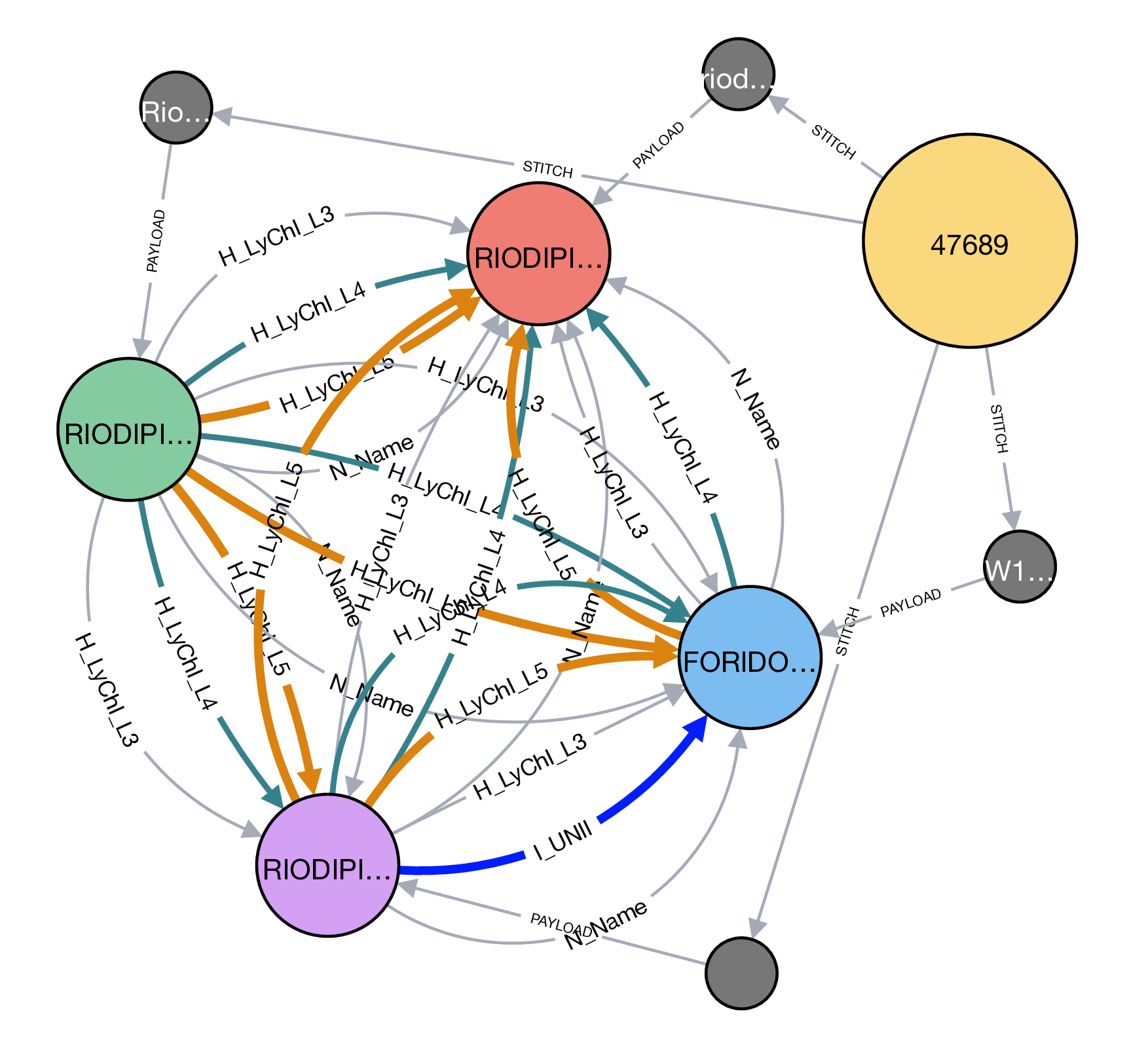
The authors declare that they have no competing interests.

# Author’s contributions

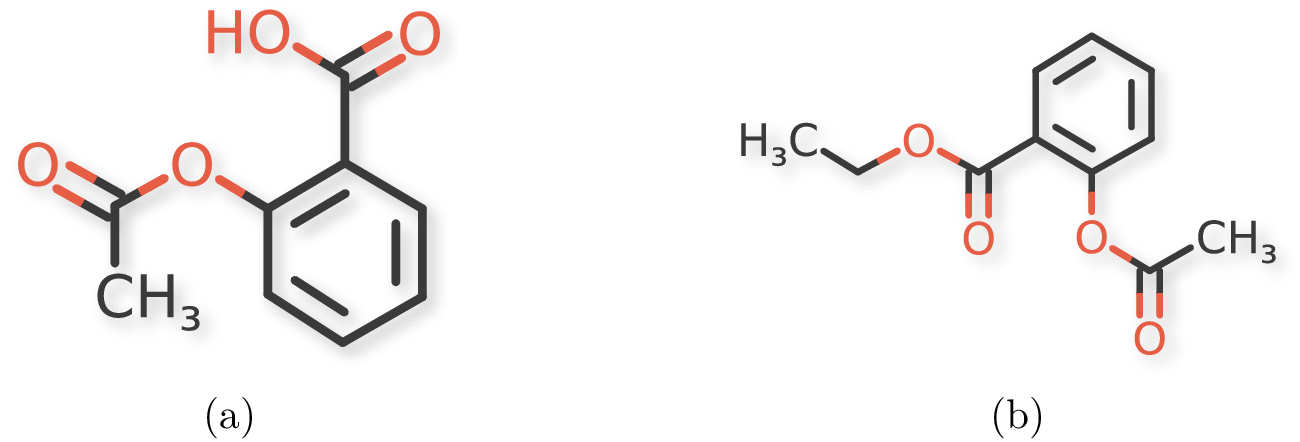
# Acknowledgements

We thank our colleagues Mark Williams and Tyler Beck for their valuable proofreading 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.

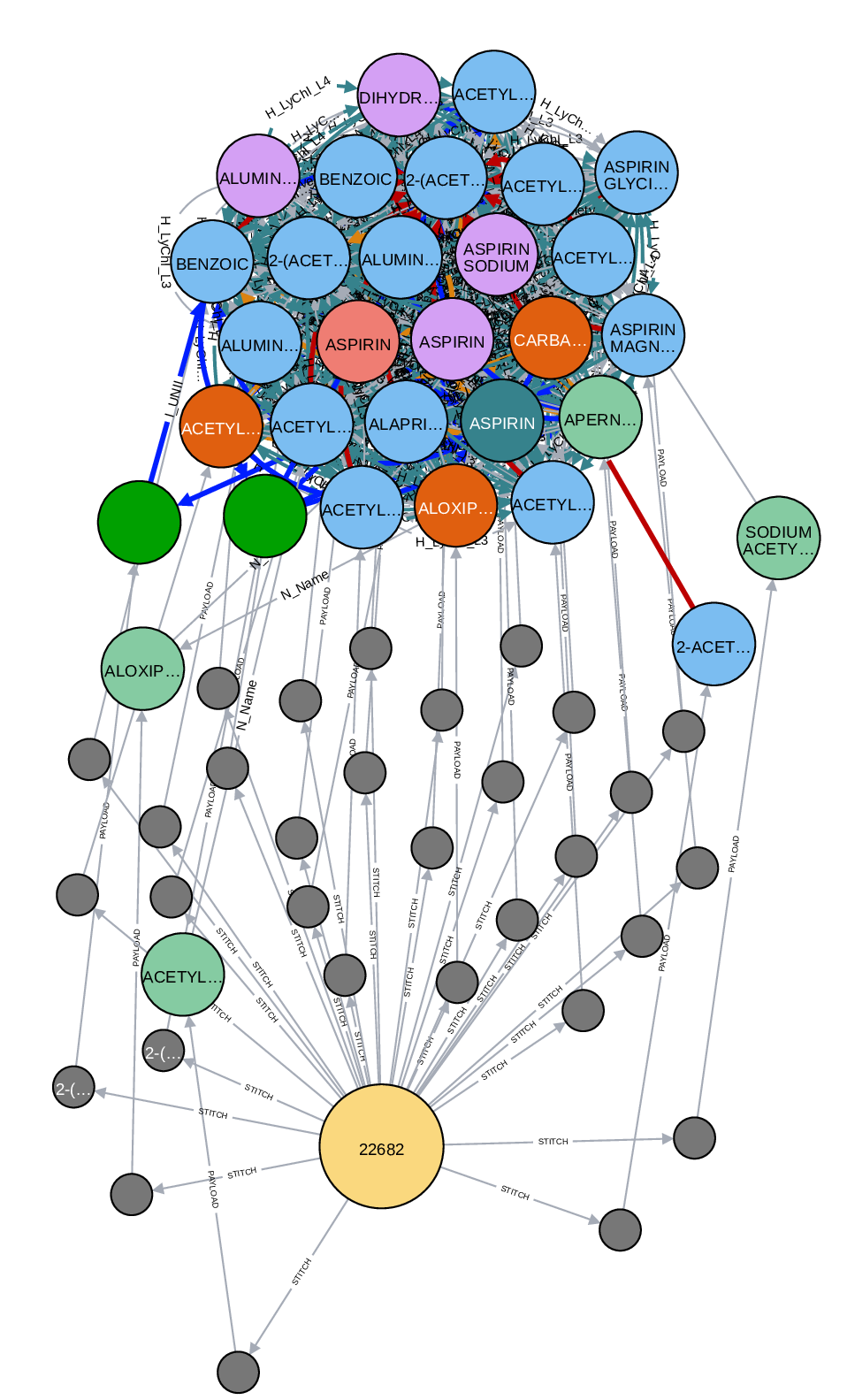
# Figures



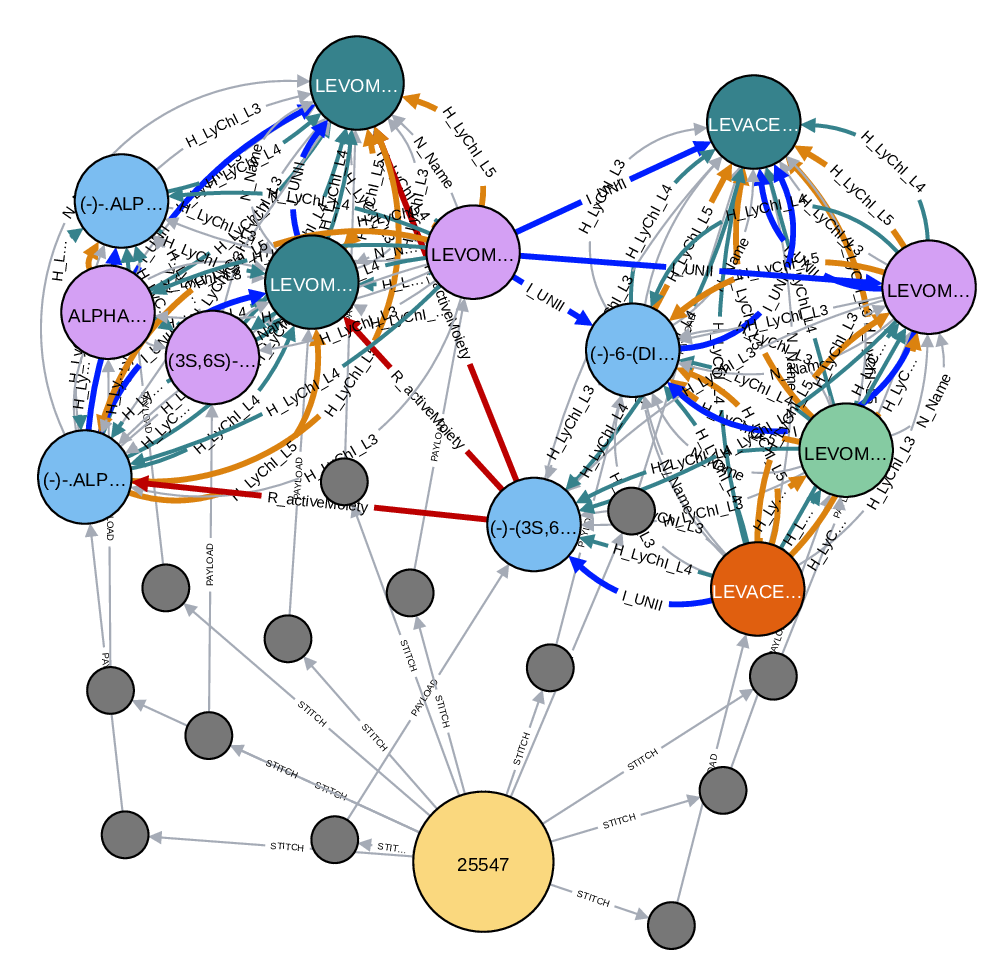
A connected component in the stitch multigraph with four stitch nodes (medium) and corresponding data nodes (small). Each stitch value forms a clique within this connected component. The edge labels between stitch nodes are the stitch keys. The largest node is an entity called “sgroup node” derived from entity resolution that establishes equivalence between the stitch nodes.



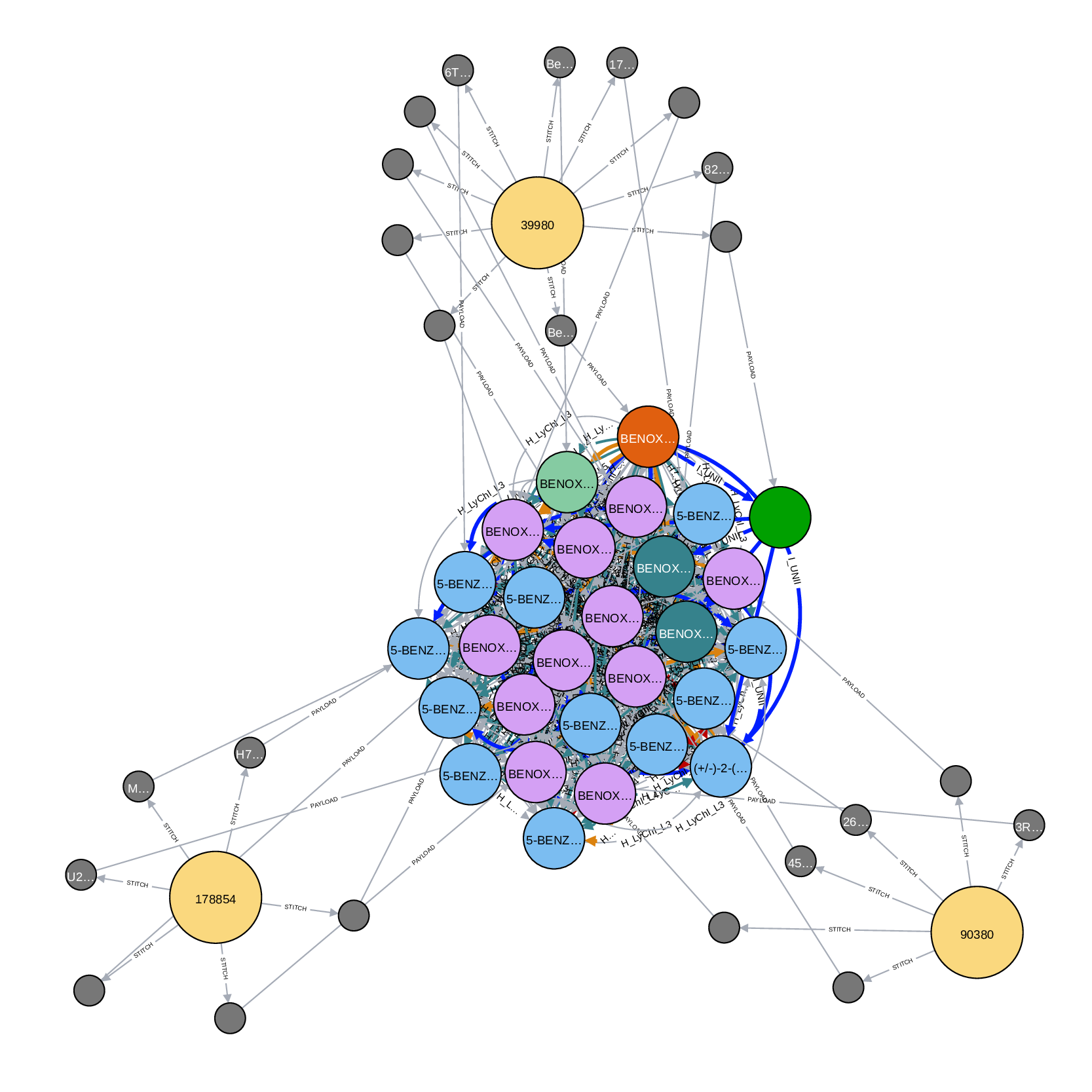
Chemical structures for (a) acetylsalicylic acid and (b) ethyl acetylsalicylate.



A connected component for ASPIRIN.



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



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

# Tables

An example of integrating data from multiple sources where each source contains only partial information.[tab:data-integration-example]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source | ID | Name | CAS | UNII | Structure |
| A | 1 |  | 914613-86-8 | SCC2RK476A | Correct |
| A | 2 | ESOMEPRAZOLE | 217087-09-7 | N3PA6559FT | Correct |
| A | 3 | OMEPRAZOLE | 95382-33-5 | KG60484QX9 | Correct |
| A | 4 |  | 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 |  |  | Incorrect |
| D | 3 | esomeprazole |  |  | None |
| E | 1 | Omeprazole | 73590-58-6 |  | None |

Stitch keys and stitch values for the drug imatinib mesylate.[tab:imatinib]

|  |  |
| --- | --- |
| 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 |

Distribution of edge types for the stitch multigraph.[tab:stitch-keys]

|  |  |
| --- | --- |
| 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 |

Top stitch values for each stitch key. The LyChI hash keys L3 and L4 correspond to the potassium ion (K).[tab:stitch-values]

|  |  |  |
| --- | --- | --- |
| 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 |

Data sources used in the current version of Stitcher.[tab:data-sources]

|  |  |
| --- | --- |
| 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 |

# Algorithms

[algo:stitching] Let denote the set of stitch nodes created in the data ingestion step for a given data source .  
Let be the tuple of stitch key and value, respectively, defined for a stitch node .  
 is the current stitch multigraph.  
is a function that returns all stitch nodes in containing stitch key and stitch value .  
is a union-find algorithm for tracking disjoint sets (i.e., connected components).

[algo:untangle] Let be the disjoint set data structure for all entities.  
Let denote the set of unlabeled entities (i.e., singletons).  
 is the connected component.  
is a function that performs transitive closure on stitch nodes in which are connected by a relation . The results are accumulated in .  
is a function that takes a set of stitch keys , finds overlapping cliques that span two or more stitch keys, and performs transitive closure on the entities.  
is a function that also takes in a set of stitch keys , a set of singleton stitch node , and find the best mapping to an already labeled stitch node.