

Unlocking Insights in the Life Sciences Domain through
Knowledge Graph Construction and Hypothesis Generation
Using Machine Learning

By

JAE SUNG YOUN
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Ilias Tagkopoulos, Chair

Justin B. Siegel

Xin Liu

Committee in Charge

2024

Acknowledgments

I thank my wife, Xing Liu, whose constructive criticisms have been my guiding light throughout the tumultuous journey of graduate school. My daughter, Yelynn, has been my little inspiration, sharing the experience of academic pursuit as we both navigated the world of learning together. I am immensely thankful to my parents, whose unconditional support has been the bedrock of my academic endeavors, providing assistance and encouragement without hesitation.

I am grateful to my advisor, Professor Ilias Tagkopoulos, for believing in me and offering me the opportunity to embark on this transformative journey. The lessons I learned from Ilias during my Ph.D. transcended the realm of academia, shaping not only my skills as a researcher but also nurturing personal growth. I also thank Ilias for his understanding and accommodation of my familial responsibilities, providing invaluable support during the challenges of supporting my family throughout my Ph.D. I extend my appreciation to the members of my dissertation committee, Professor Xin Liu and Professor Justin Siegel, for their insightful feedback that significantly contributed to the completion of my research and the refinement of this dissertation. Finally, a special acknowledgment goes to the members of the Tagkopoulos lab, Navneet, Xiaokang, Beatriz, Ameen, Gabriel, Tarini, Christian, Erol, Trevor, Cheng-En, Fang, Arielle, Adil, Bobby, Miachel, and Keer for their friendship and support, making my time in the lab enriching and memorable.

Abstract

In the dynamic landscape of life sciences data, the inherent noise and lack of machine-friendliness present significant challenges. The pressing need arises to transform this complex and unstructured data into a machine-friendly format, fostering efficient utilization for the generation of novel scientific discoveries in a faster and more cost-effective manner. This dissertation presents a comprehensive exploration of automated knowledge management and discovery across various domains using advanced machine learning techniques. We first address the challenges associated with the manual creation and maintenance of food ontologies. A semi-supervised framework employing word embeddings is proposed, demonstrating an 89.7% improvement in precision compared to the expert-curated FoodOn ontology. Second, a machine learning framework is introduced for automated knowledge discovery through the construction of a comprehensive *Escherichia coli* antibiotic resistance knowledge graph. Iterative link prediction and wet-lab validation led to the identification of 15 antibiotic-resistant genes, including 6 previously unassociated with antibiotic resistance. Third, the Knowledge Graph Language Model (KGLM), which incorporates a novel entity/relation embedding layer, achieves state-of-the-art performance in link prediction tasks on benchmark datasets. Finally, an integrated pipeline is presented for the automated generation of large-scale knowledge graphs in an active learning setting. Applied to 155,260 scientific papers, the pipeline extracts 230,848 food-chemical composition relationships, the largest in the domain. This dissertation exemplifies evidence-driven decisions in automating knowledge discovery, providing high confidence, and accelerating the pace compared to traditional methods.

Table of contents

Chapter 1 Introduction.....	1
1.1 Introduction to knowledge graphs and ontologies	1
1.2 Current efforts for knowledge graph construction.....	2
1.2.1 Existing knowledge graphs.....	2
1.2.2 Automated knowledge graph construction.....	3
1.3 Limitations and challenges.....	4
1.4 Knowledge graph completion	5
1.5 Overview of the dissertation	7
Chapter 2 Using word embeddings to learn a better food ontology.....	8
2.1 Introduction	8
2.2 Methods	10
2.2.1 Data preprocessing and training of word embeddings.....	10
2.2.2 Ontology population.....	11
2.2.3 Evaluation metrics of the ontology structure	12
2.2.4 Success metric of ontology population	13
2.3 Results	14
2.3.1 Structural topology of FoodOn.....	14
2.3.2 Granularity and cohesiveness impair the precision of automated methods.....	14

2.3.3	Learning ontology via embeddings leads to substantially better performance	15
2.4	Discussion.....	16
Chapter 3 Knowledge integration and decision support for accelerated discovery of antibiotic resistance genes		
3.1	Introduction	23
3.2	Methods	26
3.2.1	Knowledge graph constructor.....	26
3.2.2	Inconsistency resolver.....	27
3.2.3	Hypothesis generator.....	28
3.2.4	Wet-lab validation.....	31
3.3	Results	32
3.3.1	The landscape of <i>E. coli</i> antibiotic resistance genes and processes.....	32
3.3.2	Resolved inconsistencies help discover new knowledge.....	33
3.3.3	KIDS accelerates knowledge discovery.....	34
3.3.4	AI-driven discovery of 6 antibiotic resistance genes.....	36
3.4	Discussion.....	37
Chapter 4 Integrating knowledge graph structure in language models for link prediction		
4.1	Introduction	51

4.2	Background.....	52
4.3	Proposed approach.....	53
4.4	Experiments and results.....	56
4.4.1	Datasets	56
4.4.2	Settings.....	56
4.4.3	Link prediction results	57
4.5	Conclusion	58
Chapter 5 Automated knowledge extraction of food and chemicals from the literature .		64
5.1	Introduction	64
5.2	Methods	66
5.2.1	Premise-hypothesis pair generation.	66
5.2.2	Premise-hypothesis pair annotation.	67
5.2.3	Entailment model.....	68
5.2.4	Active learning strategy.	69
5.2.5	Knowledge graph generation.....	70
5.2.6	Link prediction.	71
5.2.7	Link prediction literature validation.	72
5.3	Results	73
5.3.1	The FoodAtlas Knowledge Graph contains a wide spectrum of food-chemical composition information.....	73

5.3.2	FoodAtlas discovers complementary information to benchmark datasets.	74
5.3.3	Maximum likelihood active learning strategy discovers knowledge 38% faster than without.	75
5.3.4	Sources of error and impact of large language model general knowledge.	76
5.3.5	Link prediction, GPT model, and the impact of ontologies in performance.	77
5.3.6	Link prediction reveals previously unknown food-chemical relationships. .	77
5.3.7	AI-driven discovery of six food-chemical relationships.....	78
5.4	Discussion.....	79
Chapter 6	Conclusion.....	94
Appendix	132

Chapter 1

Introduction

1.1 Introduction to knowledge graphs and ontologies

A knowledge graph (KG) is defined as a directed, multi-relational graph where entities (nodes) are connected with one or more relations (edges)¹. It is represented with a set of triplets, where a triplet consists of (*head entity, relation, tail entity*) or (*h, r, t*) for short. In the contemporary landscape of information and data-driven decision-making, the concept of knowledge graphs, a term first coined by Google in 2012², has emerged as a pivotal paradigm for representing and organizing knowledge in a structured and interconnected manner³. By leveraging semantic relationships, knowledge graphs facilitate a nuanced comprehension of information, enabling more sophisticated queries and analyses. This inherent semantic richness empowers applications ranging from information retrieval^{4,5} and recommendation systems^{6,7} to advanced artificial intelligence algorithms that thrive on a deeper understanding of contextual relationships.

A knowledge graph is a powerful framework that transcends traditional data models by capturing the relationships and connections inherent in information, providing a more holistic and context-rich representation of knowledge. Compared to relational databases that store data in tables with predefined schemas and are often limited by their scalability⁸, graph databases are optimized for the efficient processing of dense, interrelated datasets,

and allow for fast traversals along the edges between vertices^{9,10}. This design structure has a few advantages including the ability to detect correlations and patterns, discover explanations, and construct predictive models^{11–13}.

An ontology, which shares a similar structure as the knowledge graph, is defined as the body of formally represented knowledge in some area of interest expressed by objects and concepts, and the relationships that hold among them¹⁴. The structure of an ontology is based on the triplet of (*subject, predicate, object*) similar to that of knowledge graphs¹⁵, yet there exist subtle distinctions. Ontologies are usually smaller in size, are domain-specific, capture complex relationships between the classes and instances, and can enforce their structure by applying sets of restrictions and rules^{16,17}. Moreover, compared to the multi-relational knowledge graphs where different types of predicates can exist, ontologies connect concepts predominantly through subsumption or hypernymy relationships.

1.2 Current efforts for knowledge graph construction

1.2.1 Existing knowledge graphs

Due to their effectiveness in identifying patterns among data and gaining insights into the mechanisms of action, associations, and testable hypotheses^{18,19}, there is a constant effort to construct knowledge graphs in diverse domains. Some of the prominent examples include YAGO²⁰, NELL²¹, Freebase²², and Google Knowledge Graph²³ for storing general facts like people, cities, etc. Notable examples in the biological domain

include KnowLife²⁴, which houses information regarding health and life sciences, and BioGraph²⁵, which comprises information between biomedical entities. In the case of antibiotics, limited efforts curate large-scale knowledge graphs for predicting drug-drug similarity, and contain comprehensive molecular information, mechanisms, interactions, and targets about drugs^{26,27}. To curate the relationship between chemicals and human health, knowledge bases like CTD²⁸ and KEGG²⁹ curate information about the interactions among chemicals, genes, and/or disease entities, while other resources, including ChemFont³⁰ and GO³¹, are dedicated to creating an ontology of chemicals.

When it comes to food production and composition, various initiatives have proposed data repositories and ontologies regarding ingredients, processes, and final food products. Some examples of food compositional databases are USDA's FDC³² which provides nutrient composition data for approximately 300,000 food entries and FooDB³³ which provides quantitative chemical composition data in foods covering 80,000 chemicals in 800 foods. Other databases highlight non-ontological aspects, for instance, the GPC database³⁴ contains barcodes for food products, the European EFSA database³⁵ is a 32-feature categorization system, KNApSACK³⁶ houses information regarding 24,704 plant species and 62,647 metabolites, and Phenol-Explorer³⁷ documents 501 polyphenol contents of 459 foods. Concomitantly, there are multiple ontologies in various stages of development and usage^{38,39}. A notable example is FoodOn³⁹, an open-source and formal food ontology curated by the FoodOn consortium, which represents food by its properties, and adheres to the FAIR standards⁴⁰.

1.2.2 Automated knowledge graph construction

While knowledge bases have traditionally been curated manually from text data⁴¹, recent approaches utilize deep learning-based models for constructing knowledge bases. For example, language models were utilized to construct domain-specific knowledge graphs from unstructured texts^{42–47}, with some combined with active learning (AL) to reduce human annotation^{48–50}. Moreover, relation extraction models, a task in natural language processing that extracts semantic relations between entities in natural language sentences⁵¹ (e.g., given a sentence ‘Joe Biden is the president of United States’, a relation extraction model extracts a relation of ‘*isPresidentOf*’ between the entities ‘Joe Biden’ and ‘United States’), have been used for constructing knowledge bases⁵². Aside from simply extracting knowledge from natural language sentences that require a well-defined knowledge graph schema to populate the extracted relationships, there also exist efforts to automatically construct knowledge graphs from scratch without any manual intervention^{53,54}.

1.3 Limitations and challenges

Existing approaches to creating and expanding these knowledge bases are often bottlenecked by the need for time-consuming manual annotation processes that often require the expertise of domain experts. Although manual extraction by the domain experts is often precise, it does not scale well with bibliographic literature sources such as PubMed⁵⁵, which contains 34 million citations and abstracts, and PubMed Central (PMC)⁵⁶, which includes 7.6 million full-text scientific literature articles. From PMC, we estimate we can extract at least 2 million unique food-chemical associations from the

unstructured text data (**Appendix A.3.1.2**). The sheer amount of available scientific literature necessitates the need for an automated framework for constructing knowledge graphs. Moreover, these KGs often suffer from incompleteness. For example, 71% of the people in FreeBase have no known place of birth⁵⁷. To address this issue, knowledge graph completion (KGC) methods aim at connecting the missing links in the KG.

When it comes to creating the AI-ready knowledge graph in the domain of biological sciences, key challenges still exist despite the numerous attempts at applying the knowledge graphs. First, although knowledge conflicts exist and are reported in the literature^{58,59}, these inconsistencies are not curated in the existing knowledge bases. The inconsistencies are especially prevalent between high-throughput measurements and biological networks, making it non-trivial to draw biologically meaningful conclusions in an automated way⁶⁰. Second, negative findings are not curated well in existing biological knowledge bases despite their importance in training the machine learning models to classify what knowledge is likely to be true⁶¹. Finally, existing knowledge bases do not annotate temporal information about antibiotic exposure despite its importance in the emergence of antibiotic resistance^{62,63}, which obscures understanding of the time series dynamics of antibiotic resistance mechanisms.

1.4 Knowledge graph completion

The incomplete nature of knowledge graphs spurred the research topic of knowledge graph completion⁶¹, a task commonly referred to as link prediction in the field of statistical relational learning (SRL)⁶⁴ and aims at automatically augmenting the missing knowledge.

For instance, suppose that a given knowledge graph does not contain information about Obama’s birthplace. An SRL model can use pre-existing related facts about Obama (such as his profession being US president) to infer that he was born in the USA. In the case of antibiotic resistance, there is a strong possibility that all genes conferring resistance to a certain antibiotic are not currently represented in the knowledge graph. This leads to the use of machine learning to perform knowledge graph completion.

Graph feature models like path ranking algorithm (PRA)^{65,66} attempt to solve the KGC tasks by extracting the features from the observed edges over the KG to predict the existence of a new edge⁶¹. For example, the existence of the path *Jennifer Gates* $\xrightarrow{\text{daughterOf}}$ *Melinda French* $\xleftarrow{\text{divorcedWith}}$ *Bill Gates* can be used as a clue to infer the triplet (*Jennifer Gates*, *daughterOf*, *Bill Gates*). Other popular types of models are latent feature models such as TransE⁶⁷, TransH⁶⁸, and RotatE⁶⁹ where entities and relations are converted into a latent space using embeddings. TransE, a representative latent feature model, models the relationship between the entities by interpreting them as a translational operation. That is, the model optimizes the embeddings by enforcing the vector operation of head entity embedding \mathbf{h} plus the relation embedding \mathbf{r} to be close to the tail entity embedding \mathbf{t} for a given fact in the KG, or simply $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

Recently, pre-trained language models like BERT⁷⁰ and RoBERTa⁷¹ have shown state-of-the-art performance in all of the natural language processing (NLP) tasks. As a natural extension, models like KG-BERT⁷² and BERTRL⁷³ that utilize these pre-trained language models by treating a triplet in the KG as a textual sequence, e.g., (*Bill Gates*, *founderOf*, *Microsoft*) as ‘*Bill Gates founder of Microsoft*’, have also shown state-of-the-art results on

the downstream KGC tasks. Although such *textual encoding*⁷⁴ models are generalizable to unseen entities or relations⁷³, they still fail to learn the intrinsic structure of the KG as the models are only trained on the textual sequence. To solve this issue, a hybrid approach like StAR⁷⁴ has recently been proposed to take advantage of both latent feature models and textual encoding models by enforcing a translation-based graph embedding approach to train the textual encoders. Yet, current textual encoding models still suffer from entity ambiguity problems⁷⁵ where an entity *Apple*, for example, can refer to either the company Apple Inc. or the fruit. Moreover, there are no ways to distinguish forward relation (*Jennifer Gates, daughterOf, Melinda French*) from inverse relation (*Melinda French, daughterOf¹, Jennifer Gates*).

1.5 Overview of the dissertation

This document consists of four studies covering knowledge graph construction and hypothesis generation using machine learning in the domains of food science and biomedical science domains. Chapter 2 addresses the challenge of creating a better food ontology using word embeddings. Chapter 3 utilizes statistical machine learning methods to find novel antibiotic resistance genes using manually curated knowledge about *E. coli* antibiotic resistance. Chapter 4 proposes a state-of-the-art knowledge graph completion method that utilizes natural language processing techniques for better hypothesis generation. Chapter 5 addresses semi-automated knowledge graph construction using natural language processing in an active learning setting. We conclude in Chapter 6.

Chapter 2

Using word embeddings to learn a better food ontology

Disclaimer. All the work that is presented in this chapter has been published in *Frontiers in Artificial Intelligence*⁷⁶.

2.1 Introduction

In the realm of knowledge representation, ontologies serve as structured frameworks that formalize and define the relationships among entities within a particular domain. These conceptual models not only capture the semantics of a domain but also provide a shared understanding of human and machine reasoning⁷⁷. As we delve into the landscape of information organization, ontologies seamlessly intertwine with the concept of knowledge graphs. Knowledge graphs, characterized by their interconnected entities and relations, extend the principles of ontologies into a dynamic, graph-based structure. By leveraging ontological principles, knowledge graphs transcend static categorizations, facilitating the representation of complex relationships and enabling a more fluid, interconnected representation of knowledge. The synergy between ontologies and knowledge graphs thus becomes integral in enhancing the depth and flexibility of knowledge representation systems, paving the way for more nuanced and context-aware applications.

Due to the structural similarities between knowledge graphs and ontologies, several methods developed for the knowledge graph can also be applied to the area of ontology learning which includes tasks ranging from creating ontologies to extending and populating existing ontologies. However, in practice, the choice of embedding depends on the available corpus, and the method is specific to the task at hand. A task commonly seen in knowledge graphs is link prediction where the starting state is a knowledge graph and the end result is a more accurate and/or more complete knowledge graph. Link prediction uses methods that explain the triplets using latent features like Poincaré embeddings⁷⁸ or extract triplets using contextual patterns from some text data. In the area of ontology learning, word embeddings created from text data are used to create and populate an ontology in a one-shot fashion using unsupervised methods like clustering⁷⁹, or to populate a skeleton knowledge graph initialized with seed instances in an iterative fashion^{21,80}. As we move towards a detailed atlas of chemical food composition⁸¹, there is a current and present need for tools and frameworks that are data-driven and automated, to support the creation and/or extension of evidence-based, detailed ontologies at scale.

Here, we address the challenge of how to populate new instances into an existing ontological structure. We introduce LOVE (Learning Ontologies Via Embeddings), a semi-supervised framework for automated ontology population (**Figure 2.1**), that uses word embeddings trained on a corpus obtained from Wikipedia. The required memory and computational time of the proposed method scales linearly with the increasing number of instances. LOVE was applied to the FoodOn dataset to create the first food ontology using word embeddings. We evaluate the predicted ontology against FoodOn and achieve an

increased precision of 89.7% when compared to the best alternate non-embedding-based method that uses Hamming distance (0.34 vs. 0.18 respectively, with baseline precision of 4.7×10^{-4}).

2.2 Methods

2.2.1 Data preprocessing and training of word embeddings

There are a total of 2,764 classes and 10,865 instances in FoodOn. Every class or food instance is identified by its label. For example, 'cow milk cheese' is a class label, and 'Brie cheese food product' is a food instance label. These labels are constructed using 4,139 unique words (e.g. 'cow', 'milk', 'cheese', 'brie', 'food', 'product'). We searched both the labels and their unique constituent words to obtain corresponding Wikipedia pages (**Figure 2.1**), which we refer to as Wikipedia corpus. We preprocessed the corpus as follows: lower-case conversion, synonym mapping, punctuation stripping, white space stripping, numeric stripping, stop-word removal, short words stripping, and lemmatization. Note that the Wikipedia corpus consists of 142,948 unique words. For their training, we used the gensim⁸² implementation of the word2vec skip-gram model⁸³. Default settings of the gensim word2vec model were used except for the following parameters: number of epochs of 100, window size of 5, and minimum count of 1. We trained 4 different dimensions of word embeddings for word2vec: 50d, 100d, 200d, and 300d. In addition to word2vec, we also tested using the pre-trained word embeddings trained with GloVe⁸⁴ and fastText^{85,86}. For GloVe, we downloaded pre-trained word embeddings of dimensions 50d, 100d, 200d, and 300d known as glove.6B. For fastText word embeddings, we used

two different versions of word embeddings of size 300d that have been trained using different training corpora. Please refer to **Table 2.1** for complete information.

2.2.2 Ontology population

As illustrated in **Figure 2.1**, our algorithm aims to map a food instance (e.g. ‘plum’) through an ‘is a’ relationship to its parent (e.g. ‘fruit’ ideally), which we refer to as its target class. If we let i be a food instance and c be a target class, then $i \in I$ and $c \in C$, where I is the group of all food instances we seek to map, and C is the group of target classes to which we map the food instance. We also define I_c to be all the food instances within a class c . To map the instance to its appropriate target class, we propose an approach based on the similarity of word embeddings. Our criteria for optimal population consider a linear combination of two scores

$$score(c; i) = \alpha \cdot score_{siblings}(c; i) + (1 - \alpha) \cdot score_{parent}(c; i),$$

where α controls the ratio of the two terms. The $score_{siblings}$ is the similarity of the food instance i with the seed instances in I_c :

$$score_{siblings}(c; i) = sim\left(\vec{i}, \sum_{i' \in I_c} \frac{\vec{i'}}{|I_c|}\right),$$

where \vec{i} is the word embedding vector created by taking the average of the constituent word embeddings, $|I_c|$ is the number of all the seed instances in I_c , and $sim()$ is the measure of similarity between the two word embedding vectors. The $score_{parent}$ is the similarity of the food instance i with the target class c :

$$score_{parent}(c; i) = sim(\vec{i}, \vec{c}).$$

Finally, predicting which target class \bar{c} the food instance i will get mapped to can be formulated as follows:

$$\bar{c} = arg \max_c score(c; i).$$

For the scope of this work, we map the food instance to a single target class even if it was originally mapped to multiple classes. For the case of FoodOn, we observed that the precision of the ontology learning increases as the number of seed food instances per class (n_{seed}) increases (**Supplementary Figure 1**), as a class is better represented as the number of seed instances increases. For $sim()$, we used Euclidean distance and cosine similarity, with the latter having better performance and being used throughout this work (**Supplementary Figure 2**). We empirically set $\alpha = 0.8$ after testing all values between 0.0 and 1.0 with an interval of 0.1 (**Supplementary Figure 3**).

2.2.3 Evaluation metrics of the ontology structure

The granularity and cohesiveness metrics have to do with fundamental design questions of ontologies such as the optimum number of classes and whether a class is overspecified or underspecified⁸⁷. *Granularity* is semantically defined as the ability to represent different levels of detail in data⁸⁸. In our work, we quantitatively define the granularity of a certain ontology superclass c_A as

$$granularity(c_A) = \frac{|I_{c_A}|}{|C^A|},$$

where $C^A \subseteq C$ is the set of all the classes that have c_A as their superclass, I_{c_A} is the set of all food instances belonging to C^A , and c_A is a superclass of c_B if every instance of c_A is also an instance of c_B ⁸⁹. *Cohesiveness* of a superclass is a measure of subclass semantic relevance, and by corollary, the degree its subclasses have the same relation to each other⁹⁰. Here, we quantitatively define the cohesiveness of a certain ontology superclass c_A as

$$cohesiveness(c_A) = \frac{|C'^A|}{|C^A|},$$

where C'^A is the set of all correct subclasses within the superclass c_A . For example, in the superclass “cheese food product by the organism” in FoodOn, the subclasses “cow cheese”, “goat cheese”, “sheep cheese”, and “buffalo milk cheese” are correct, while the subclass “blue cheese” is not, since it describes a method/process and not the point of origin. In this case, the cohesiveness value would be $4/5 = 0.8$. Another example is in the case of the bean superclass where the subclasses that are bean varieties are correct and subclasses for processed forms of beans such as ‘bean flour’ are not. The cohesiveness of the cheese superclass is 0.52 implying that only half the subclasses are correct and the bean superclass has a higher cohesiveness of 0.93.

2.2.4 Success metric of ontology population

We use precision to assess the performance of the ontology population and define it as follows:

$$precision = \frac{TP}{(TP + FP)},$$

where a food instance $i \in I_c$ is considered a *TP* if and only if the mapping function correctly placed i under c , and *FP* otherwise. In addition, we define path distance to be the shortest distance (hops) between the predicted class and the ground truth class, where a perfect ontology population algorithm would have a path distance of 0.

2.3 Results

2.3.1 Structural topology of FoodOn

Figure 2.2a provides a visualization of the ontology structure for the 2,764 classes in FoodOn. At the highest level of the ontology, every food is described by various features, which minimally includes its source organism and up to 11 other features, with each feature represented as a class (processes and material quality, among others; **Figure 2.2b**). A complete examination indicates that the ontology structure is heterogeneous in its granularity, with some classes having many subclasses and interconnectivity, while others have only one subclass. In a similar trend, while some classes have hundreds of instances, other classes have only one (**Figure 2.2c,d**). **Figure 2.2e** illustrates the variation in ontology depth for a given class, which is defined as the number of intermediate classes present in a given path that connects it to the root⁹¹. Considering all factors mentioned above, the FoodOn ontology is highly granular with an average of 3.15 food instances per class.

2.3.2 Granularity and cohesiveness impair the precision of automated methods

We trained word embeddings for the 13,629 instance and class labels in FoodOn to use in our method. These embeddings capture latent information about the food type as revealed by dimensionality reduction⁹² and subsequent analysis (**Figure 2.3a**). Regarding the structure of FoodOn, the granularity differs substantially as shown in **Figure 2.3b-e**, where we compare the superclasses ‘wine’ and ‘beans’, with a granularity of 5.64 vs 1.96, respectively. We also noticed inconsistencies in the further classification of each superclass which we quantify by the cohesiveness. Relevant to our work of ontology learning, we found that both the cohesiveness and the granularity are positively associated with better ontology population performance (PCC of 0.56 and 0.51, respectively; p -value = 2.5×10^{-2} and 4.5×10^{-2} , respectively) (**Supplementary Figure 4**).

2.3.3 Learning ontology via embeddings leads to substantially better performance

We kept the ontological structure of FoodOn unchanged with 2,433 target ontology classes and created 100 different seeded-skeleton ontologies to test the statistical significance of the methods by selecting two random seeds for each target class. This process resulted in 3,124 food instances used as seeds from a total of 10,865 instances, and the task was to map the remaining 7,741 food instances to the target classes (**Figure 2.4a**). The LOVE-generated ontology, which uses the word embeddings of size 300d trained using the Wikipedia corpus, had a significantly reduced path distance from what is expected from random chance (p -value = 4.8×10^{-102} ; **Figure 2.4b**). Moreover, ontology population methods based on the word embeddings performed better when compared to the traditional text similarity methods regardless of the embedding size or the training algorithm, with an 89.7% increased precision (0.34 vs. 0.18, respectively, p -value =

2.6×10^{-138}) and a 43.6% shorter path distance (2.91 vs. 5.16, respectively, p -value= 4.7×10^{-84} ; **Figure 2.4c** and **Table 2.1**).

2.4 Discussion

As shown in **Figure 2.3a**, there is an alignment of the word embeddings and the FoodOn classes at a high level. However, through deeper analysis of the ontology structure and the results of the automated ontology learning, we discovered the causes for discrepancies between the user-defined ontology and ontology representation from the corpus. The granularity and cohesiveness issues impacting the precision, have to do with a well-known and fundamental design question of how many classes are too few or too many⁸⁹. The classes with lower than average granularity of 4, combine several features of a food like its source, process, and organoleptic quality. However, the nomenclature is not consistent, as it varies from a long and precise class name to less precise representations. This is not a scalable approach to a data-driven automated ontology since it will require manually curated classes when mapping foods of yet unknown features like sources and processes. Moreover, it will lead to errors in mapping class-class and class-instance relations if done manually, as the ontology grows. To avoid these issues, an extension would be for every variety-specific subclass to contain a flat list of instances. For example, in **Figure 2.3e** the food instance 'adzuki bean flour' is mapped to two parent classes, in the bean superclass. Instead, the 'product by process' class at a depth of one, can have a subclass of 'milled food' which aggregates all the flour variants and notably the 'bean flour' class. This also addresses the problem of cohesiveness

described in **Methods**. The ontology learning function can then be applied to each of the 12 highest parent classes (**Figure 2.2b**).

Taking into account the structural similarity between ontology and knowledge graph, we considered applying observable and latent feature-based link prediction models^{65,93,94} to populate the ontology. However, such models either are dependent on external data or require at least one pre-existing path connecting the candidate instance to the target class. A possible extension to our work is to train the word embeddings using other related corpora such as food-related literature and databases, for example, the FDC database³². Moreover, the pertinent information can be extended to chemical composition, phenotypic effects, and association with health states. Another natural extension would be to train methods that encode the hierarchical structure of the knowledge graphs, such as Poincaré embeddings⁷⁸, with hierarchical food domain data⁹⁵ for the ontology population task. Along with an optimally designed skeleton ontology, we expect that these improvements would lead to much-improved accuracy of the automatically generated ontology.

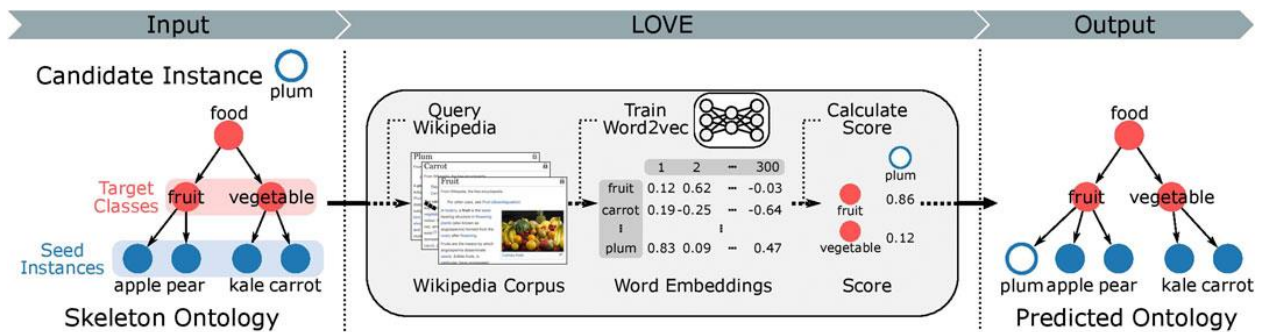


Figure 2.1. Overview of the LOVE ontology population framework. The hierarchical structure of the ontology is organized as a directed acyclic graph, where a class connects to its parent classes through directed edges. The target class is the parent class of the food instances. Note that some classes are part of the hierarchical ontological structure and do not contain any instances. All class and instance labels are used to query the Wikipedia corpus, which is then used to train food word embeddings. The mapping function then uses the word embeddings to map the candidate instances to the target classes. All relations between the instances and classes are of type ‘is a’. We compare the predicted ontology to the ground truth ontology and report the performance using precision.

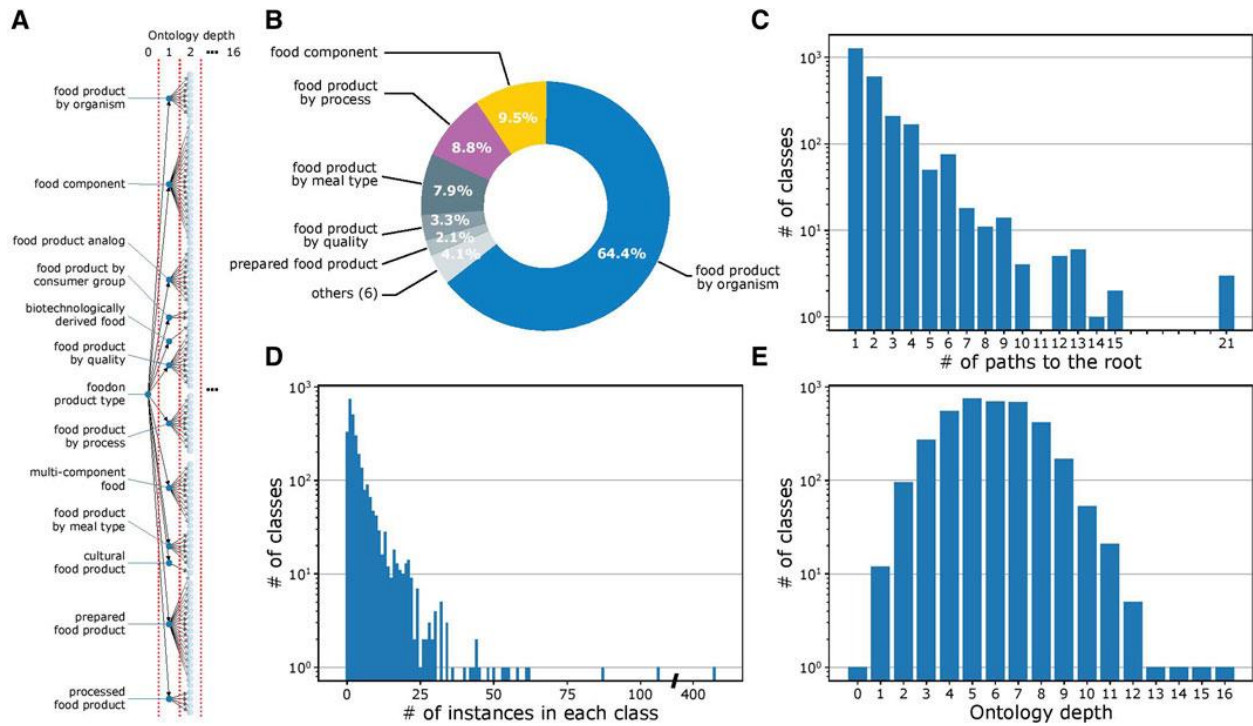


Figure 2.2. FoodOn structure analysis. **a** Visualization of the 2,764 classes in FoodOn. The 10,865 food instances that are mapped to the classes are not shown in the visualization. All classes in FoodOn branch out from a single root class 'foodon product type' located at depth 0. **b** Pie-chart showing the proportion of subclasses for each of the 12 classes in the highest, *i.e.* the first, level of the ontology. Each of these classes represents one of the 12 features of a food. **c** Histogram showing the number of paths to the root class for each of the 2,764 classes. The number of paths considers the multi-parent architecture. **d** Histogram showing the number of instances in each class. Only 2,433 of 2,764 classes have instances. Certain classes only have subclasses aimed at providing further levels of differentiation. The vast range in instances per class indicates that specialized classes with fewer instances are more typical to the ontology, though there are some classes with up to 100s of instances. **e** Histogram showing the number of the target classes at the respective ontology depth. This representation defines the ontology depth of class as the number of intermediate classes in a path connecting it to the root class, and it varies from 1 to 16 for FoodOn.

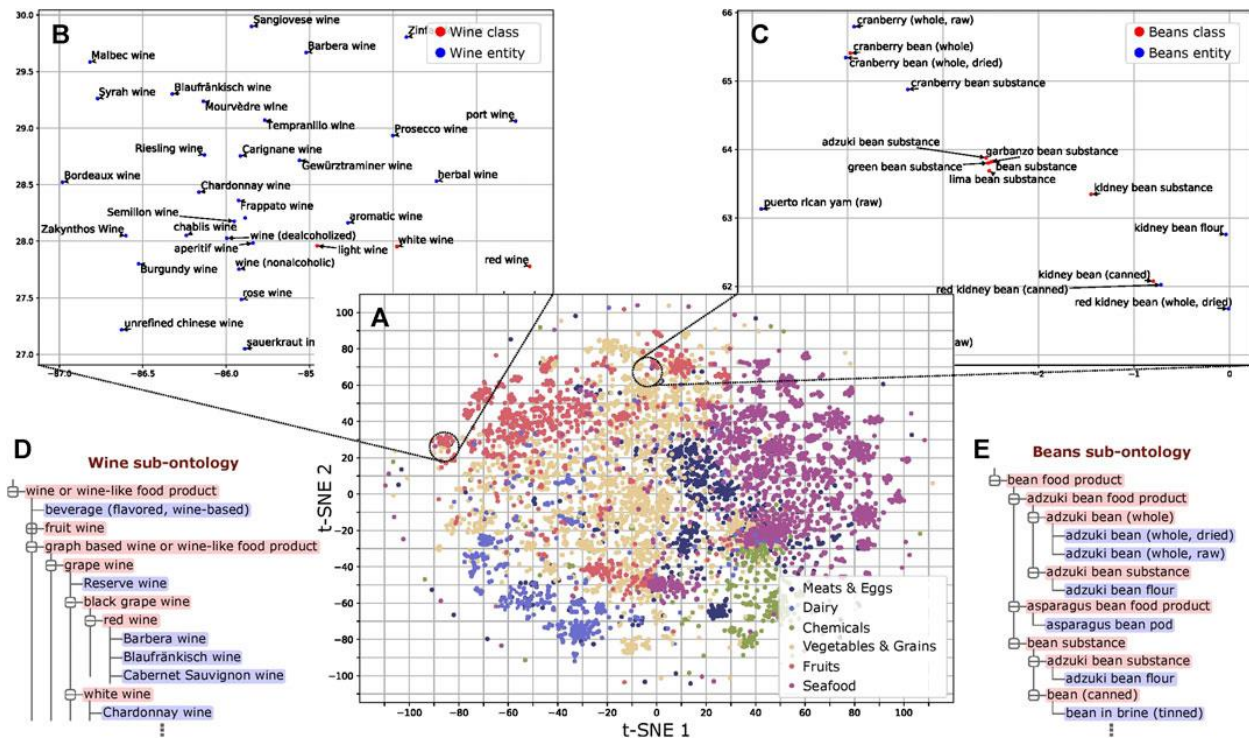


Figure 2.3. Analysis of word embeddings. **a** t-SNE plot of the FoodOn class and instance labels wine based on the word embeddings. The distribution pattern of the embeddings shows an ordering consistent with that of the FoodOn hierarchy (p -value < 0.0001). **b** Wine subsection follows the uniform spatial distribution of instances and classes. **c** Bean subsection shows regional crowding of instances/classes due to the repetitive wine words in the label. **d,e** Wine and Bean-related ontologies, with the bean being significantly more granular (more classes) than expected. Classes and food instances are highlighted in red and blue, respectively.

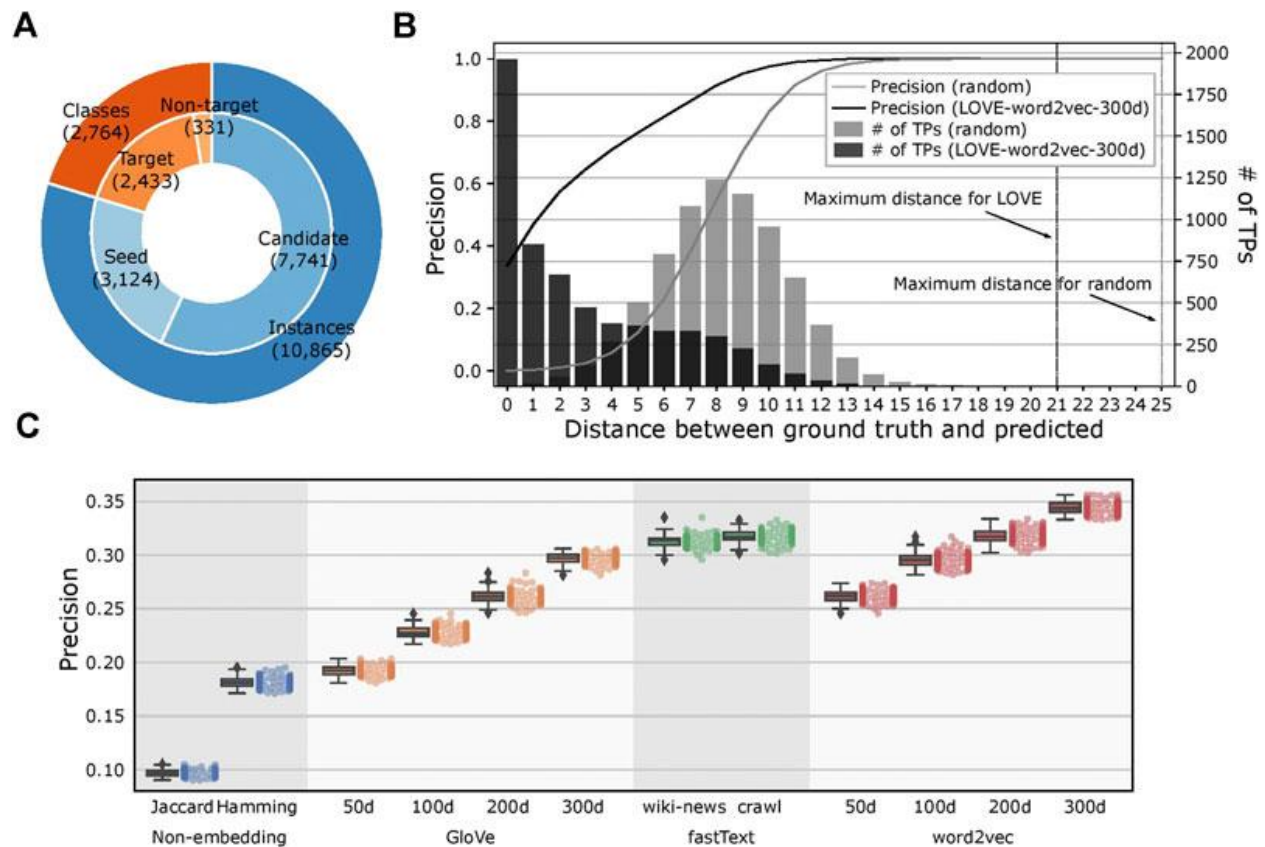


Figure 2.4. Evaluation of the LOVE framework on a food ontology. **a** The number of ontology classes and food instances that were used for the LOVE-derived ontologies. Candidate instances are mapped to one of the target classes by LOVE, and each target class is initialized by seed instances. Classes without instances are not considered target classes. **b** Distribution of precision and number of true positives of the mapped ontology as a function of shortest distance (hops) between the predicted class and the ground truth class, for LOVE (black) and random assignment (grey) (p -value = 4.8×10^{-102}). **c** Precision of ontology population for different similarity methods.

Table 2.1. Comparison of word similarity methods and their performance. Average distance denotes the average path distance between the predicted class and the ground truth class among 100 multiple randomly selected seed instances. All methods were run parallel on 8 core CPU (16 threads) with 32GB of memory. Note that running time excludes the time used for training the word embeddings.

Method		Training Corpus	Average Precision	Average Distance (hops)	Running time (s)
-	Random	n/a	4.6×10^{-4}	8.23	65.1
Non-embedding	Jaccard ⁹⁶	n/a	0.097	6.45	169.9
	Hamming ⁹⁷		0.181	5.16	111.7
GloVe⁸⁴	50d	Wikipedia 2014 + Gigaword 5	0.192	4.29	201.8
	100d		0.228	3.87	
	200d		0.261	3.53	
	300d		0.297	3.32	
fastText^{85,86}	Wiki-news	Wikipedia 2017 + UMBC webbase + statmt.org	0.313	2.98	
	Crawl	Common Crawl	0.317	3.00	
Word2Vec⁸³	50d	Subset of Wikipedia 2020	0.262	3.32	
	100d		0.295	3.09	
	200d		0.318	2.99	
	300d		0.344	2.91	

Chapter 3

Knowledge integration and decision support for accelerated discovery of antibiotic resistance genes

Disclaimer. All the work that is presented in this chapter has been published in *Nature Communications*⁹⁸.

3.1 Introduction

The pace of knowledge discovery within the life sciences domain has long been hindered by the intricate and multifaceted nature of biological systems. The complexity of biological processes, coupled with the vastness of available data, has resulted in a painstakingly slow and often costly process of unraveling new insights. Traditional methods of hypothesis generation and experimentation can be time-consuming, resource-intensive, and may not efficiently explore the expansive landscape of potential relationships within biological data^{99,100}. In this context, the integration of advanced machine learning models for hypothesis generation emerges as a transformative solution. These models^{101–103}, equipped with the ability to analyze large-scale datasets, discern intricate patterns, and make data-driven predictions, offer a means to expedite the discovery of novel biological knowledge. By leveraging these sophisticated tools, researchers can navigate the complexities of the life sciences domain more efficiently, accelerating the pace of

discovery and potentially mitigating the associated costs, thereby ushering in a new era of precision and innovation in biological research.

For computational methods to be effective, the integration and ingestion of biological data at scale are paramount^{102,104,105}. To this end, various initiatives^{106–108} have transitioned from relational databases that store data using tables and are often limited by their scalability⁸ to graph databases that efficiently process dense interrelated datasets⁹ by utilizing the Resource Description Framework (RDF) triplet of subject, predicate, and object¹⁵. This design helps to identify patterns among data, and to utilize the information content they carry to gain insights into the mechanisms of action, associations, and testable hypotheses^{105,109}. In the biomedical domain, knowledge graphs¹⁷ with thousands to millions of RDF triplets are used to organize knowledge in life sciences^{24,110}, including health conditions such as cancer¹¹¹ and cardiovascular disease¹¹². In the case of antibiotic resistance genes (ARGs), there exist both graph databases like CARD¹¹³ and ARDB¹¹⁴ that represent ontologies, as well as traditional databases like MEGARes¹¹⁵, ARGO¹¹⁶, and ARG-ANNOT¹⁰⁰ that store ARG sequencing data. Current challenges include unreported or unresolved conflicted information between two or more sources^{58,59}, lack of negative findings^{117,118} that is necessary to train machine learning models, focus on only one relation type¹¹⁹, inability to directly integrate results across sources due to incompatible meta-data¹²⁰, all of which limits their suitability as a training set for machine learning models. Similarly, extracting training data from published literature is challenging as it is often hidden in supplementary tables and figures^{121,122}, may be inaccessible or incompatible^{123,124}, which hinders any knowledge synthesis and analysis^{125,126}.

Automating the integration of heterogeneous biomedical data and their organization so they are machine learning-ready for downstream analysis and knowledge discovery is important for any life science field. One such area is the discovery of ARGs and relationships. Antibiotic resistance poses a major threat to the efficacy of antibacterial drugs, which leads to increased mortality and costs¹²⁷. Identification of ARGs has traditionally been performed through time-consuming and expensive culture-based methods¹²⁸ and more recently through bioinformatics analysis of whole-genome sequencing samples, including BLAST-based^{100,129} and deep learning-based^{130,131} methods. Outside of the domain of antibiotic resistance, there have been multiple attempts to discover biological knowledge from knowledge graphs^{132–136} by formulating it as a knowledge graph completion (KGC)¹³⁷ problem, where the objective is to complete (discover) the missing links (new knowledge) in the graph. Graph feature models^{65,138} and latent feature models^{139,140} have traditionally been used for KGC, whereas models that utilize pre-trained language models^{141,142} have recently achieved state-of-the-art results.

In this study, we present a methodology (Knowledge Integration and Decision Support, or KIDS) that constructs an inconsistency-free knowledge graph that supports multiple triplet types and can be used to generate hypotheses over multiple iterations (**Figure 3.1**). We apply the KIDS framework to the area of *Escherichia coli* antibiotic resistance, which leads to a knowledge graph consisting of 651,758 triplets of 23 RDF triplet types in total, among which 9 triplet types are negative. To resolve inconsistencies, we computationally predicted, and experimentally validated 236 sets of inconsistencies with 94.07% accuracy. We then demonstrate how the automated process allows the discovery of previously unknown ARGs. KIDS achieved an average of 0.77 AUCPR and 0.86 AUROC in

predicting the ARGs over two iterations of hypothesis generation, validation, and integration with existing knowledge, with the predicted ARG probability being highly correlated with validated findings ($R^2=0.94$). Furthermore, our analysis led to the discovery of 6 ARGs that we have validated experimentally, among which 5 homologs in *Salmonella enterica* also showed antibiotic resistance.

3.2 Methods

3.2.1 Knowledge graph constructor.

The knowledge graph construction process is shown in **Supplementary Figure 6** with detailed examples.

Data integration. We merge distinctive sets of knowledge from 10 different sources (**Appendix A.1.1.1**) in a unified format using binary relationships known as an RDF triplet of the form (subject, predicate, object), where subject and object are the nodes (biological entities) in the graph, and the predicate is the edge (relation) between them.

Synonym resolution. For entity types gene and antibiotic in the integrated knowledge graph, a name mapping table is applied to resolve the synonyms as multiple representations may exist for a single entity. For gene name mapping, Accession IDs to external databases and synonym lists of all *E. coli* genes downloaded from EcoCyc¹⁴³ are mapped to the original gene symbol. For all antibiotics, we map all synonyms listed in ChemIDplus¹⁴⁴ to their MeSH name (defined as MeSH heading in ChemIDplus).

Knowledge inference. As a data augmentation step, we added 15 sets of rules that we manually created to bridge existing gaps in the knowledge representation. As an example, a new triplet (*sucD*, has, response to antibiotic) can be inferred from an existing triplet (*sucD*, confers resistance to antibiotic, Cephadrine).

3.2.2 Inconsistency resolver.

The inconsistency resolution process is shown in **Supplementary Figure 6** with detailed examples.

Inconsistency detection. To detect inconsistencies in the knowledge graph, we manually defined 9 sets of rules upon close inspection of the knowledge graph. In this work, we treat a set of triplets that share the same subject and object entities connected by conflicting predicates as an inconsistency. For example, triplets (*atpA*, confers resistance to antibiotic after 18 hours, Ampicillin) and (*atpA*, confers no resistance to antibiotic after 18 hours, Ampicillin) are considered one set of inconsistency.

Inconsistency resolution. Let t be a triplet and s be a source, then $t \in T$ and $s \in S$, where T and S are the groups of all triplets and all sources, respectively. If we let T_s be all the triplets of source s , then $T = \bigcup_{s \in S} T_s$. Each triplet $t \in T$ belongs to a mutual exclusion set $M_t \subseteq T$, a set of triplets that are mutually exclusive with one another. In an inconsistency-free setting, a triplet t belongs to one unique M_t . In other words, $|M_t| = 1$ means there exist no conflicts in M_t . Assuming there exists one true triplet \bar{t} in each mutual exclusion set M , the goal of the inconsistency correction methods is to predict \bar{t} for all M with $1 < |M|$. Prediction of \bar{t} is done by measuring the belief of triplet t , $B(t)$ (i.e., the level of confidence that triplet t is true), among all t in M and by assigning t with the

highest belief $\operatorname{argmax}_{t \in M} B(t)$. Although the specific way to measure $B(t)$ varies across methods, it is commonly estimated based on the source trustworthiness $R(S_t)$ (i.e., level of trust assigned to the source), where $S_t = \{s : s \in S, t \in T_s\}$ is the set of sources with t . We compute the trustworthiness $R(s)$ and belief $B(t)$ iteratively until convergence. We used the AverageLog¹⁴⁵ among others (**Appendix A.1.2.1**), and the equations to update $R^i(s)$ and $B^i(t)$ for each iteration i are as follows:

$$R^i(s) = \log |T_s| \frac{\sum_{t \in T_s} B^{i-1}(t)}{|T_s|}, \quad (1)$$

$$B^i(t) = \sum_{s \in S_t} R^i(s), \quad (2)$$

where $R^i(s)$ and $B^i(t)$ are normalized to prevent a numerical explosion by dividing with $\max_{s \in S} R^i(s)$ and $\max_{t \in T} B^i(t)$, respectively. $B^0(t)$ is set to 0.5 for all $t \in T$. Performance evaluation of alternative inconsistency resolution methods can be found in **Supplementary Figure 10** through **Supplementary Figure 15**, **Supplementary Table 7**, and **Supplementary Table 8**.

3.2.3 Hypothesis generator.

Preprocessing. There was not enough training data to train the hypothesis generator if we were to treat each predicate of varying temporal information distinctly. Therefore, we ultimately decided to modify the knowledge graph by removing the temporal information from the 14 predicates (e.g., ‘CRA after 15 hours’ to CRA). This process reduced the size

of the knowledge graph by 24.1% from 651,758 triplets to 494,819 triplets and the number of predicates from 23 to 12 (**Supplementary Table 4**).

Path Ranking Algorithm (PRA)^{65,146}. The set of paired entities from the training set, linked by the CRA predicate, is first used to identify the paths used to train the model. This is done by initiating a random walk at a bounded step size starting at the subject entity. If the random walk ends up at the object entity, this path is considered successful. To reduce the size of the feature space, a path will only be considered if it links at least one object entity. Additionally, the object entity found by a path must be supported by at least a fraction, α , of the training samples. Finally, L1-regularization is used during logistic regression to reduce the feature space even more. Each path retained for the model is treated as a path feature. The value of each feature is the prior probability of reaching the object entity from the subject entity for the given sample. These path probabilities are computed recursively by assuming that every step of the path, an outgoing link to an entity, is chosen uniformly at random. After training a regularized logistic regression model to identify the parameters of these features, the final score to predict the existence of an edge in the graph is as follows:

$$score(s, o) = \sum_{P \in P_l} h_{s,P}(o) * \omega_P, \quad (3)$$

where s and o are the subject and object entities, P is one of the paths chosen by the model, P_l , $h_{s,P}(o)$ is the path probability, and ω_P is the weights determined using logistic regression. We set L1-regularization to 0.008, L2-regularization to 0.0001, and the fraction, α , to 0.01 based on a hyperparameter search performed on 5-fold cross-

validation. More details on computing these probabilities can be found in their original work.

Multilayer Perceptron (MLP). The MLP, a fully connected feed-forward artificial neural network, outputs a probability of whether a given triplet is true. Each entity and predicate of the knowledge graph is converted to a dense numerical vector of length 50, which is created by taking the average of the constituent word embeddings¹⁴⁷. These word embeddings are randomly initialized and treated as learnable parameters for the model. A dense numerical vector of length 150, which is created by concatenating the subject, predicate, and object embeddings, is fed as an input to the network. The network contains four hidden layers, each with 60 nodes. We used ReLU¹⁴⁸ activation functions until the third hidden layer, followed by a Tanh activation function for the last hidden layer. Finally, the output layer uses the sigmoid activation function to produce a score between 0 and 1. We used dropout¹⁴⁹ after all but the last hidden layer to reduce overfitting. The model was trained to leverage the margin-based ranking loss¹³⁷:

$$l(\omega) = \sum_{i=1}^N \sum_{c=1}^C \max(0, \gamma - g(T^i) + g(T_c^i)) + \lambda \|\omega\|_2^2, \quad (4)$$

where N is the number of training edges, C is the corruption size, function $g()$ represents a complete forward pass of the network or scoring function on a given edge T , ω is the weights of the model, λ is the L2-regularization parameter, and γ is the margin that the correct edge must score higher than the corrupted edge. Based on a hyperparameter search performed on 5-fold cross-validation, we used Adam¹⁵⁰ optimization with a

learning rate of 0.001, λ was set to 0.001, the dropout rate was set to 0.5, C was set to 100, and the margin used for training was set to 0.20.

Stacked. We stacked the two models PRA and MLP using AdaBoosted¹⁵¹ decision stumps, in line with¹⁵². The training inputs to the model were three features: the scores produced by the PRA and the MLP and one binary value for the PRA to indicate whether the PRA was able to predict that certain sample. Note that the PRA cannot predict if no paths were found. We performed random search hyperparameter optimization over the validation set for each fold and found optimal parameters of 680 estimators and a learning rate of 1.65. Since our dataset is unbalanced, we also used Smote¹⁵³ sampling to synthetically create positive samples for a balanced set of positive and negative samples.

3.2.4 Wet-lab validation.

To validate whether a gene confers resistance or not, wild-type Keio strain BW25113 and its derivative single-gene knockout (KO) strains were used¹⁵⁴. MIC values of the following antibiotics were measured: Amoxicillin (Sigma, Cat# A8523), Ampicillin (Roche Diagnostics, Cat# 10835269001), Apramycin (Alfa Aesar, Cat# AAJ6661603), Cephadrine (Alfa Aesar, Cat# AAJ664960), Chloramphenicol (Calbiochem, Cat# 220551), Geneticin (Teknova, Cat# 50841719), Hygromycin B (Calbiochem, Cat# 400050100MG), Kanamycin (Acros Organics, Cat# AC611290050), Levofloxacin (Chem-Impex, Cat# 50508743), Norfloxacin (Sigma, Cat# SIAL-N9890), Novobiocin (Calbiochem, Cat# 491207), Oxycarboxine (Sigma, Cat# 36185), Paromomycin (Chem-Impex, Cat# 501602750), Rifampin (Alfa Aesar, Cat# AAJ6083603), Sisomicin (TCI, Cat# I1049250MG), Spectinomycin (RPI, Cat# 50213656), Streptomycin (Across Organics,

Cat# AC455340050), Sulfanilamide (Alfa Aesar, Cat# AAA1300122), Triclosan (Cayman Chemical Company, Cat# 501599771), Troleandomycin (Enzo Life Sciences, Cat# BML-EI249-0010), and Vancomycin (VWR Life Science, Cat# 97062-554). Since KO strains had a kanamycin resistance gene, the kanamycin resistance gene was removed from the required KO strains¹⁵⁵ to measure the resistance in kanamycin. Antibiotics and strains were preserved at -80°C until used.

1 µL of the required preserved strain was inoculated in 200 µL LB broth and grown overnight in an incubator shaker (BioTek Synergy HTX) at 37°C. ~3 µL of grown culture was transferred, using a replicator, to LB agar plates containing different amounts of antibiotics, and plates were incubated overnight (~18 hours) at 37°C in an incubator. The next day, the absence and presence of colonies were monitored. The minimum concentration of antibiotic, at which no colonies were observed, was defined as minimum inhibitory concentration (MIC). In the case of metronidazole, colonies were observed at all concentrations. Metronidazole is a pro-drug and inactive but in anaerobic conditions, this is converted to an active form by the bacteria^{156,157}. The active form is toxic which leads to the killing of bacteria. As our experimental conditions were aerobic, metronidazole was converted to an active form, and we observed colonies at all concentrations. Subsequently, we removed metronidazole from our study.

3.3 Results

3.3.1 The landscape of *E. coli* antibiotic resistance genes and processes.

We applied the KIDS framework to the biological domain of *E. coli* and constructed a multi-relational knowledge graph¹⁵⁸ (see **Methods**) that consists of 651,758 triplets (**Figure 3.2a,b**). Raw data to construct the knowledge graph were curated from a total of 10 sources (**Appendix A.1.1.1**) that include information about antibiotic resistance, effects of antibiotics on the expression patterns, gene-regulatory relations with transcription factors, and the impact of genes on the biology of an organism at the molecular, cellular, and organism levels¹⁵⁹, all regarding *E. coli* genes (**Figure 3.2c**). The resulting knowledge graph provides a comprehensive view of the positive *E. coli* antibiotic resistance with 18-fold more genes and 3-fold more antibiotics than CARD¹¹³ (**Figure 3.2d, Supplementary Table 1**). Among the 23 triplet types of the knowledge graph, 14 positive triplet types account for the 31,216 (4.8%) associations as genes are less likely to confer resistance to an antibiotic (**Figure 3.2e**). The knowledge graph contains antibiotic exposure times at 6 different time points ranging from 30 minutes to 7 days (**Supplementary Table 2**). From the total of 466,752 possible gene-antibiotic pairs, 358,674 pairs (76.9%) were connected via either a positive or negative ‘confers resistance to antibiotic (CRA)’ predicate, with the rest being candidates for either association (**Supplementary Figure 5**).

3.3.2 Resolved inconsistencies help discover new knowledge.

We identified 236 sets of inconsistencies in our intermediate knowledge graph (**Figure 3.3a**) between the findings of two sources Tamae et al.⁵⁸ and Liu et al.¹⁶⁰ for positive and negative counterparts of the predicate ‘CRA after 18 hours’ despite their identical experimental setup (**Supplementary Table 6**). We then applied the AverageLog¹⁴⁵

inconsistency resolution algorithm (see **Methods**) to select which one of the two conflicting facts is more likely to be true by iteratively updating the source trustworthiness and belief of the triplet (**Figure 3.3b**). Results show that we were able to accurately resolve these inconsistencies (94.07% accuracy, 50.0% F1-score, 33.3% precision, 3.0% baseline precision) when compared to the ground truth wet-lab validation (**Figure 3.3b**, **Supplementary Table 3**), which was performed by measuring and comparing the minimum inhibitory concentrations (MICs) of the single-gene knock-out strain and the wild-type strain on the LB agar plate (see **Methods**). We then trained the hypothesis generator before and after resolving inconsistencies, to test how inconsistency resolution affects knowledge discovery. This led to two previously unidentified antibiotic-resistant relationships (*surA*, CRA, Vancomycin) and (*asmA*, CRA, Vancomycin) with significantly increased probabilities after the inconsistency resolution (0.024 to 0.882 and 0.005 to 0.213, respectively) that we validated experimentally.

3.3.3 KIDS accelerates knowledge discovery.

The hypothesis generator module performs link prediction¹³⁷ on the incomplete knowledge graph to identify the missing links (i.e., generate hypotheses). We focused on exploring the missing CRA links between all pairwise combinations of *E. coli* genes and antibiotics (108,078 hypotheses). To this end, we applied five different variations of the hypothesis generation methods (PRA^{65,146}, MLP, a stacked model that combines PRA and MLP using AdaBoost¹⁶¹, TransE¹³⁹, and TransD¹⁶²; see **Figure 3.4b** and **Methods**) on a reduced knowledge graph without temporal information (see **Methods**) that has 494,819 triplets and 12 predicate types (**Supplementary Table 4**). From those methods,

PRA^{65,146} finds observable predicate paths between subject (source) and object (target) nodes in the graph and treats them as human-interpretable features (**Supplementary Table 9**). In contrast, MLP¹⁵² is a fully connected neural network that uses the triplets represented by latent vector embeddings to predict whether any given edge is valid. We also tested translation-based graph embedding methods TransE¹³⁹ and TransD¹⁶² (**Appendix A.1.3.5**), but we selected the stacked model as it had superior performance in testing. Evaluation of these methods, which have been optimized for F1-score, using 5-fold cross-validation shows that the stacked model had the best performance in terms of AUCPR with a 154.4% increase when compared to PRA (0.28 vs. 0.11, respectively, p -value = 2.1×10^{-6}) and a 27.7% increase when compared to MLP (0.28 vs. 0.22, respectively, p -value = 3.0×10^{-3}) (**Figure 3.4c, Supplementary Table 5**), while the baseline was 0.02.

We used the stacked model to generate 226 CRA hypotheses of varying probability that we subsequently tested experimentally. Of those hypotheses, 64 (28.3%) were validated as positives (**Figure 3.5a**). After adding those results to the knowledge graph, we ran a second iteration of KIDS, which produced another 90 hypotheses, from which 29 (28.8%) were positively validated (**Figure 3.5a**). From these two iterations, we computationally predicted and experimentally validated, similar to the wet-lab validation performed for the inconsistency resolver (**Appendix A.1.3.12**), a total of 93 CRA hypotheses for 83 *E. coli* genes that confer resistance to one or more of 15 antibiotics (**Figure 3.5e**). The KIDS-generated hypotheses are reliable as the calibrated output for each hypothesis is a highly correlated confidence score ($R^2=0.94$) with the validated outcome (**Figure 3.5a**). For instance, hypotheses with probability >0.8 have a high true positive rate with 29 out of 37

tested hypotheses (78.4%) yielding an ARG, whereas hypotheses with probability ≤ 0.2 have a true positive rate with only 14 out of 163 tested hypotheses (8.59%) to yield an ARG. Interestingly, KIDS produced improved hypotheses in the second iteration with the addition of the newly discovered results (**Figure 3.5b~d**). The KIDS-generated hypotheses are positively correlated with high consistency when compared to the random baseline (Kendall's tau¹⁶³ = 0.96 vs. 0.00, respectively, p -value $< 2.2 \times 10^{-308}$; RBO¹⁶⁴ = 0.56 vs. 0.00, respectively, p -value $< 2.2 \times 10^{-308}$; **Appendix A.1.3.11**).

3.3.4 AI-driven discovery of 6 antibiotic resistance genes.

An extensive literature search on the 83 *E. coli* genes that are implicated in the CRA hypotheses, identified 15 genes that are previously unknown ARG for *E. coli*, with 6 of them (1 from the first iteration and 5 from the second iteration) not appearing as an ARG for any bacteria. Those 6 are the following: *ftsP*, *hdfR*, *lrp*, *proV*, *qorB*, and *rbsK* (**Figure 3.6**), which have never been reported to be involved in antibiotic resistance. Further investigation of the biological processes reveals they are part of a diverse repertoire of functions related to amino acid metabolism, nutrient transport, and regulation. More specifically, FtsP is a cell division protein that is required for bacterial growth during stress conditions. FtsP stabilizes or protects the divisional assembly during stress condition¹⁶⁵. HdfR, which is an H-NS-dependent *flhDC* regulator, represses the expression of the flagellar master operon *flhDC*¹⁶⁶ and induces the expression of the *gltBD* operon, which is involved in acid resistance^{167,168}. Lrp encodes a leucine-responsive regulatory protein, which regulates at least 10% of the genes in *E. coli*, including regulation of major porins OmpC and OmpF that determine the permeability of the cell membrane^{169,170}. ProV is

predicted to be a component of an osmoresponsive ABC transport system and involved in osmosensing¹⁷¹. QorB is a NAD(P)H:quinone oxidoreductase, which catalyzes the reduction of quinone. *E. coli* strain overexpressing *qorB* shows defects in growth and a significant decrease in several enzymes involved in carbon metabolism¹⁷². Interestingly, oxidoreductases have been reported to involve antibiotic resistance¹⁷³. RbsK is a sugar kinase that, in addition to phosphorylation of ribose, facilitates stress-induced mutagenesis in *E. coli*¹⁷⁴. Mutations in sugar kinase genes such as *waaP* of *S. enterica* lead to increased susceptibility to antibiotic polymyxin^{175,176}.

We did not identify any statistically significant homologs (E-value < 0.05) of these 6 genes among the 4,577 ARGs from CARD¹⁷⁷, while the best hit was for OXA-541 of *Pseudomonas putida* for *Irp* (91.7% sequence similarity, E-value = 0.12) (**Appendix A.1.3.13**). The prevalence of these 6 genes across the human digestive microbiome ranges from 0.67% to 8.79% (**Appendix A.1.3.14, Supplementary Table 12**). Finally, to investigate the antibiotic resistivity of genes homologous to the 6 previously unknown ARGs in other bacterial genera, we identified 5 homologs *ftsP*, *Irp*, *proV*, *rbsK*, and *yifA* (*hdfR* in *E. coli*) in *S. enterica* with >78% similarity in nucleotide sequences, while the homolog of *qorB* was not identified (**Appendix A.1.3.15 and A.1.3.16**). Wet-lab validation revealed that knocking out these 5 genes in *S. enterica* also increased susceptibility to antibiotics.

3.4 Discussion

In this work, we presented the KIDS framework as an automated method for knowledge organization and discovery. We demonstrated the power of the KIDS platform in the context of *E. coli* antibiotic resistance, a research area with a need for such a method, as the emergence of antibiotic resistance renders existing antibacterial drugs less efficient and thus necessitates a constant race to discover new ways to fight microbial infections¹⁷⁸. Out of the 6 ARGs discovered in this work, we found that 5 homologs in *S. enterica* also conferred resistance to an antibiotic, indicating that the knowledge gained in this study can be easily translated to closely related bacteria. Current computational tools identify potential ARGs by genomic and metagenomic sequence analysis, which has limited performance when the reference database does not include similar ARG sequences. Similarly, just looking at homology is not sufficient for discovering ARGs. Among the 129 genes from the lowest probability range [0.0, 0.2] that we have validated to have no antibiotic resistance, we found 9 homologous ARGs in CARD that have significant E-value (< 0.05) with $>68.6\%$ sequence similarity. KIDS removes the dependency on reference sequences as its power stems from guilt-by-association and pattern discovery within the knowledge graph. Although manual literature curation and experimental validation were tedious and time-consuming, we found that the KIDS framework generates actionable hypotheses that lead to automated knowledge discovery with high confidence and efficiency.

On the summary statistics, the improvement from resolved inconsistencies was small, most likely because only 7 out of the 236 inconsistencies (3.0%) we experimentally resolved and further validated in the wet lab were positive triplets, and therefore reinstating them back to the knowledge graph where 1,606 positive CRA triplets exist did

not affect the knowledge graph much (1,606 to 1,613, a 0.44% increase). However, we found two previously unknown antibiotic-resistant relationships (*surA*, CRA, Vancomycin) and (*asmA*, CRA, Vancomycin) only after reinstating the resolved inconsistencies into the knowledge graph, something that demonstrates the importance of inconsistency resolution and coherence in our knowledge. For the lack of negative findings, our knowledge graph is the first to include both the positive findings (14 triplet types, 31,216 triplets) and the negative findings (9 triplet types, 620,542 triplets) to the best of our knowledge. Although the majority of the hypothesis generation models we tested did not use these negatives and instead generated them either through closed-world assumption or corruption through random sampling, our best model (stacked) did utilize these negatives. We believe there is still a potential to take advantage of these negative findings in other machine learning models. To address the focus on only one relation type, our knowledge graph contains 23 relation types (**Supplementary Table 2**) as opposed to a single relation type from other sources (**Appendix A.1.1.1**). Finally, regarding the inability to directly integrate results across sources due to incompatible metadata, this is still a problem for this and any other framework, as it is related to data incompatibility during their generation and reporting, something that we as a community need to collaboratively work on by adhering to standards like FAIR⁴⁰.

Although translation-based graph embedding models have shown state-of-the-art performance in some benchmark datasets^{179,180}, they performed worse than models like MLP and Stacked for our *E. coli* knowledge graph (**Supplementary Table 5**). This may be due to the known limitations of these methods where they are unable to handle knowledge graphs with complex and diverse entities and relations (e.g. one-to-many,

many-to-one, many-to-many)¹⁸¹ or do not utilize semantic information¹⁸². For example, in our knowledge graph, many genes are known to confer resistance to a specific antibiotic (many-to-one). Therefore, these genes will be close to each other in the embedding space, making it difficult to differentiate them from each other. This leaves room for performance improvement of the hypothesis generation methods by utilizing the current state-of-the-art link prediction methods^{141,142} which take advantage of pre-trained language models (LM) like BERT¹²² and RoBERTa¹⁸³ and approach the problem as a natural language processing task. Unlike graph embedding approaches^{139,184–187}, LM-based methods generalize well to unseen nodes or edges in graph¹⁴². However, the application of such methods on the biological domain remains a challenge as LM models are usually not trained on biological data, except BioBERT¹⁸⁸, in which case further fine-tuning of the LM model to the specific domain (*E. coli* ARG here) is desired. For the scope of this work, we used a stacked (MLP and PRA) hypothesis generation method, inspired by the Knowledge Vault¹⁵² project.

There are a few areas of improvement. First, knowledge inference rules (see **Methods**) were generated upon visual inspection of the 23 triplet types of the knowledge graph. There are automatic knowledge graph construction methods^{53,54} that can potentially do this automatically, but we leave it for future work as their precision is not at the human level nor has been tested in the biomedical domain. Second, although our knowledge graph contains temporal information, we discarded it when training the hypothesis generator. Allowing temporal features^{189–191}, we could expand our research to generate time-specific hypotheses, using techniques such as sequence-to-sequence learning methods^{192,193}. Third, the major bottleneck of the KIDS framework is its dependency on

expert-guided manual curation of data in RDF triplet format. An automated data curator would be a boon to adding information from existing literature^{53,194}. Additionally, we expect better initialization schemes, such as those based on pre-trained word embeddings trained using scientific literature instead of random initialization, to further improve performance⁸³⁻⁸⁵. Concomitantly, we would like to apply KIDS to other bacteria and replicate the success that we observe in *E. coli*. Finally, evaluating the impact of data size on learning performance can help to determine how well this method can generalize to other microbes with limited training data.

Taken together with other advances in optimal experimental design^{101,195}, interpretable machine learning^{196,197}, and automated research and development approaches¹⁹⁸, the proposed framework paves the way for a systematic, optimized, and reproducible way to elucidate complex biological systems in shorter timescales, with less manual labor, and unprecedented fidelity.

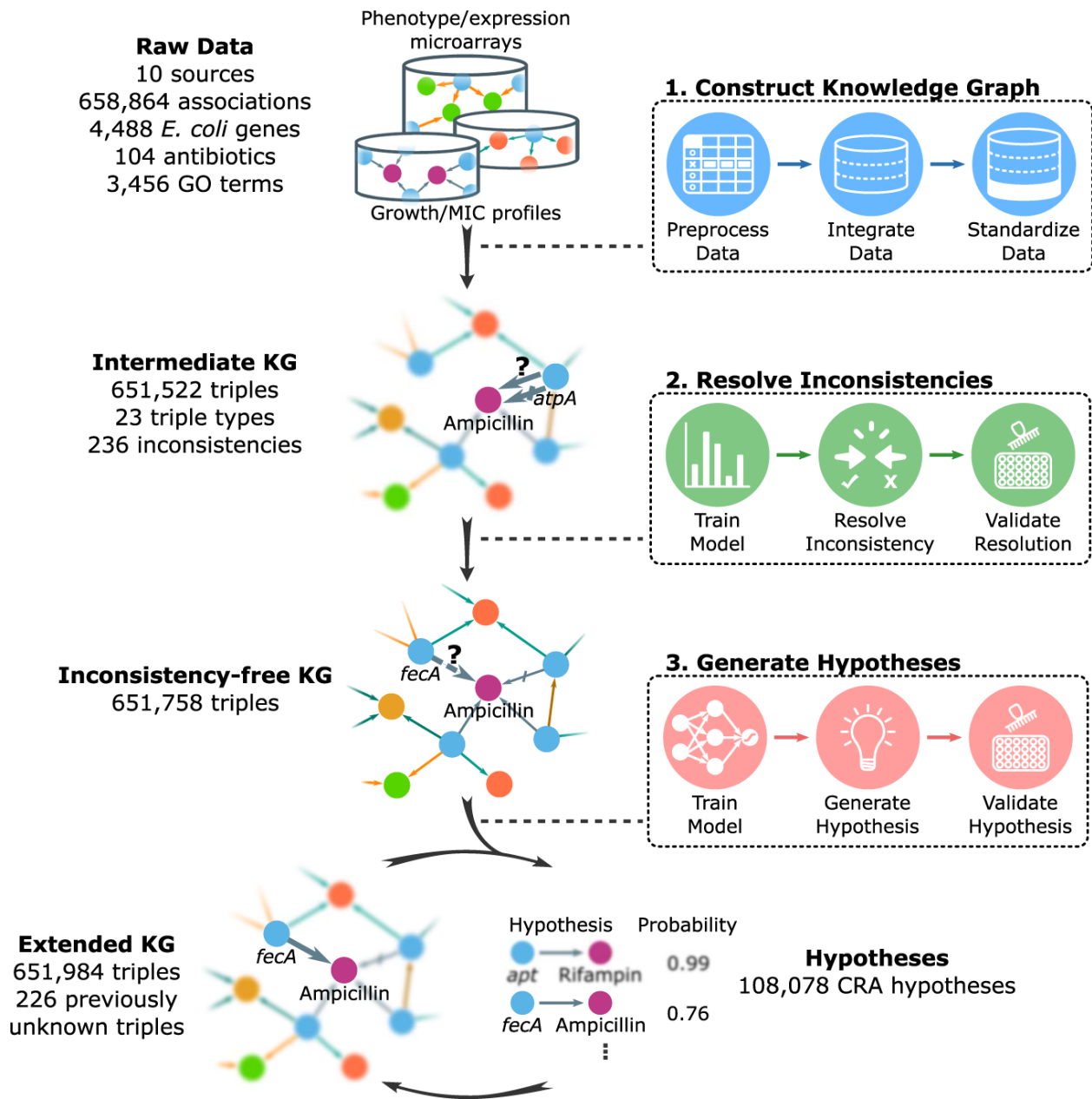


Figure 3.1. Overview of the KIDS framework. First, an intermediate knowledge graph is created from 10 sources by processing RDF triplets that encode 23 types of associations. Second, inconsistencies are computationally resolved and experimentally validated to construct an inconsistency-free knowledge graph. Third, a hypothesis generator is trained on the knowledge graph and assigns probabilities for the missing links. Hypotheses with high probability are experimentally validated, and the results are integrated into the knowledge graph, which is used for the next iteration of hypothesis

generation. GO refers to the Gene Ontology, and CRA (gray arrow) denotes a predicate 'confers resistance to antibiotic'.

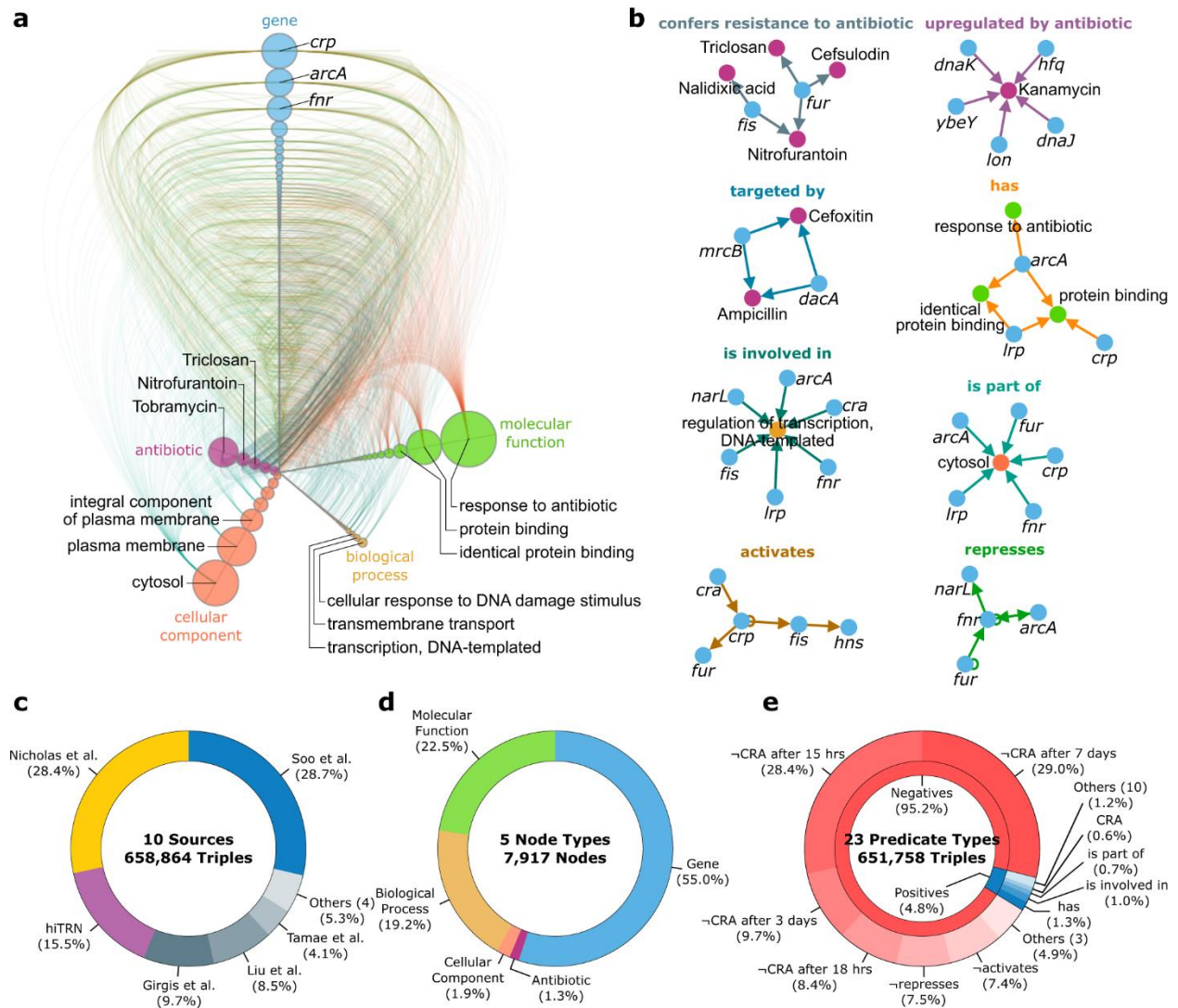


Figure 3.2. The inconsistency-free *E. coli* knowledge graph. **a** Hive plot visualization of the knowledge graph’s major components, with each axis corresponding to one of five different node types: gene, antibiotic, cellular component, biological process, and molecular function. The size of a node is its in and out degree. Only the 5% highest degree nodes from each node type and their positive connections are shown. **b** The top highest degree nodes for each of the 8 positive predicates in the knowledge graph. **c, d, e** Breakdown of the knowledge graph representation in terms of data sources, node, and predicate types. CRA denotes the predicate ‘confers resistance to antibiotic’, whereas \neg CRA denotes ‘confers no resistance to antibiotic’.

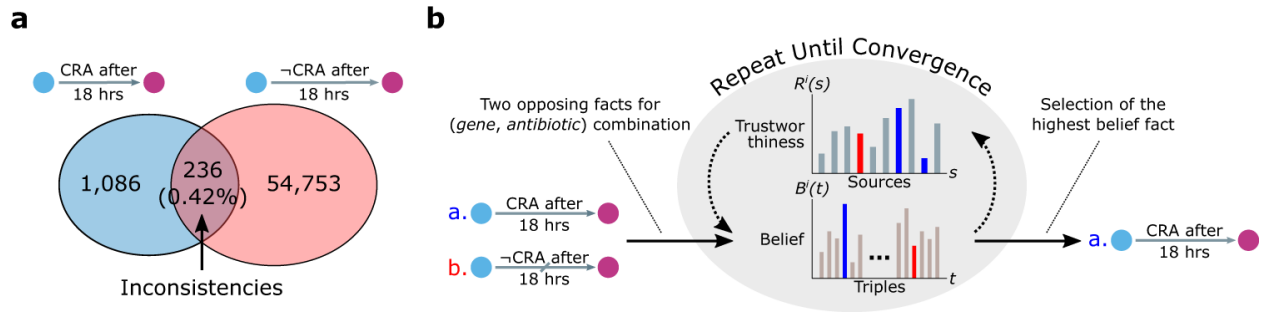


Figure 3.3. Inconsistency resolution. **a** Venn diagram showing the inconsistencies detected in the intermediate knowledge graph, where the inconsistency is defined as two or more sources supporting a conflicting fact. \neg CRA corresponds to a negative predicate ‘confers no resistance to antibiotic.’ **b** The inconsistency resolution algorithm is iteratively trained using the intermediate knowledge graph. Once the training converges, it is used to select the triplet with the higher belief among the inconsistencies. The blue and purple nodes represent genes and antibiotics, respectively. CRA denotes the predicate ‘confers resistance to antibiotic’, whereas \neg CRA denotes ‘confers no resistance to antibiotic’.

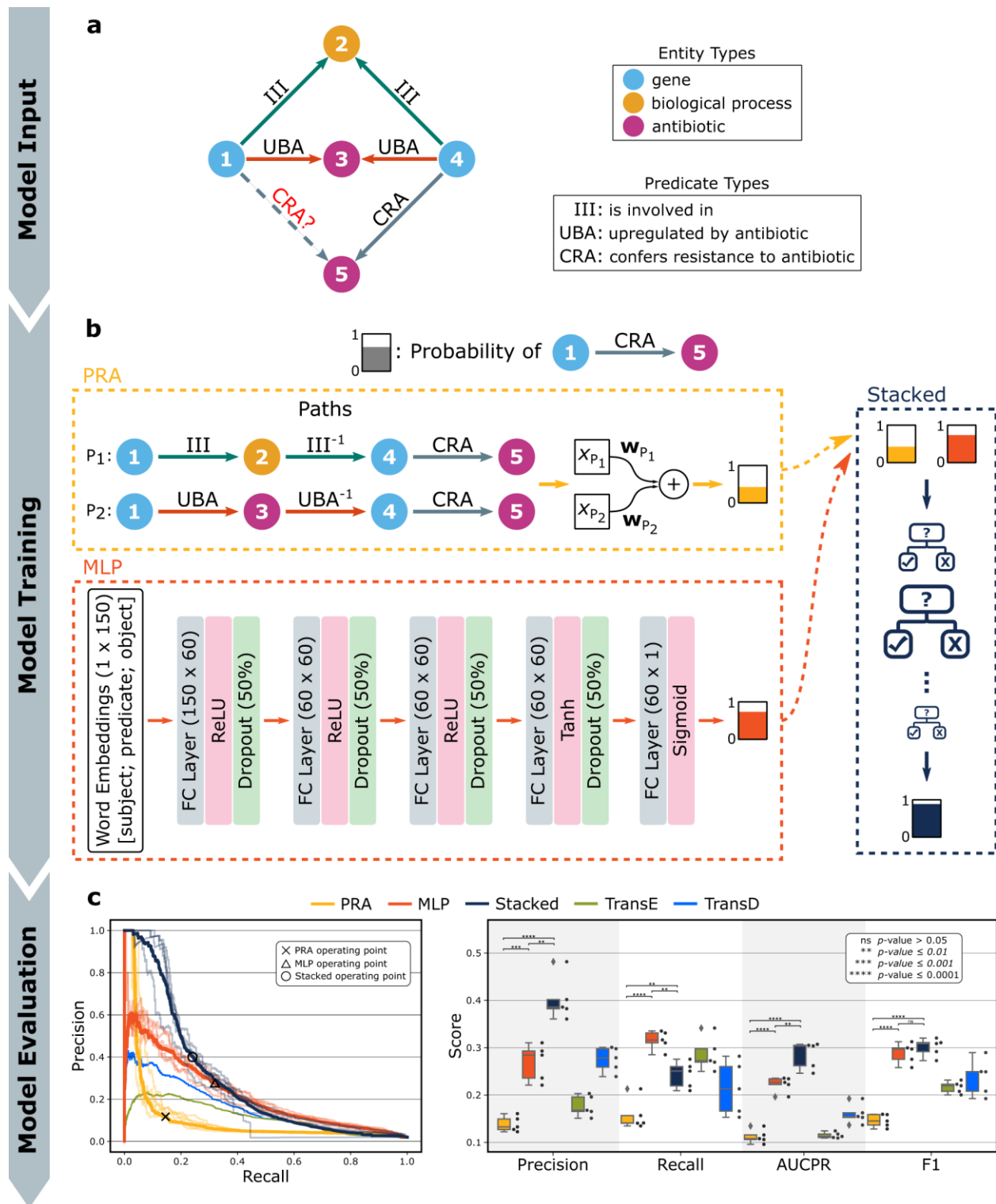


Figure 3.4. Hypothesis generator architecture, training, and evaluation. **a** Illustration of the training and evaluation of the hypothesis generator (HG). The task of the HG is to associate a probability to a putative link for the ‘confers resistance to antibiotic’ or CRA

between two nodes (nodes 1 and 5 here). **b** Three HG architectures, PRA, MLP, and Stacked, an ensemble method of a majority voting schema of the other two, were constructed and evaluated. Additional translation-based models like TransE and TransD were also tested although not illustrated here. **c** Precision-recall, AUCPR, and F1-score for the five methods ($n = 5$, 5-fold cross-validation). Black circles denote raw data points. The box represents the interquartile range, the middle line represents the median, the whisker line extends from minimum to maximum values, and the diamond represents outliers. For PRA vs MLP, all scores were statistically significant (precision p -value = 1.1×10^{-4} , recall p -value = 1.4×10^{-5} ; AUCPR p -value = 2.9×10^{-6} ; F1-score p -value = 1.6×10^{-6}). For PRA vs Stacked, all scores were also statistically significant (precision p -value = 2.2×10^{-6} , recall p -value = 2.0×10^{-3} ; AUCPR p -value = 2.1×10^{-6} ; F1-score p -value = 3.9×10^{-7}). Finally, for MLP vs Stacked, all scores were significant (precision p -value = 1.1×10^{-3} , recall p -value = 1.5×10^{-3} ; AUCPR p -value = 3.0×10^{-3}) except for F1-score (p -value = 0.37). Note that all methods have been optimized for the F1-score, and the p -values were calculated using the two-sided t-test.

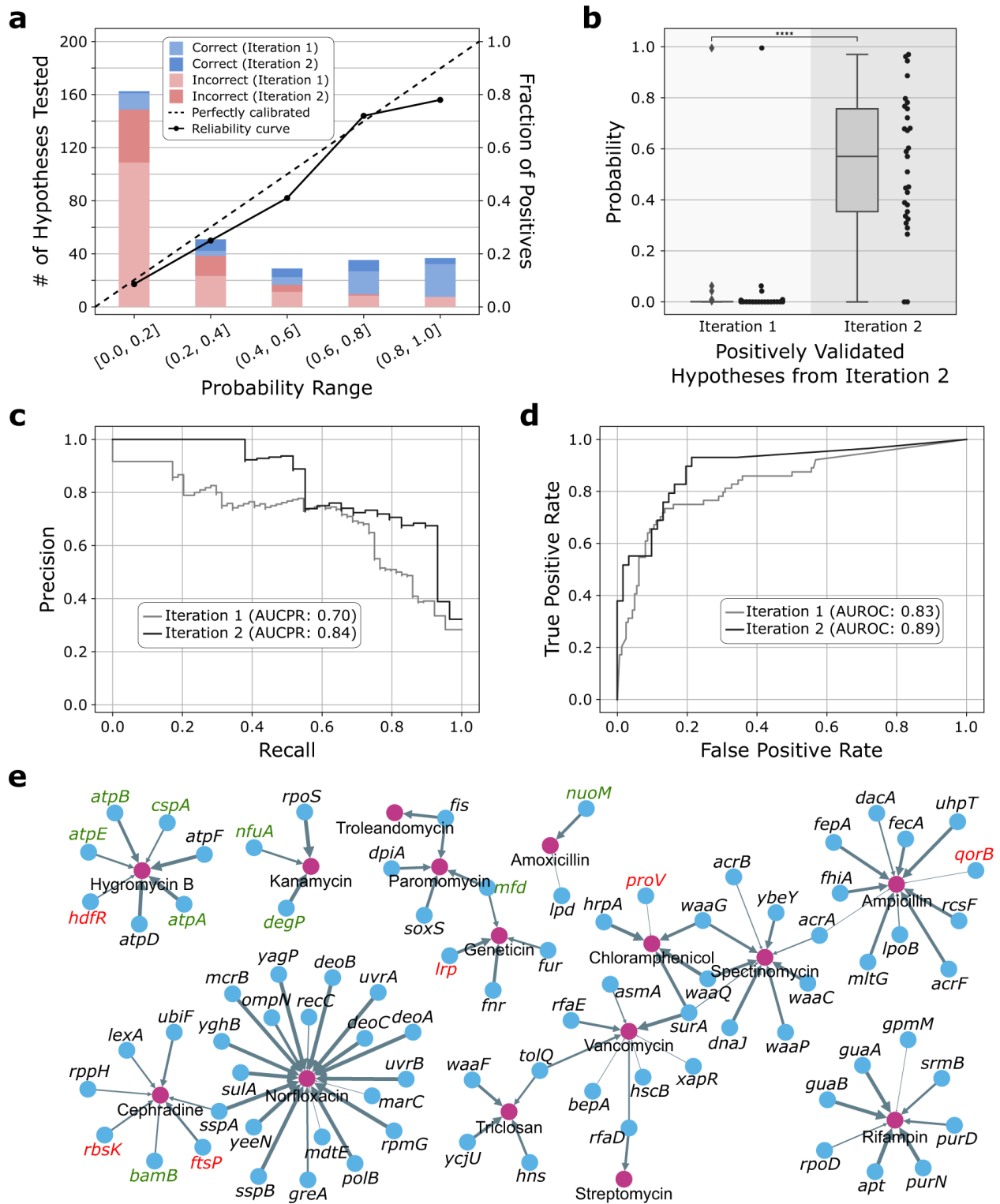


Figure 3.5. Accelerated missing link discovery through iterative learning. **a** A high correlation between the probability assignment by the hypothesis generator and forward experimental validation (226 and 90 validated hypotheses from the first and second

iteration, respectively; $R^2=0.94$). **b** The probability distribution of the positively validated hypotheses from the second iteration (i.e., the dark blue bar in **Fig. 5a**) compared to the probability of the same hypotheses from the first iteration ($n = 29$ positively validated second iteration hypotheses). Updating the knowledge graph with the validated hypotheses in the first iteration (i.e., light blue and red bars in **Fig. 5a**) and re-training of the hypothesis generator led to the 14-fold probability increase (0.55 vs. 0.04, respectively, p -value = 1.1×10^{-11}), which in turn enabled the discovery that would not have been possible with only one iteration of hypothesis generation. The box represents the interquartile range, the middle line represents the median, the whisker line extends from minimum to maximum values, and the diamond represents outliers. The p -value was calculated using the two-sided t-test. **c, d** The precision-recall (PR) and receiver operating characteristic (ROC) curves of the generated hypotheses compared against our wet-lab validation results. The AUCPR and AUROC of the second iteration hypotheses increased by 19.4% and 7.3%, respectively, when compared to the first iteration hypotheses. **e** We predicted and validated 64 CRA hypotheses from iteration 1 and 29 CRA hypotheses from iteration 2 for a total of 83 *E. coli* genes (blue node) that confer resistance (gray arrow) to one or more of 15 antibiotics (purple node). Genes with green and red labels indicate previously unknown genes that are not associated with antibiotic resistance in *E. coli* (9 genes) or any microbe (6 genes), respectively. The edge thickness is proportional to the KIDS predicted probability.

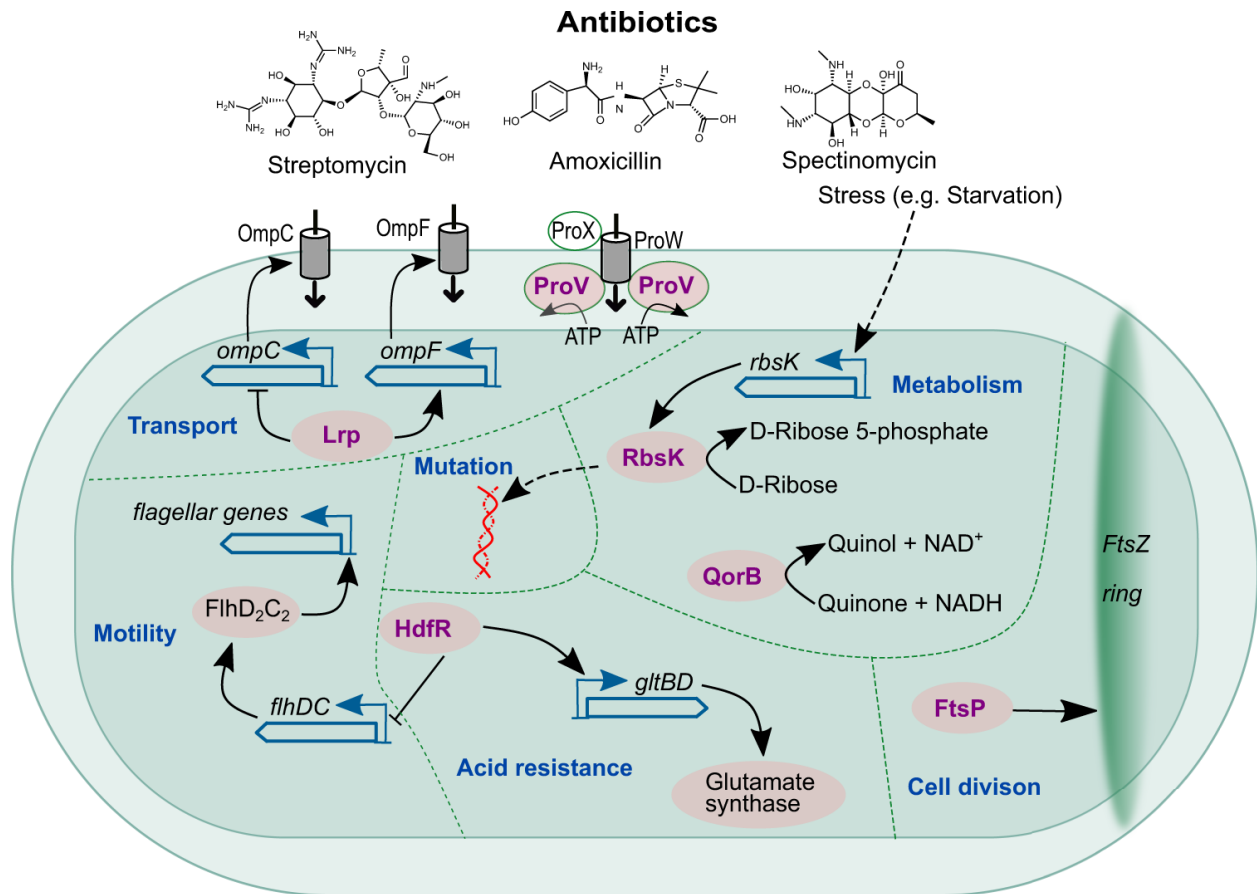


Figure 3.6. Mode of action of 6 previously unknown genes discovered to be involved in antibiotic resistance*. The proteins of these genes are shown in purple. Solid arrows indicate upregulation while blocking bars indicate downregulation. The dotted arrows indicate indirect regulation.

*Credit to Navneet Rai, reproduced from “*Knowledge integration and decision support for accelerated discovery of antibiotic resistance genes*”⁹⁸ with permission.

Chapter 4

Integrating knowledge graph structure in language models for link prediction

Disclaimer. All the work that is presented in this chapter has been published in *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*¹⁹⁹.

4.1 Introduction

While the utilization of traditional statistical machine learning models for hypothesis generation in the life sciences has shown promise^{25,98}, the landscape has evolved, necessitating a shift towards more advanced and nuanced methodologies. Recognizing the intricacies of biological data and the need for more sophisticated approaches, there is a growing consensus on the incorporation of state-of-the-art machine learning models that integrate natural language processing (NLP) along with knowledge graphs as input^{74,200}. These advanced models, armed with the ability to comprehend and analyze textual information, can extract valuable insights from the ever-expanding biomedical literature. By harnessing the power of NLP, researchers can not only navigate the vast sea of unstructured text but also uncover latent relationships and patterns within the scientific literature, contributing to a more comprehensive and contextually informed

hypothesis generation process. As we delve into the realm of cutting-edge methodologies, the integration of NLP into hypothesis generation models emerges as a pivotal step forward in ensuring the relevancy and efficacy of computational approaches in the life sciences.

In this work, we propose the Knowledge Graph Language Model (KGLM) (**Figure 4.2**), a simple yet effective language model pre-training approach that learns from both the textual and structural information of the knowledge graph. We continue pre-training the language model that has already been pre-trained on other large natural language corpora using the corpus generated by converting the triplets in the knowledge graphs as textual sequences while enforcing the model to better understand the underlying graph structure and by adding an additional entity/relation-type embedding layer. Testing our model on the WN18RR dataset for the link prediction task shows that our model improved the mean rank by 21.2% compared to the previous state-of-the-art method (51 vs. 40.18, respectively). All code and instructions on how to reproduce the results are available online (<https://github.com/ibpa/KGLM>).

4.2 Background

The link prediction (LP) task, one of the commonly researched knowledge graph completion tasks, attempts to predict the missing head entity (h) or tail entity (t) of a triplet (h, r, t) given a KG $G = (E, R)$, where $\{h, t\} \in E$ is the set of all entities and $r \in R$ is the set of all relations. Specifically, given a single test positive triplet (h, r, t) , its

corresponding link prediction test dataset can be constructed by corrupting either the head or the tail entity in the filtered setting⁶⁷ as

$$D_{LP}^{(h,r,t)} = \{(h,r,t') | t' \in (E - \{h,t\}) \wedge (h,r,t') \notin D\} \cup \{(h',r,t) | h' \in (E - \{h,t\}) \wedge (h',r,t) \notin D\} \cup \{(h,r,t)\},$$

Where $D = D_{train} \cup D_{val} \cup D_{test}$ is the complete dataset. Evaluation of the link prediction task is measured with mean rank (MR), mean reciprocal rank (MRR), and hits@N²⁰¹. MR is defined as

$$MR = \frac{\sum_{(h,r,t) \in D_{test}} rank((h,r,t) | D_{LP}^{(h,r,t)})}{|D_{test}|}$$

where $rank(\cdot | \cdot)$ is the rank of the positive triplet among its corrupted versions and $|D_{test}|$ is the number of positive test triplets. MRR is the same as MR except that the reciprocal rank $1/rank(\cdot | \cdot)$ is used. Hits@N is defined as

$$hits@N = \frac{\sum_{(h,r,t) \in D_{test}} \begin{cases} 1, & \text{if } rank((h,r,t) | D_{LP}^{(h,r,t)}) < N \\ 0, & \text{otherwise} \end{cases}}{|D_{test}|}$$

where $N \in \{1,3,10\}$ is commonly reported. Higher MRR and hits@N values are better, whereas, for MR, lower values denote higher performance.

4.3 Proposed approach

In this work, we propose to continue pre-training, instead of pre-training from scratch, the language model RoBERTa_{LARGE}⁷¹ that has already been trained on English-language corpora of varying sizes and domains, using both the forward and inverse knowledge graph textual sequences (**Figure 4.2**). Following the convention used in the KG-BERT and StAR (see Appendix~\ref{sec:previous_work}), we use a textual representation of a given triplet, e.g., (*Bill Gates, founderOf, Microsoft*) as '*Bill Gates founder of Microsoft*', to generate the pre-training corpus. However, instead of extracting only the forward triplet as done in the previous work, we extract both the forward and inverse versions of the triplet, e.g., (*Jennifer Gates, daughterOf, Bill Gates*) and (*Bill Gates, daughterOf⁻¹, Jennifer Gates*), where the ⁻¹ notation denotes the inverse direction of the corresponding relation.

To enforce the model to learn the knowledge graph structure, we introduce a new embedding layer *entity/relation-type embedding* (ER-type embedding) in addition to the pre-existing token and position embeddings of RoBERTa as shown in **Figure 4.2**. This additional layer aims to embed the tokens in the input sequence with its corresponding entity/relation-type, where the set of entities E in the knowledge graph can have t_E different entity types depending on the schema of the knowledge graph, (e.g., $t_E = 3$ for person, company, and location in **Figure 4.1**). Note that many knowledge graphs do not specify the entity types, in which case $t_E = 1$. For the set of relations R , there exist $t_R = 2n_R$, where n_R is the number of unique relations in the knowledge graph and the multiplier of 2 comes from forward and inverse directions (e.g., $n_R = 10$ for the sample knowledge graph in **Figure 4.1**).

In this work, we propose three different variations of ER-type embeddings. $KGLM_{Base}$ is the simplified version where all entities are assigned a single entity type and relations are assigned either forward or inverse relation type regardless of their unique relation types, resulting in a total of 3 ER-type embeddings. The $KGLM_{GR}$ is a version with granular relation types with $t_R + 1$ ER-type embeddings. The $KGLM_{GER}$ is the most granular version where we utilize all $t_E + t_R$ ER-type embeddings. In other words, all entity types as well as all relation types including both directions are considered.

To be specific, we convert a triplet (h, r, t) to a sequence of tokens $w^{(h,r,t)} = \langle [s]w_a^h w_b^r w_c^t [/s] : a \in \{1..|h|\} \& b \in \{1..|r|\} \& c \in \{1..|t|\} \rangle \in \mathbb{R}^{(|h|+|r|+|t|+2)}$, where $[s]$ and $[/s]$ are special tokens denoting the beginning and end of the sequence, respectively. The input to the RoBERTa model is then constructed by adding the ER-type embedding $\mathbf{t}^{(h,r,t)}$ and the $\mathbf{p}^{(h,r,t)}$ position embeddings to the $\mathbf{w}^{(h,r,t)}$ token embeddings, as

$$\mathbf{X}^{(h,r,t)} = \mathbf{w}^{(h,r,t)} + \mathbf{p}^{(h,r,t)} + \mathbf{t}^{(h,r,t)}.$$

Unlike the segment embeddings in the KG-BERT and StAR that were used to mark the input tokens with either the entity (s_e) or relation (s_r), the ER-type embedding now replaces its functionality. Finally, we pre-train the model using the masked language model (MLM) training objective⁷¹.

For fine-tuning, we extend the idea of how the KG-BERT scores a triplet to take advantage of the ER-type embeddings learned in our pre-training stage. For a given target triplet, we calculate the weighted average score of both directions as

$$score_{KGLM}(h, r, t) = \alpha SeqCls(\mathbf{X}^{(h,r,t)}) + (1 - \alpha) SeqCls(\mathbf{X}^{(t,r^{-1},h)}),$$

where $SeqCls(\cdot)$ is a RoBERTa model transformer with a sequence classification head on top of the pooled output (last layer hidden-state of the [CLS] token followed by dense layer and \tanh activation function), (t, r^{-1}, h) denotes the inverse version of (h, r, t) , and $0 \leq \alpha \leq 1$ denotes the weight used for balancing the scores from forward and inverse scores. For example, $\alpha = 1.0$ considers only the forward direction score.

4.4 Experiments and results

4.4.1 Datasets

We tested our proposed method on three benchmark datasets WN18RR, FB15k-237, and UMLS as shown in **Table 4.1**. WN18RR²⁰² is derived from WordNet²⁰³, a large English lexical database of semantic relationships between words, FB15k-237²⁰⁴ is extracted from Freebase²², a large community-drive KG of general facts about the world, and UMLS contains biomedical relationships. WN18RR and FB15k-237 are subsets of WN18⁶⁷ and FB15k⁶⁷, respectively, where the *inverse relation test leakage* problem, i.e. the problem of inverted test triplets appearing in the training set, has been corrected.

4.4.2 Settings

We used RoBERTa_{LARGE}⁷¹, a BERT_{LARGE}-based architecture with 24 layers, 1024 hidden size, 16 self-attention heads, and 355M parameters, for the pre-trained language model as it has been shown in a previous study to perform better than BERT (hits@1 0.243 vs. 0.222 and MR 51 vs. 99, link prediction on WN18RR)⁷⁴. For pre-training, we used learning rate = 5e-05, batch size = 32, epoch = 20 (WN18RR), 10 (FB15k-237), and 1,000 (UMLS),

and AdamW optimizer²⁰⁵. For fine-tuning training data, we sampled 10 negative triplets for a positive triplet by corrupting both the head and tail entity 5 times each. We used the validation set to find the optimal learning rates = $\{1e - 06, 5e - 07\}$, batch size = $\{16, 32\}$, epochs = $\{1, 2, 3, 4, 5\}$ for WN18RR and FB15k-237 and 25,50,75,100 for UMLS, and α from 0.0 to 1.0 with an increment of 0.1. For all experiments, we set $\alpha = 0.5$ based on the WN18RR validation set performance. Both pre-training and fine-tuning were performed on $3 \times$ Nvidia Quadro RTX 6000 GPUs in a distributed manner using the 16-bit mixed precision and DeepSpeed^{206,207} library in the stage-2 setting. We used the Transformers library²⁰⁸.

4.4.3 Link prediction results

The hypothesis behind the KGLM was that learning the ER-type embedding layers in the pre-training stage using the corpus generated by the knowledge graph, followed by fine-tuning has the best performance. To test our hypothesis, we broke down the hypothesis into two separate claims. For the first claim, we only continued pre-training RoBERTa_{LARGE} followed by fine-tuning without the ER-type embeddings. This test removes the contribution from the ER-type embeddings and solely tests the performance gained by further pre-training the model with the knowledge graph as input. **Table 4.3** shows that claim 1 falls behind the KGLM_{GR} in all metrics except for hits @1 (0.331 vs. 0.330, respectively). For the second claim, we did not continue pre-training and instead used the RoBERTa_{LARGE} pre-trained weights as-is. We then learned the ER-type embeddings in the fine-tuning stage. This test shows if the ER-type embeddings can be learned only during the fine-tuning stage. **Table 4.3** shows that KGLM_{GR} outperforms all

of the metrics obtained using the second claim. This result shows that the combination of these two claims works in a non-linear fashion to maximize performance.

The results of performing link prediction on the benchmark datasets are shown in **Table 4.2**. Compared to StAR, which had the best performance on MR and hits@10 on WN18RR, KGLM_{GR} outperformed all the metrics with 21.2% improved MR (40.18 vs. 51, respectively) and 4.5% increased hits@10 (0.709 vs. 0.741, respectively). Although still inferior compared to the graph embedding approaches, KGLM_{GR} has 35.8% improved hits@1 compared to the best language model-based approach StAR (0.243 vs. 0.330, respectively). Across all model types, KGLM_{GR} has the best performance on all metrics for WN18RR except for hits@1. Although we did not observe any improvement compared to StAR for the FB15k-237 dataset, we had the best performance on all metrics for UMLS with 21.2% improved MR than ComplEx (1.19 vs. 1.51, respectively). KGLM_{GR} outperformed KGLM_{Base} in all metrics.

4.5 Conclusion

In this work, we presented KGLM, which introduces a new entity/relation (ER)-type embedding layer for learning the structure of the knowledge graph. Compared to the previous language model-based methods that only fine-tune for a given task, we found that learning the ER-type embeddings in the pre-training stage followed by fine-tuning resulted in better performance. In future work, we plan to further test the version of KGLM that takes into account entity types, KGLM_{GER}, on domain-specific knowledge graphs like KIDS⁹⁸ with entity types in their schema.

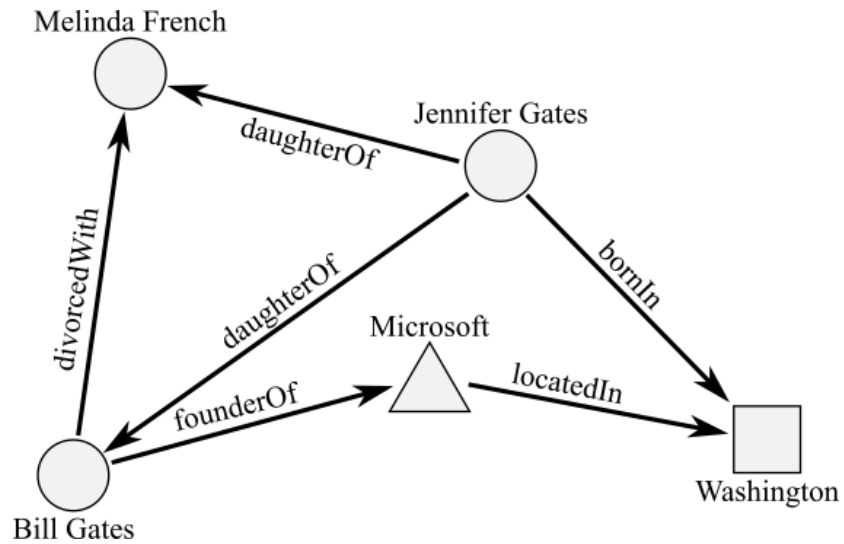


Figure 4.1. Sample knowledge graph with 6 triplets. The graph contains three unique entity types (circle for person, triangle for company, and square for location) and 5 unique relation types, or 10 if considering both the forward and inverse relations. The task of the knowledge graph completion is to complete the missing links in the graph, e.g., (*Bill Gates, bornIn?, Washington*) using the existing knowledge graph.

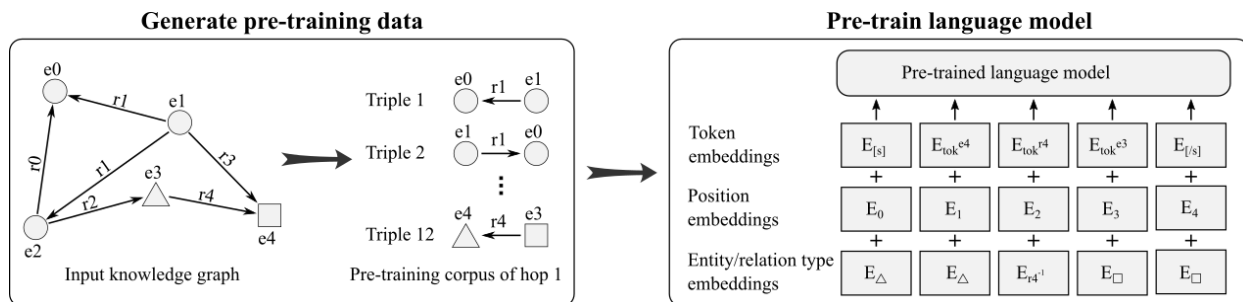


Figure 4.2. The proposed pre-training approach of the KGLM. First, both the forward and inverse triplets are extracted from the knowledge graph to serve as the pre-training corpus. We then continue pre-training the language model, RoBERTa in our case, using the masked language model training objective, with an additional entity/relation-type embedding layer. The entity/relation-type embedding scheme shown here corresponds to the KGLM_{GER}, the most fine-grained version where both the entity and relation types are considered unique. Note that the inverse relation denoted by $^{-1}$ is different from its forward counterpart. For demonstration purposes, we assume all entities and relations to have a single token.

Table 4.1. Statistics of the benchmark knowledge graphs used for link prediction.

Dataset	# ent	# rel	# train	# val	# test
WN18RR	40,943	11	86,835	3,034	3,134
FB15k-237	14,951	237	272,115	17,535	20,466
UMLS	135	46	5,216	652	661

Table 4.2. Link prediction results on the benchmark datasets WN18RR, FB15k-237, and UMLS. Bold numbers denote the best performance for a given metric and class of models. Underlined numbers denote the best performance for a given metric regardless of the model type. Note that we do not report KGLM_{GER} performance since the tested datasets do not specify entity types in their schema.

Method	WN18RR					FB15k-237					UMLS	
	Hits @1	Hits @3	Hits @10	MR	MRR	Hits @1	Hits @3	Hits @10	MR	MRR	Hits@10	MR
<i>Model type: Not based on language models</i>												
TransE	.043	.441	.532	2300	.243	.198	.376	.441	323	.279	.989	1.84
TransH	.053	.463	.540	2126	.279	.306	.450	.613	219	.320	-	-
DistMult	.412	.470	.504	7000	.444	.199	.301	.446	512	.281	.846	5.52
ComplEx	.409	.469	.530	7882	.449	.194	.297	.450	546	.278	.967	2.59
ConvE	.390	.430	.480	5277	.46	.239	.350	.491	246	.316	.990	1.51
RotatE	.428	.492	.571	3340	.476	.241	.375	.533	177	.338	-	-
GAAT	.424	.525	.604	1270	.467	<u>.512</u>	<u>.572</u>	<u>.650</u>	187	<u>.547</u>	-	-
LineaRE	<u>.453</u>	.509	.578	1644	<u>.495</u>	.264	.391	.545	155	.357	-	-
QuatDE	.438	.509	.586	1977	.489	.268	.400	.563	<u>90</u>	.365	-	-
<i>Model type: Based on language models</i>												
KG-BERT	.041	.302	.524	97	.216	-	-	.420	153	-	.990	1.47
StAR	.243	.491	.709	51	.401	.205	.322	.482	117	.296	.991	1.49
KGLM _{Base}	.305	.518	.730	47.97	.445	-	-	-	-	-	-	-
KGLM _{GR}	.330	<u>.538</u>	<u>.741</u>	<u>40.18</u>	.467	.200	.314	.468	125.9	.289	<u>.995</u>	<u>1.19</u>

Table 4.3. Breakdown of the original hypothesis and their results on WN18RR. For claim 1, we continued to pre-train RoBERTa_{LARGE} using the knowledge graph without the ER-type embeddings. Note that we did not also use the ER-type embeddings layer in the fine-tuning stage. For claim 2, we learned the ER-type embeddings in the fine-tuning stage only without any further pre-training.

Model	ER-type embeddings			Hits @1	Hits @3	Hits @10	MR	MRR
	Continue pre-training	Pre-train	Fine-tune					
Claim 1	o	x	x	0.331	0.529	0.728	53.5	0.462
Claim 2	x	-	o	0.322	0.489	0.672	66.4	0.439
KGLM _{GR}	o	o	o	0.330	0.538	0.741	40.18	0.467

Chapter 5

Automated knowledge extraction of food and chemicals from the literature

Disclaimer. All the work that is presented in this chapter is currently under review in *Science Advances*.

5.1 Introduction

As we navigate the evolving landscape of knowledge representation and hypothesis generation in the life sciences, it becomes evident that addressing the challenges of data curation and reproducibility is paramount for fostering a more robust scientific foundation. The integration of advanced machine learning models, particularly those incorporating natural language processing, marks a significant leap forward in accelerating knowledge discovery. However, the intricacies of data curation persist, prompting us to explore novel approaches. In the following chapter, we shift our focus to the realm of chemical food composition, where the need for precise and reliable data is equally crucial. Here, we delve into the complexities of characterizing the chemical makeup of food, exploring how advancements in computational methodologies can revolutionize the understanding and representation of nutritional information. By bridging the gaps in data curation and reproducibility, we aim to contribute to a more comprehensive and reliable foundation for

scientific inquiry, particularly in the context of understanding the chemical composition of the foods we consume.

Mapping the chemical composition of food and ingredients is essential for unlocking their potential and informing decisions. From creating healthier and tastier food products^{209,210} to enriching food with the right compounds^{211,212} or building personalized diets^{213–215}, understanding what is in each ingredient and at what concentration is paramount. Food composition at the molecular level is usually found in food composition tables like the USDA's FoodData Central (FDC)²¹⁶ or the ANSES-CIQUAL database²¹⁷. This enables several stakeholder groups, from researchers to policymakers, to assess the nutrition quality of various foods and their regulatory status and to use them in the respective industries²¹⁸. However, despite the established importance of the food composition information, most of the food-chemical information that is present in the scientific literature is not captured in the structured databases²⁰⁹. For instance, the total size of food composition space is estimated at tens of thousands of chemicals²¹⁹, while FDC and ANSES-CIQUAL focus on only 500 compounds. To expand the coverage of chemicals in foods, several initiatives attempt to capture food composition from scientific literature, such as FooDB²²⁰ (797 foods and 15,750 detected chemicals) and DietRx²²¹ (2,222 foods and 6,992 chemicals), which further aggregate data from several other databases like FDC²¹⁶, KNApSAcK²²², Dr. Duke's Phytochemical and Ethnobotanical Databases²²³, Phenol-Explorer^{37,224,225}, and PhytoHub²²⁶. However, existing databases require laborious annotation effort from experts or lack consistent quality control as the majority of their food-chemical composition information is not linked to evidence that allows reproducible results. For example, less than 1% of associations in FooDB, one of the

most notable DBs in this space, have literature citations to support them (**Appendix A.3.1.1**).

In this work, we present the Lit2KG framework (**Figure 5.1a**) that extracts information from scientific literature using a large language model in an AL setting to construct a large-scale KG. The entailment model of the Lit2KG framework uses a premise from the scientific literature to extract and predict multiple hypotheses with high performance (F1 score of 83%), with the predicted probabilities being highly correlated to the ground-truth annotations ($R^2 = 0.94$). We also tested four different AL strategies and found that selecting samples that maximize the likelihood leads to discovering new knowledge 38.2% faster than the baseline. Applying graph-embedding link prediction models for graph completion followed by validation through literature search revealed 355 missed food-chemical composition associations that were further verified manually and 11 additional associations that were novel, 6 of which we have found strong evidence to support them. The resulting knowledge graph contains 285,077 triplets of three entity types (food, part, chemical) and four relation types (*contains*, *has part*, *is a*, *has child*) on three evidence quality levels (high, medium, low) with 4,318 of them evaluated by human experts (**Figure 5.1b**).

5.2 Methods

5.2.1 Premise-hypothesis pair generation.

We collected a total of 1,959 raw and non-processed food names that have a known National Center for Biotechnology Information (NCBI) Taxonomy ID²²⁷ from multiple food databases. We then used the LitSense API²²⁸, which is a search system for biomedical literature at the sentence level provided by the NCBI, to query for the search keyword “{*food name*} contains” (**Appendix A.3.2.1**). The LitSense API returns sentence-level text snippets from the PubMed abstracts and the PMC open-access full-text articles, as well as the named entity recognition (NER) service for species and chemical entities, along with their corresponding NCBI Taxonomy IDs and MeSH IDs, respectively. We further processed these text snippets by discarding non-food entities and tagging the part entities (e.g., leaf and root) using our manually generated lookup table consisting of 70 food parts.

For each LitSense-returned sentence $s_i \in S$, which we refer to as a *premise* in our work, there exist three sets of named entities F_i , P_i , and C_i for food, parts, and chemicals, respectively, where P_i can be an empty set as not all sentences have parts in them. We then generated a set of hypotheses H_i for each premise s_i by taking the cartesian product of the entity sets F_i , P_i , and C_i as

$$H_i = \{template(f, p, c) \forall (f, p, c) \in F_i \times P_i \times C_i\} \cup \{template(f, c) \forall (f, c) \in F_i \times C_i\},$$

where $template(\cdot)$ is the hypothesis template that generates a triplet of type ($\{\{food\} \{part\}, contains, \{chemical\}\}$ or ($\{\{food\}, contains, \{chemical\}\}$), respectively. We refer to these pairs of premise and the extracted hypotheses as premise-hypotheses (PH) pairs in our work (see **Supplementary Figure 21**).

5.2.2 Premise-hypothesis pair annotation.

We annotated the PH pairs to generate a dataset for training, validating, and testing the entailment model using the AL strategy (described in the following sections). During the annotation process, a given PH pair was assigned one of three possible classes *entails*, *does not entail*, and *skip*. More specifically, *entails* was assigned if the premise supported the underlying relationship used to construct the hypothesis, and *does not entail* was assigned if there was insufficient evidence in the premise to support the hypothesis. Note that the hypothesis from a PH pair marked as *does not entail* is not necessarily a negative, as another premise may support the hypothesis. Finally, *skip* was assigned if the premise the LitSense API returned was not formatted correctly or if the NER tagging by LitSense API was wrong (**Appendix A.3.2.2**). To ensure the annotation was of high quality, two experts annotated each PH pair independently, and only the PH pairs that had agreed annotation results by the two experts were kept. We randomly split the data into training, validation, and test sets with approximate ratios of 70%, 15%, and 15%. To avoid data leakage, we ensured that the three datasets did not share the same premises or hypotheses during the splitting. In the end, we had a training set with 4,120 PH pairs (1,899 *entails*, 2,221 *does not entail*), a validation set with 825 PH pairs (295 *entails*, 530 *does not entail*), and a test set with 840 PH pairs (312 *entails*, 528 *does not entail*).

5.2.3 Entailment model.

We trained the entailment model to predict whether the premise logically would entail the hypotheses. To this end, we used the BioBERT²²⁹ over other language models^{230–232} (**Appendix A.3.2.3**), as it was pre-trained on the same corpus as where the premises were extracted from (PubMed abstracts and PMC full-text articles) and have

demonstrated improved performance on biomedical benchmarks²²⁹. We then fine-tuned the BioBERT entailment model by utilizing the binary classification schema, where the input sequence was formatted by concatenating the premise and hypothesis with the [SEP] token in between, and the model predicted if the given PH pair *entails* or *does not entail*. We used the held-out validation set to optimize the hyperparameters, where the tunable hyperparameters were learning rate = $\{2 \times 10^{-5}, 5 \times 10^{-5}\}$, epochs = $\{3, 4\}$, and batch size = $\{16, 32\}$. The hyperparameter set with the best held-out validation precision was selected, and the performance of each round was reported using the held-out test set. Note that we trained a production entailment model using all the labeled data (*i.e.*, training, validation, and test sets) (**Appendix A.3.2.3**).

5.2.4 Active learning strategy.

In this work, we tested four active learning (AL) strategies, *maximum likelihood*, *maximum entropy*, *stratified*, and *random*. We simulated the AL strategy by splitting the training pool with 4,120 PH pairs into ten rounds $r = \{1, 2, \dots, 10\}$, with 412 new PH pairs selected in each round and appended to the existing training data by the respective strategy. In other words, at round r , we trained the entailment model m_r using $412 \times (r - 1)$ training PH pairs plus 412 new PH pairs selected from the remaining $412 \times (10 - r + 1)$ PH pairs. We call this training and evaluation process a *run*, and we repeated 100 *runs* for each AL strategy to test the statistical significance. The *stratified* strategy first ranked the remaining PH pairs from high to low probability and split them into ten equally sized bins, randomly drawing the same number of samples from each bin. The *maximum likelihood* strategy chose the top 412 positive samples based on their probability score. The *maximum*

entropy sampling strategy first computed the uncertainty for each PH pair as $\min(1 - p, p)$, where p is the probability of the given PH pair predicted by the entail model. All PH pairs were then ranked using the uncertainty value from high to low, and the top 412 uncertain PH pairs were selected. Finally, the *random* sampling strategy chose 412 PH pairs randomly. Note that for the first round, all four AL strategies randomly selected the first round of PH pairs to train on, and for the last round, all four AL strategies were trained on a whole training pool of 4,120 PH pairs regardless of the sampling strategy taken. More detailed information can be found in **Appendix A.3.2.3.3**, and a visual illustration of the sampling strategies is in **Supplementary Figure 22**.

5.2.5 Knowledge graph generation.

The FoodAtlas Knowledge Graph $FAKG = (E, R)$ encodes information using a bag of triplets (h, r, t) , where $\{h, t\} \in E$ is the set of all entities (h for the head entity and t for the tail entity) and $r \in R$ is the set of all relation. Each triplet in the KG can have one or more sources and qualities. In this work, we define three qualities *high*, *medium*, and *low* for a triplet. The *high*-quality triplets have been validated by the FoodAtlas team and have PMID and/or PMCID. The *medium*-quality triplets are not validated by the FoodAtlas team but have PMID and/or PMCID. Taxonomy and ontology also are medium-quality triplets. The *low*-quality triplets are not validated by the FoodAtlas team and do not have PMID or PMCID. Please refer to **Appendix A.3.3** for the details of the FAKG design including the entity and relation types.

The first source of information was from the PH pair annotation process, where two relation types, *contains* and *has part*, exist. The triplets with the *contains* relation type

were from the positive annotated PH pairs, whereas the triplets with the *has part* relation type were automatically extracted from the *contains* triplets. For example, a triplet (*coconut, has part, coconut seed*) was extracted from the triplet (*coconut seed, contains, lauric acid*). All triplets from this source were high-quality. The second source was the entailment model predictions, also with the *contains* and *has part* relation types. However, these were not annotated and thus were assigned a medium-quality. The third source was the enrichment through the NCBI Taxonomy and MeSH tree ontology. The NCBI Taxonomy, which contains medium-quality triplets with the *has child* relation type, encodes the hierarchical structure of the taxonomic lineage (*Cocos (genus), has child, Cocos nucifera (species)*). The MeSH tree, which contains medium-quality triplets with the *is a* relation type, encodes the ontological relationship of the chemical entities. We also included the triplets extracted from the external databases (Frida²³³, FDC, and Phenol-Explorer) with either *medium-* or *low-*quality triplets with the *contains* relation type. Finally, we also included the link prediction results (triplets with the *contains* relation type) as low-quality.

5.2.6 Link prediction.

Link prediction is a widely studied field that refers to the task of predicting missing relationships or links between entities in a graph, (food, contains, chemical) triplet type in our case, and contributes to the enhancement and enrichment of knowledge graphs²³⁴. Using the Python library PyKEEN²³⁵, we trained a set of benchmark link prediction models TransE²³⁶, ER-MLP²³⁷, DistMult²³⁸, TransD²³⁹, ComplEx²⁴⁰, and RotatE⁶⁹ on different versions of the FAKG (**Figure 5.5a,b**), performed hyperparameter optimization on the

held-out validation set using mean rank (MR), and reported the results on the held-out test set (**Appendix A.3.2.4**). The link prediction models were also calibrated using isotonic regression to provide an interpretable probability score. Link prediction models are commonly evaluated using rank-based metrics like mean rank (MR), mean reciprocal rank (MRR), hits@1, hits@3, and hits@10²⁴¹. However, our end goal was to generate hypotheses that were either true or false, and therefore, we decided to also evaluate using standard binary classification metrics like confusion matrix, precision, and recall. To this end, we randomly sampled two negatives for each positive triplet in the validation and test set by corrupting the head and tail entity once, which resulted in a validation set with 1,335 triplets (445 positives and 890 negatives) and a test set with 1,341 triplets (447 positives and 894 negatives). Due to the nature of the graph-embedding models that cannot make predictions on test triplets with an entity that is never seen during the training, we report our binary classification metrics in a stricter *unfiltered* setting, where the test triplets that would be dropped in the *filtered* setting are kept and assigned a default majority label 0.

5.2.7 Link prediction literature validation.

To validate the link prediction-generated food-chemical triplets, we searched the following four sources sequentially: PubChem taxonomy²⁴², Bing Chat, Google Scholar, and Google. Specifically, for a given food-chemical pair, we first checked if the Taxonomy section of PubChem entry for the chemical of interest lists the scientific name of the food and has a reference. If not, we then asked Bing Chat, a search engine based on a large language model, to find the reference (**Supplementary Figure 23**). Next, we searched

Google Scholar using a set of pre-defined search queries (**Appendix A.3.1.8.2**). If the initial Google Scholar search did not return a positive relationship within the first three pages (30 papers, 10 papers per page), we repeated the process with the synonyms of the entities. Finally, we searched the first 30 contents of Google using the same search method as Google Scholar. A complete procedure for the link prediction validation can be found in **Appendix A.3.1.8**.

5.3 Results

5.3.1 The FoodAtlas Knowledge Graph contains a wide spectrum of food-chemical composition information.

We utilized the Lit2KG framework (**Figure 5.1a**) to extract the food-chemical composition information from the PubMed abstracts and open-access articles using raw food ingredients as queries (see **Methods**). From this search, we generated 3,596,755 premise-hypotheses (PH) pairs where the hypotheses are (food, contains, chemical) or (food part, contains, chemical) triplets. We then used BioBERT²²⁹, a biomedical language representation model for triplet binary classification that we fine-tuned with 4,318 manually curated positive triplets in an active learning setting. This resulted in 230,504 additional positive triplets, for a total of 234,822 unique positive triplets. In addition, we curated and added the food-chemical composition information based on quality criteria from three external databases (8,375 triplets from Frida²³³, 1,055 triplets from Phenol-Explorer³⁷, and 529 triplets from FDC²¹⁶), taxonomical information of the foods using the NCBI Taxonomy (1,526 triplets), and ontological information of the chemicals using the

MeSH tree (43,691 triplets) (**Figure 5.2d**). Applying link prediction on the knowledge graph generated an additional 9,756 triplets of food and chemical pairs, 355 of them manually validated as positives. The final FoodAtlas knowledge graph (FAKG, **Figure 5.1b**) contains 536 food entities, 4,608 food parts, 15,462 chemical entities, and 285,077 unique triplets about food-chemical composition with four different relation types and three different entity types (**Figure 5.2a-g**).

In terms of triplet quality, FAKG has 4,318 (1.5%) high-quality (*i.e.*, validated by two experts), 264,455 (92.8%) medium-quality (*i.e.*, with at least one reference, but not manually validated), and 16,304 low-quality (5.7%) triplets (*i.e.*, no references, see **Methods** and **Figure 5.2b**). From those, 4,318, 226,437, and 9,756, respectively, have been uniquely captured by our Lit2KG pipeline and the link prediction analysis (**Appendix A.3.1.3** and **Figure 5.2b,c**). The top five foods whose chemical composition is most well documented in the knowledge graph are soybean (*Glycine max*), maize (*Zea mays*), rice (*Oryza sativa*), cucumber (*Cucumis sativus*), followed by tomato (*Solanum lycopersicum*) (**Supplementary Figure 24**).

5.3.2 FoodAtlas discovers complementary information to benchmark datasets.

To test how good the coverage of the food-chemical composition triplets from the Lit2KG pipeline is, we compared them with FoodMine²⁴³, a database that contains a manually curated chemical composition of two selected foods, cocoa (592 chemicals) and garlic (289 chemicals). Although there were initially 1,289 cocoa and 1,376 garlic chemicals in FAKG, we adopted the same method used by FoodMine to make chemicals in the two sources comparable and created an additional chemical identifier, specifically for

matching FoodMine chemicals with those in FAKG (**Appendix A.3.1.4**). After this processing step, the FAKG has 379 cocoa and 406 garlic chemicals, whereas FoodMine has 301 and 176, respectively. Out of 575 chemicals for cocoa, 274 (47.7%) chemicals were found in FAKG but not in FoodMine, 105 (18.3%) chemicals were common between the two, and 196 (34.1%) chemicals were not found in FoodAtlas (**Figure 5.3a**). For garlic, FoodAtlas was able to capture 51.1% (90 out of 176) of FoodMine chemicals, while 316 chemicals were unique to FAKG (**Figure 5.3a b**; see **Supplementary Figure 25** for a similar comparison with FooDB).

5.3.3 Maximum likelihood active learning strategy discovers knowledge 38% faster than without.

We fine-tuned the BioBERT-based entailment models based on four different AL strategies over ten rounds (see **Methods**; **Supplementary Figure 26**). Although all four AL strategies eventually discovered the same set of 1,899 positives among the 4,120 PH pairs in the training pool at the final round ($r = 10$), the maximum likelihood strategy identified the positives in training set by $38.2\% \pm 27.3\%$ faster than the active learning baseline of choosing random pairs, followed by the maximum entropy ($10.7\% \pm 6.6\%$) and stratified learning ($9.3\% \pm 5.3\%$; **Figure 5.4a,b** and **Appendix A.3.1.5**). Concomitantly, we observed lower performance for the entailment models trained using the maximum likelihood strategy than the others on all metrics for rounds 2 through 4 (adjusted p -value $< 3.6 \times 10^{-2}$). This was due to data imbalance, as the maximum likelihood strategy samples PH pairs that were highly probable, and thus its entailment models were trained on an unbalanced training set where on average, 74.9% of the

training data for rounds 2 through 4 was positive compared to 53.6%, 52.2%, and 46.1% for maximum entropy, stratified, and random, respectively, (**Supplementary Figure 27**).

For the final entailment models, AUCPR = 0.90 and AUROC = 0.94, where the baselines were 0.37 and 0.50, respectively (**Figure 5.4d,e** and **Supplementary Table 14**). The model PH prediction probability was well-calibrated and highly correlated with the actual ground truth statistics after manual validation ($R^2 = 0.94$, **Figure 5.4c**). For instance, 88.6% out of all triplets with a probability ≥ 0.9 were positives, whereas only 3.9% with a probability < 0.1 were positives (**Supplementary Figure 28**).

5.3.4 Sources of error and impact of large language model general knowledge.

Not surprisingly, the entailment model predicted best on straightforward, simple sentence structure, while its performance deteriorated when domain expertise was needed or premises were hypotheses posed by the authors as shown in index 5-8 of **Table 5.1** (see **Appendix A.3.1.6**). Variance across bootstrapped models was maximized with uncertainty: predictions with 40% to 60% probability had a standard deviation of 0.31 vs. 0.04 for predictions with less than 10% or more than 90% probability (p -value = 2.2×10^{-177}). Furthermore, analyzing the entailment model prediction results based on which section of the literature the premise was taken from (e.g., introduction, methods, etc.) revealed higher precision in certain sections. Unexpectedly, hypotheses stemming from the introduction and methods sections were associated with high precision (0.91 and 0.89, respectively) when compared to sections like abstract, title, and conclusion (0.77, 0.75, and 0.74, respectively; p -value: 9.7×10^{-76}) (see **Appendix A.3.1.7** and **Supplementary Table 15**).

5.3.5 Link prediction, GPT model, and the impact of ontologies in performance.

We trained a set of link prediction models for the *contains* relation between previously unknown food-chemical pairs (**Figure 5.5a**). The best performance was from TransD trained on the $FA_{A,R}$ (TransD- $FA_{A,R}$) with an overall best performance (precision: 79.3%, recall: 75.4%, and F1: 77.2%) (**Figure 5.5b**). However, as these models cannot classify triplets with entities not seen during the training phase, we used the next best model, RotatE- $FA_{A,E,R,P80}$ that has this capacity (precision: 76.8%, recall: 70.6%, and F1: 73.5%; **Figure 5.5c**). Interestingly, the inclusion of ontological information (Enrichment in **Figure 5.5b**), increases the F1 score by 22.2% (63.2% of FA_A vs. 77.2% of $FA_{A,R}$; p -value = 2.4×10^{-5}). Moreover, RotatE- $FA_{A,E,R,P80}$ is highly calibrated with $R^2 = 0.99$ (**Figure 5.5d**) and has an AUCPR of 0.82 (baseline 0.33) and AUROC of 0.88 (baseline 0.5) (**Figure 5.5e,f**). All link prediction models performed better than the generalized GPT-3.5 model (text-davinci-003), which was not fine-tuned using the KG (precision: 64.8%, recall: 31.8%, and F1: 42.7%) (**Appendix A.3.2.4**).

5.3.6 Link prediction reveals previously unknown food-chemical relationships.

The final FAKG contains 536 food entities (excluding food part entities) and 15,462 chemical entities, which translates to 8,287,632 possible food-chemical pairs. Only 1.72% (142,253 triplets) of these food-chemical pairs are connected via the *contains* relation, with the rest, 98.28% (8,145,379 triplets), being unknown. We, therefore, used RotatE to assign probability scores to these unknown pairs (**Figure 5.6a**), among which 9,756 pairs (0.1%) were assigned a positive prediction label (see **Methods**). Validating 443 sampled hypotheses from these pairs through an extensive literature search (**Figure 5.6b** and

Appendix A.3.1.8) revealed 355 positive *contains* triplets between 203 foods and 153 chemicals (**Figure 5.6c**), while 11 triplets remained yet unknown with no direct evidence (**Supplementary Table 16**).

A closer look at the 355 triplets demonstrated the importance of link prediction for knowledge graph completion. Linolenic acid, which is an essential omega-3 fatty acid that must be obtained through the diet and helps reduce inflammation²⁴⁴, lower blood pressure²⁴⁵, and improve cholesterol levels²⁴⁵, was validated to be found in 14 different foods (**Figure 5.6c**). The link prediction also discovered evident relationships such as the iodide ion, which is an essential trace element for vertebrates, and manganese(2+), which is a cofactor for many enzymes involved in metabolism²⁴⁶, including those that are important for bone development²⁴⁷ and antioxidant defense²⁴⁸, each with relationship to 10 different foods (**Figure 5.6c**). When it comes to foods, we identified five foods, *Lota lota* (NCBI:txid69944), *Brassica oleracea var. italica* (NCBI:txid36774), *Lupinus albus* (NCBI:txid3870), *Panax ginseng* (NCBI:txid4054), *Musa x paradisiaca* (NCBI:txid89151), that have largest number of positively validated positives to 5 chemicals each (**Figure 5.6c**).

5.3.7 AI-driven discovery of six food-chemical relationships.

We performed additional analysis for the 11 potential novel food-chemical candidates not reported in the literature (**Supplementary Table 16**) and found strong evidence that supports the relationships for 6 of them (**Figure 5.6d**). For the pairs (chickpea, triglyceride) and (Atlantic cod, beta-carotene), metabolic pathway analysis identified homologous enzymes directly associated with the synthesis or metabolism of chemicals in their

respective foods (**Appendix A.3.1.8**). For example, the enzyme phospholipid:diacylglycerol acyltransferase (PDAT), which produces triglyceride in genus *Arabidopsis*²⁴⁹, had the best hit for an enzyme PDAT 1-like in chickpea with 78.5% sequence similarity, while the enzyme beta-carotene-15,15'-dioxygenase, which metabolizes beta-carotene in human²⁵⁰, had 58.5% sequence similarity with beta,beta-carotene 15,15'-dioxygenase-like in the Atlantic cod. Similarly, for (dudaim melon, matairesinol), we found an enzyme secoisolariciresinol dehydrogenase for the biosynthesis of matairesinol in genetically close species *Cucumis melo* and varietas *Cucumis melo var. makuwa*²⁵¹, as we did not have the *Cucumis melo var. dudaim* genome to run a direct search. For the (bearded tooth, lumisterol) pair, we found the existence of ergosterol in bearded tooth²⁵² that converts to lumisterol under UV irradiation²⁵³, whereas, for the (cumin, sodium caffeate) pair, we found caffeic acid present in cumin²⁵⁴, which reacts with sodium in plants through neutralization of acid to generate sodium caffeate. Finally, phenol was identified in *Cinnamomum zeylanicum*, which shares the same genus *Cinnamomum* as our target food, Chinese cinnamon (*Cinnamomum aromaticum*)²⁵⁵.

5.4 Discussion

In this work, we created an automated framework to extract information from literature and create domain-specific knowledgebase graphs. Applying to food and chemical relationships created the first AI-driven resource in the field, summarizing findings through 285,077 triplets, with 106,082 (2,091 high-, 94,095 medium-, and 9,896 low-quality) of those associations (46.0%) never been reported before in published databases

(Appendix A.3.1.9). While 98.2% of triplets from the Lit2KG pipeline were labeled as either medium or low quality (**Figure 5.2b**), our results indicate high performance for both the entailment model (medium-quality triplets; precision of 0.82) and the link prediction model (low-quality triplets; precision of 0.77). Additionally, both models exhibit strong calibration (R^2 of 0.94 and 0.99, respectively); that is, the model's predicted probabilities accurately reflect the likelihood of outcomes, providing reliability, interpretability, and better decision support. Surprisingly, in many cases, there are no indexed references associated with the reported entries and unique standardized IDs for the foods and compounds, which made reproducibility and provenance very difficult (**Appendix A.3.3.3** and **Supplementary Table 19**). FoodAtlas, by design, addresses this challenge by associating one or more references to each association.

Similarly, we are surprised that most of the associations that we have mined from the literature are not part of the existing databases, which argues that there is a plethora of information to be identified, validated, and integrated into tools like FoodAtlas. This, in turn, will be a boon for data-driven tools and pipelines for various applications, compound and source identification, product formulations, and other R&D operations that currently are serendipitous, error-prone, and time-consuming. Concomitantly, the food-chemical composition knowledge coverage of what is currently in various databases varies (22% of Frida, 35% of Phenol-Explorer, 61% of FDC, and 49% of FoodMine). There are two main reasons behind it. First, limitations to the NLP LitSense algorithms used by FoodAtlas may limit synonyms and exhaustive tagging of the various entities, co-occurrence of entities in windows that are further away in the text body, and information that is in tables, figures, or supplementary files²⁵⁶. Second, the lack of references that are

indexed and unique IDs for either foods or chemicals may introduce false positives. Further experimental validation of findings, such as the 11 novel associations with indirect evidence proposed by our link prediction pipeline, will help in accelerating the discovery and achieving completeness of the domain knowledge.

Large language models worked well in the entailment model but not for link prediction. We tested state-of-the-art language models like KG-BERT²⁵⁷ and KGLM¹⁹⁹ that have better MR metrics compared to the graph-embedding models and are generalizable to unseen entities or relations²⁵⁸. For example, we obtained an MR of 191 on the validation set by fine-tuning the KG-BERT architecture with the BioBERT as a pre-trained backbone instead of the BERT, which is a significant improvement over the RotatE MR of 1,139. However, those models were not used as other metrics were significantly worse than simpler algorithms like RotatE (MRR: 0.12, hits@1: 0.08, hits@3: 0.11, and hits@10: 0.18), and training/inference time was much longer, making it infeasible to perform proper hyperparameter optimization over our large-scale FAKG. In addition, while the GPT-3.5 performance was impressive even without refinement on domain-specific data, it was not on par with the FoodAtlas pipeline, and the lack of source reference IDs defeats the purpose of one of the main pillars behind FoodAtlas: providing high-quality, trustworthy information with evidence provenance.

We identified a conflicting food-chemical relationship from the link prediction generated hypotheses. In some cases, this supports FoodAtlas's potential to challenge established knowledge and emphasizes the necessity of experimental checking of the solid, established data. For instance, the established absence of beta-carotene synthesis in

Atlantic Cod (FDC food 171955) contrasts with a high probability score (0.84 ± 0.09) of the hypothesis (Atlantic Cod, contains, beta-carotene). We sought to reconcile this through literature validation, noting that while unused genes often degrade over time due to natural selection²⁵⁹, the Atlantic Cod retains the beta,beta-carotene 15,15'-dioxygenase-like enzyme gene. Nevertheless, our further investigation considered the Atlantic Cod's diet, particularly during its larval stage, which predominantly consists of crustaceans²⁶⁰ rich in beta-carotene²⁶¹. Thus, there exists a plausible dietary source for beta-carotene incorporation. Additionally, we discovered literature referencing the detection of beta-carotene in commercially processed cod liver oil, albeit the exact species of cod (Gadus morhua-Atlantic Cod or others like Gadus macrocephalus-Pacific Cod) was not specified²⁶².

Finally, we also tested GPT-3.5 and GPT-4 as a knowledge extractor instead of using them under the entailment settings. We fed chat completion GPT models an engineered prompt along with single example to extract knowledge from (see **Appendix A.3.1.10**). GPT-4 models excel knowledge extraction in all cases with F1 of 84.6% and 72.2% for benchmark datasets without and with concentration values, respectively (**Supplementary Table 20**), suggesting the direction towards next iteration of Lit2KG pipeline.

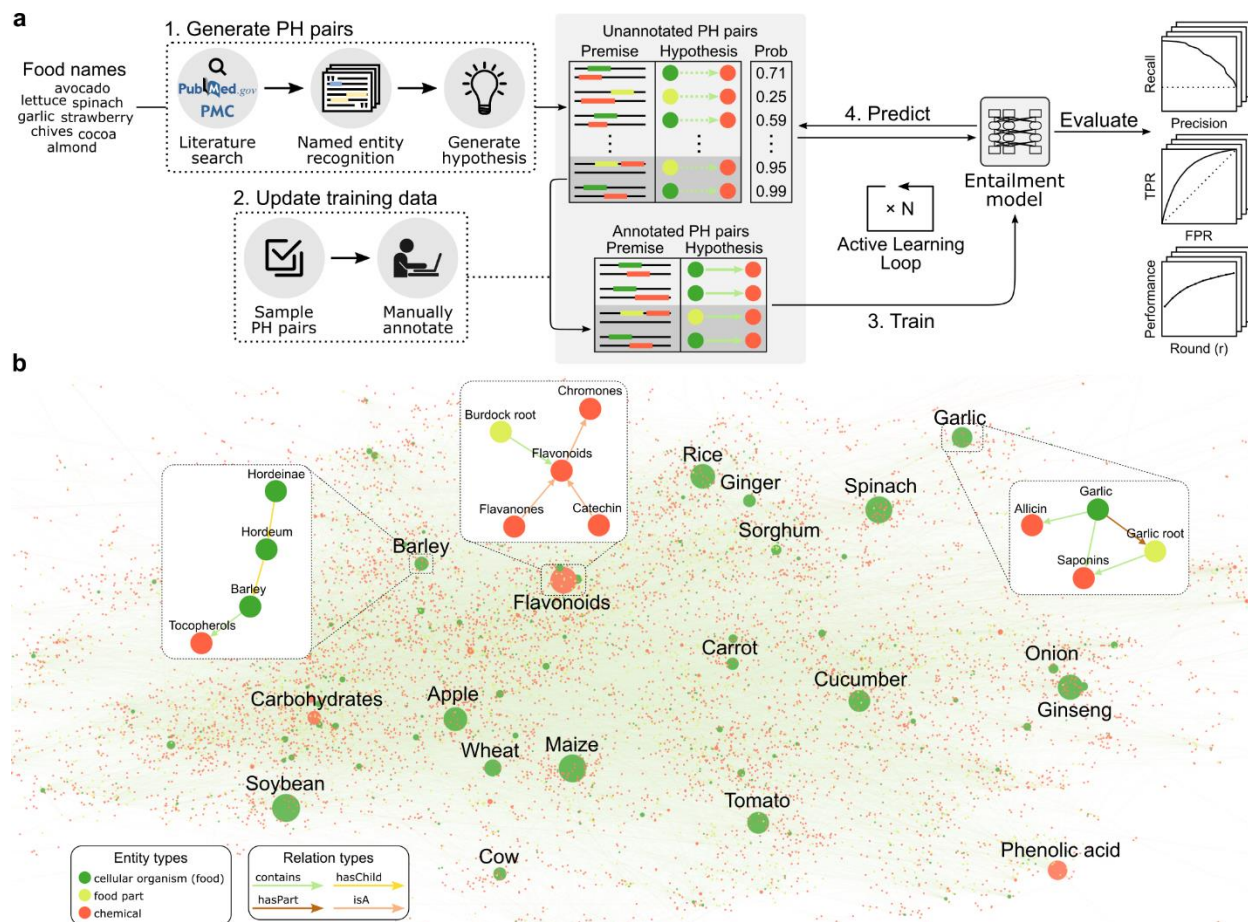


Figure 5.1. Overview of the Lit2KG framework and the FoodAtlas Knowledge Graph.

a Scientific literature is queried using raw food names and retrieved sentences (premises) where the species and chemical entities are tagged (e.g., ... cocoa_[SPECIES] is a good source of (-)-epicatechin_[CHEMICAL] ...). From these premises, hypothesis triplets are generated such as (*cocoa*, *contains*, (-)-epicatechin), which we refer to as premise-hypothesis (PH) pairs. The entailment model is then iteratively updated through active learning cycles, where a new batch of PH pairs is annotated in each cycle. Finally, both annotated and predicted positive PH pairs are used to populate the knowledge graph. **b** Visualization of the FoodAtlas Knowledge Graph (FAKG), which contains 285,077 triplets of 3 entity types and 4 relation types. Each triplet in the FAKG is assigned one of three quality types and provides a reference to the publications that support it for reproducibility.

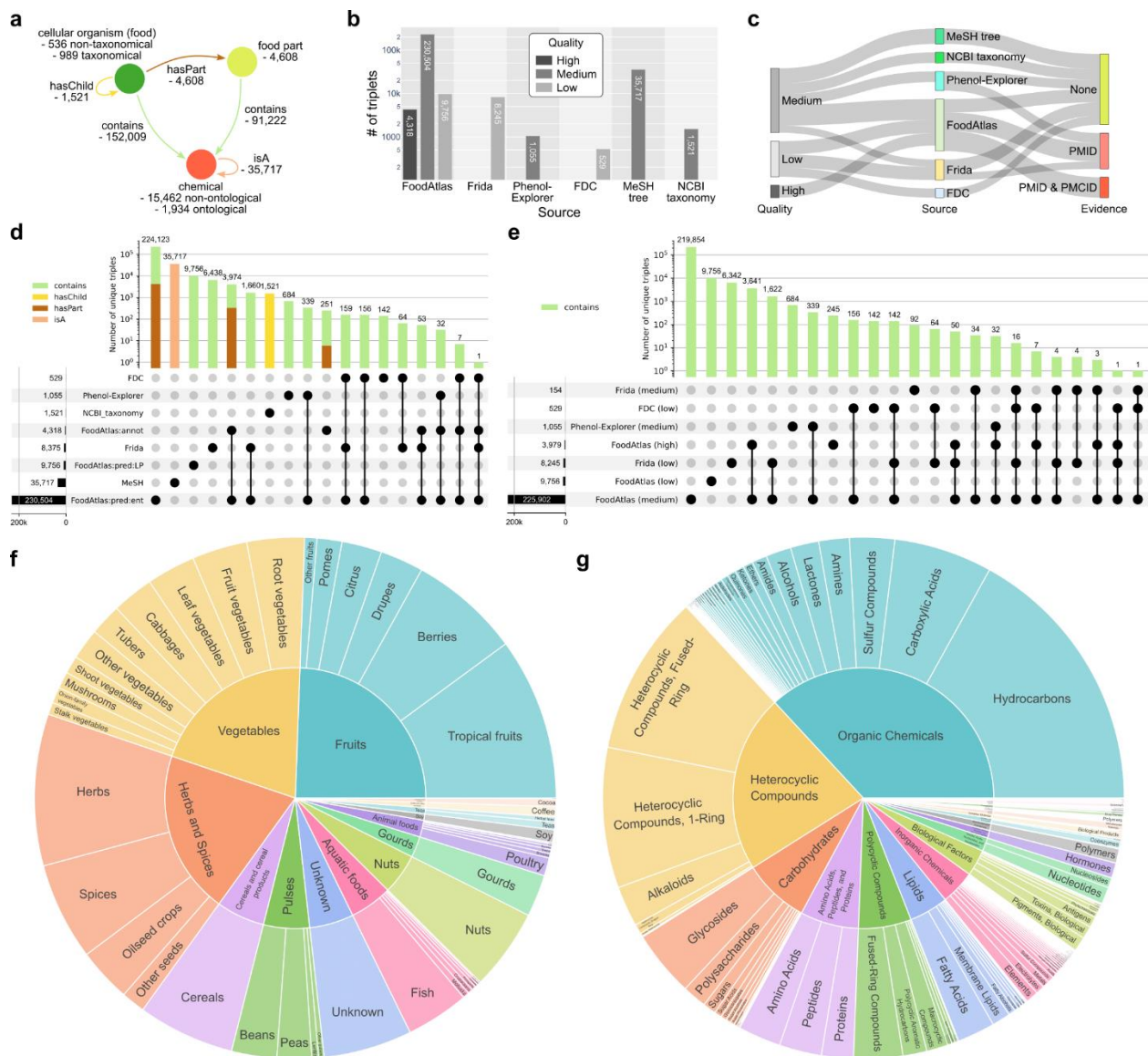


Figure 5.2. Statistics of the FoodAtlas Knowledge Graph. **a** Schema of the FAKG. The relation types *contains*, *hasPart*, *isA*, and *hasChild* encode the food-chemical composition relations, the food-food with part relations, the chemical ontological relations using the MeSH tree, and the taxonomical relations using the NCBI Taxonomy, respectively. **b** Number of triplets per data source in the FAKG depending on the quality. **c** Sankey graph showing the connections between quality, data source, and evidence. The thickness of the relations between the nodes represents the number of connections in the log scale. **d, e** UpSet plot showing the number of unique triplets for all data sources for all relation types and all sources based on quality for only the *contains* triplets. Each

row in the plot corresponds to a source, and the bar chart on the left shows the size of each source. Each column corresponds to an intersection, where the filled-in cells denote which source is part of an intersection. The bar chart for each column denotes the size of intersections. 'annot' stands for annotation, 'pred' stands for prediction, and 'LP' stands for link prediction. **f, g** Classification of foods and chemicals in FoodAtlas.

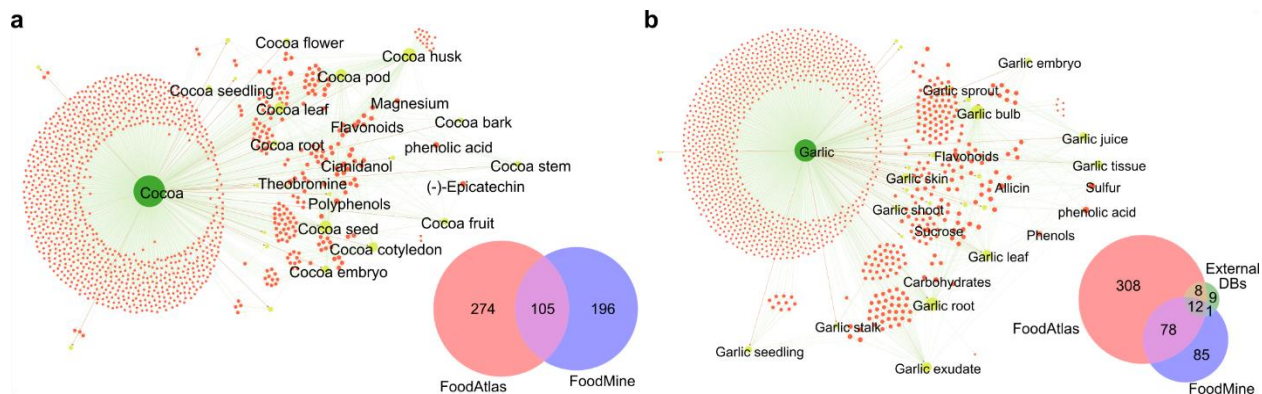


Figure 5.3. Results of comparing cocoa and garlic to the benchmark dataset FoodMine. a, b FoodAtlas subgraph of cocoa and garlic where whole food and food parts and their chemical composition are displayed. The label of the top 20 nodes with the largest degree is shown for each subgraph, and the size of the node is proportionate to its degree. The Venn diagram shows the overlap of FoodAtlas (entailment model annotation, entailment model prediction, and link prediction), external databases (Frida, Phenol-Explorer, and FDC), and FoodMine. Interestingly, none of the 3 external databases reported any chemical composition of cocoa.

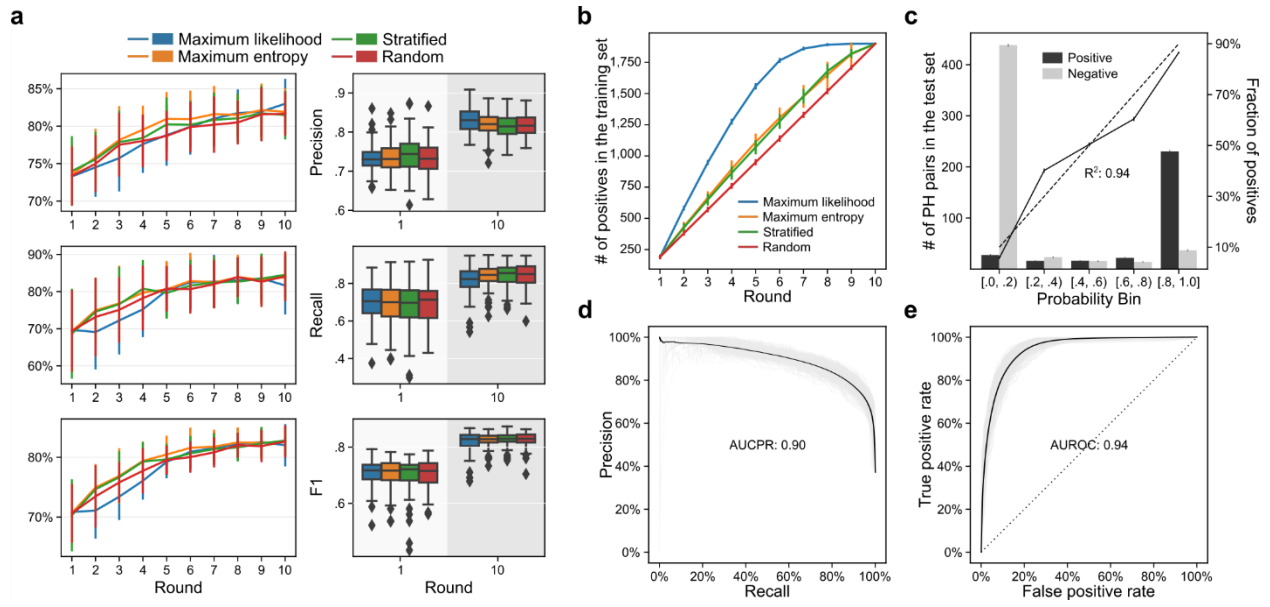


Figure 5.4. Prediction performance of the entailment model. **a** Precision, recall, and F1 score of the entailment models trained using the 4 different AL strategies for initial ($r = 1$) and final ($r = 10$) rounds ($n = 100$, 100 different random seeds). On the left, the line plot shows the mean value of each AL strategy, and the error lines denote the standard deviation of the 100 random seeds. On the right, the box represents the interquartile range, the middle line represents the median, the whisker line extends from minimum to maximum values, and the diamond represents outliers. **b** Comparison of the new knowledge discovery rate compared between the 4 AL strategies. The plot shows how early on in the AL round the 1,899 positive triplets within the simulated training pool of 4,120 triplets are discovered. The error line shows the standard deviation of the 100 random seeds. **c** Calibration plot showing a high correlation between the probability assigned by the entailment model and the ground-truth annotations on the test set ($R^2 = 0.94$). **d**, **e** The precision-recall and receiver operating characteristic curves of the entailment model predictions compared to the ground-truth annotations in the test set at the final round ($r = 10$) averaged over all 400 runs with a different random seed (100 runs for each of the 4 AL strategies).

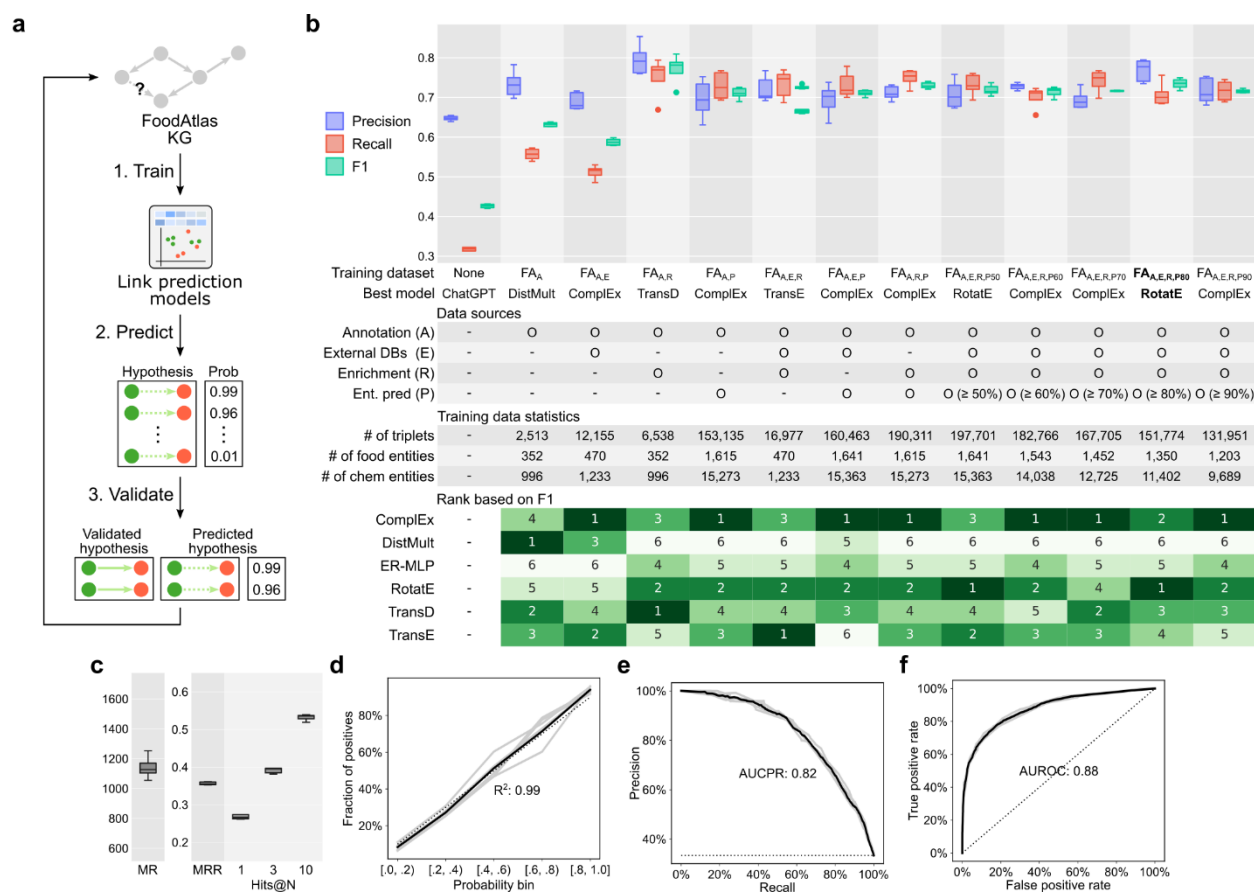


Figure 5.5. Link prediction model performance. **a** We use the FAKG to train a link prediction model whose objective is to generate hypotheses of type (food, contains, chemical) that is previously unknown in the graph. **b** Ablation study result showing the performance of 6 different link prediction models trained using 12 different versions of the FAKG, where different data sources were added or removed to understand their importance. While the training data is different for each version of the dataset, the validation and test set remain the same for fair comparison (positive to negative ratio is 1 to 2; baseline precision: 0.33, recall: 1.0, F1: 0.46). The best model for each dataset is selected based on the F1 score. The box represents the interquartile range, the middle line represents the median, the whisker line extends from minimum to maximum values, and the diamond represents outliers. **c** Standard rank-based metrics of the best model (RotatE) trained on the best training dataset (FA_{A,E,R,P80}). Lower is better for mean rank (MR), while higher is better for mean reciprocal rank (MRR), hits@1, hits@3, and hits@10. **d** Calibration plot showing a high correlation between the probability assigned by the link

prediction model and the ground-truth annotations on the test set ($n = 5$, 5 different random seeds). **e**, **f** Precision-recall and receiver operating characteristic curves of the best link prediction model.

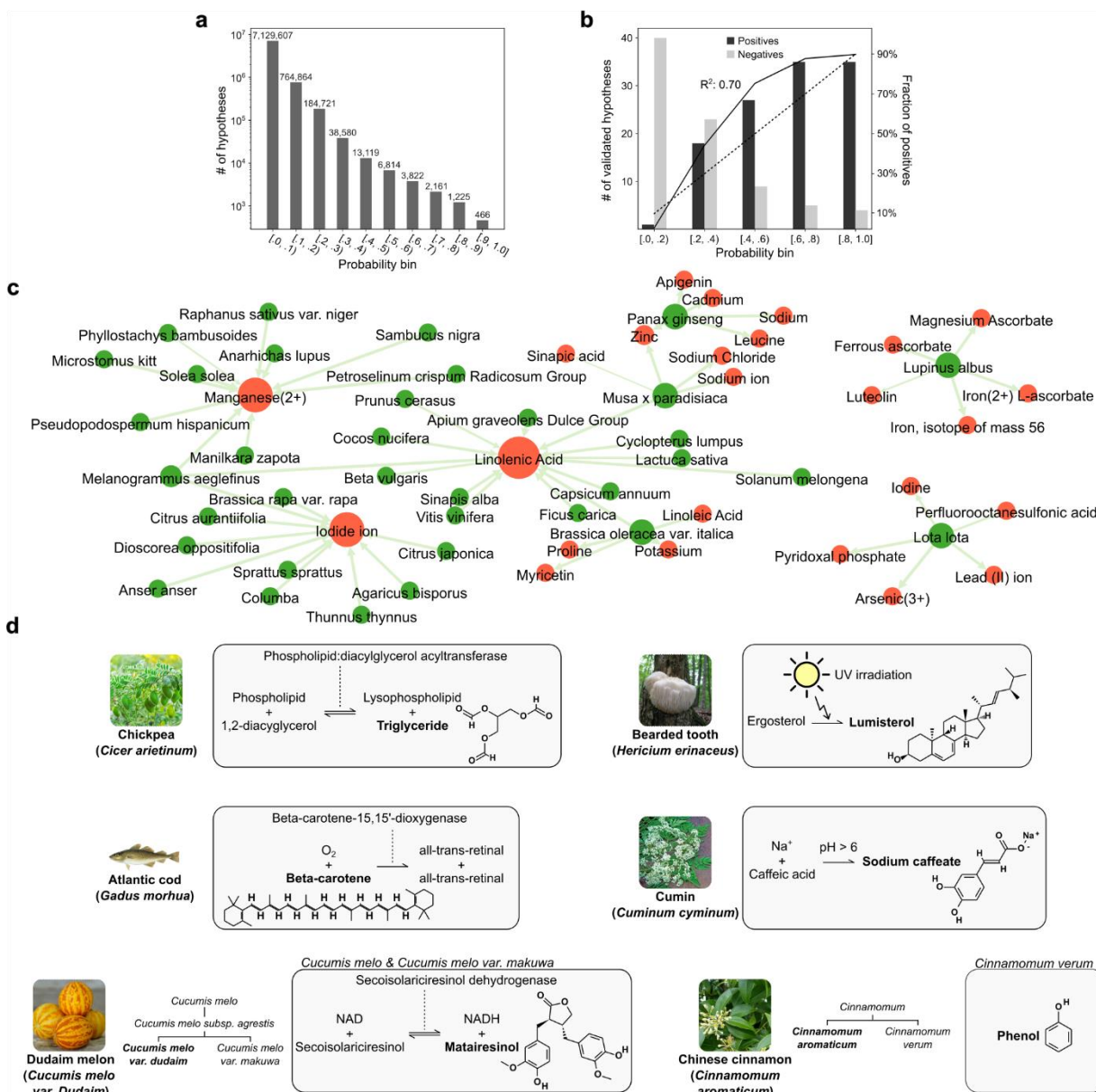


Figure 5.6. Validation of link prediction generated hypotheses. **a** Distribution of the 8,145,379 hypotheses in 10 equally spaced bins. **b** Calibration plot of the link prediction model based on randomly selected hypotheses (40 per bin) validated through manual literature search. **c** Visualization of positively validated link prediction hypotheses, where the 1-hop subgraph of the top 3 chemical and 5 food entities are shown. The edge width is proportionate to its probability score, and the size of the node is proportionate to its degree. **d** Indirect evidence for the 6 food-chemical relationships not found in the manual

literature search and suggested by the link prediction pipeline, where the food and chemical of interest are marked in bold.

Table 5.1. Comparison of the entailment model predicted premise-hypotheses pairs and the ground-truth annotation. The probability column shows the mean and standard deviation of the probability scores assigned to the corresponding PH pair at the final round (r = 10) of active learning by the 400 entailment models (100 random seeds each for 4 active learning strategies). GT stands for ground truth class assigned by the consensus of two annotators based on the premise. Samples shown in this table are from the test set.

Index	Premise	Hypothesis	Section	GT	Predicti on	Probab ility
1	Standardized extracts from the leaves of Ginkgo biloba contains 24 % ginkgo-flavone glycosides and 6 % terpenoids (ginkgolides , bilobalide). ²⁷³	(Ginkgo biloba – leaves, contains, ginkgolides)	Intro	Entails	Entails	99.6% ± 1.0 %
2	This Vaccinium myrtillus L extract is composed of flavonoids , and standardized to contain 36% anthocyanins, with conformance to the USP 31 on 'Powdered Bilberry Extract'. ²⁷⁴	(Bilberry, contains, flavonoids)	Methods	Entails	Entails	51.3% ± 33.7%
3	RYNXC consisted of 9 traditional Chinese herbs, including clove , rhubarb, frankincense, myrrh, borneol , rhizoma corydalis, cowherb seed , Rosae rugosae, Garden balsam stem. ²⁷⁵	(Clove – seed, contains, borneol)	Intro	Does not entail	Does not entail	0.3% ± 0.4%
4	For this purpose, tablets were produced containing 16 mg of ellagic acid with 100 mg of pulp from the fruit of an evergreen tree called Cherimoya, soursop, custard apple , and other common names (Annona muricata). ²⁷⁶	(Custard apple, contains, ellagic acid)	N/A	Does not entail	Does not entail	44.4% ± 37.0%
5	Previous investigations postulated that polyunsaturated fatty acids (PUFAs) are essential nutrients for the common octopus . ²⁵³	(Common octopus, contains, polyunsaturated fatty acids)	Intro	Does not entail	Entails	99.3% ± 1.0%
6	Domoic acid excretion in dungeness crabs , razor clams and mussels. ²⁷⁷	(Dungeness crabs, contains, Domoic acid)	Title	Does not entail	Entails	62.7% ± 34.4%

7	Antihyperlipidaemic and antihypercholesterolaemic effects of Anethum graveolens leaves after the removal of furocoumarins . ²⁷⁸	(Anethum graveolens, contains, furocoumarins)	Title	Entails	Does not entail	2.3% ± 8.0%
8	In the study by Keskiner et al. (2017), the patients in the test group received capsules containing 6.25 mg EPA and 19.19 mg DHA from Atlantic salmon (Vectomega tablet, Laboratoires Le Stum, Plage, France). ²⁷⁹	(Atlantic salmon, contains, DHA)	Results	Entails	Does not entail	44.7% ± 34.3%

Chapter 6

Conclusion

This dissertation deals with the application of machine learning in the domain of life sciences using a knowledge graph as its mode of input. In Chapter 2, the Learning Ontologies via Embeddings (LOVE) framework, which takes advantage of the semantic similarity of the word embeddings, was applied to the field of food ontologies. The automated method proposed here is a solution to the manual burden of populating an ontology with a continuous influx of new data. Therefore, the desired automation would be a semi-supervised method that yields high precision, with minimal manual intervention. In Chapter 3, a machine learning framework was proposed to automate knowledge discovery through knowledge graph construction, inconsistency resolution, and iterative link prediction. This work demonstrates how evidence-driven decisions are a step toward automating knowledge discovery with high confidence and accelerated pace, thereby substituting traditional time-consuming, and expensive methods. In Chapter 4, a novel Knowledge Graph Language Model (KGLM) architecture was proposed, where the new entity/relation embedding layer was introduced that learns to differentiate distinctive entity and relation types, therefore allowing the model to learn the structure of the knowledge graph. KGLM set a new state-of-the-art performance for the link prediction task on the benchmark datasets, therefore providing a step forward towards generalizable and highly accurate hypothesis generation models. Finally, in Chapter 5, the FoodAtlas pipeline to construct a large-scale knowledge graph using large language models in an active

learning setting was introduced. This work demonstrates how automated learning from literature at scale can accelerate discovery and support practical applications through reproducible, evidence-based capture of latent interactions of diverse entities, such as food and chemicals.

The profound impact of machine learning, exemplified by the advent of large language models such as GPT²⁸⁰, has ushered in a transformative era in the realm of life sciences discovery. The ability of these models to comprehend vast amounts of textual data, infer complex relationships, and generate meaningful insights has redefined the boundaries of scientific exploration. As we stand at the nexus of artificial intelligence and life sciences, the synergy between advanced machine learning algorithms and domain-specific knowledge promises unprecedented opportunities for accelerating discoveries, unveiling hidden patterns, and pushing the frontiers of our understanding. The integration of these technologies not only enhances the efficiency of data analysis but also sparks novel hypotheses, paving the way for innovative breakthroughs that have the potential to revolutionize the landscape of life sciences research. The journey from data to knowledge is now more dynamic and interconnected than ever, propelling us into a future where the collaborative efforts of human expertise and machine intelligence converge to unlock the mysteries of life.

References

1. Wang, Q., Mao, Z., Wang, B. & Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**, 2724–2743 (2017).
2. Introduction: What Is a Knowledge Graph? | SpringerLink. https://link.springer.com/chapter/10.1007/978-3-030-37439-6_1.
3. Cimiano, P. & Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* **8**, 489–508 (2017).
4. Wise, C. *et al.* COVID-19 Knowledge Graph: Accelerating Information Retrieval and Discovery for Scientific Literature. Preprint at <https://doi.org/10.48550/arXiv.2007.12731> (2020).
5. Reinanda, R., Meij, E. & Rijke, M. de. Knowledge Graphs: An Information Retrieval Perspective. *Found. Trends® Inf. Retr.* **14**, 289–444 (2020).
6. Wang, H., Zhao, M., Xie, X., Li, W. & Guo, M. Knowledge Graph Convolutional Networks for Recommender Systems. in *The World Wide Web Conference* 3307–3313 (Association for Computing Machinery, New York, NY, USA, 2019). doi:10.1145/3308558.3313417.
7. Guo, Q. *et al.* A Survey on Knowledge Graph-Based Recommender Systems. *IEEE Trans. Knowl. Data Eng.* **34**, 3549–3568 (2022).
8. Hammes, D., Medero, H. & Mitchell, H. Comparison of NoSQL and SQL Databases in the Cloud. *Proc. South. Assoc. Inf. Syst. SAIS Macon GA* 21–22 (2014).
9. Rodriguez, M. A. & Neubauer, P. Constructions from dots and lines. *Bull. Am. Soc. Inf. Sci. Technol.* **36**, 35–41 (2010).

10. Rodriguez, M. A. & Neubauer, P. The Graph Traversal Pattern. in *Graph Data Management: Techniques and Applications* 29–46 (IGI Global, 2012). doi:10.4018/978-1-61350-053-8.ch002.
11. Jia, B. *et al.* Pattern Discovery and Anomaly Detection via Knowledge Graph. in *2018 21st International Conference on Information Fusion (FUSION)* 2392–2399 (2018). doi:10.23919/ICIF.2018.8455737.
12. Chen, Z. *et al.* Knowledge Graph Completion: A Review. *IEEE Access* **8**, 192435–192456 (2020).
13. Bean, D. M. *et al.* Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. *Sci. Rep.* **7**, 16416 (2017).
14. Genesereth, M. R. & Nilsson, N. J. *Logical Foundations of Artificial Intelligence*. (Morgan Kaufmann, 2012).
15. Consortium, W. W. W. & others. RDF 1.1 concepts and abstract syntax. (2014).
16. Benslimane, D. *et al.* Contextual ontologies motivations, challenges, and solutions. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 4243 LNCS 168–176 (Springer Verlag, 2006).
17. Ehrlinger, L. & Wöß, W. *Towards a Definition of Knowledge Graphs*. *researchgate.net* (2016).
18. Li, Y. & Chen, L. Big biological data: challenges and opportunities. *Genomics Proteomics Bioinformatics* **12**, 187 (2014).
19. Silvescu, A., Caragea, D. & Atramentov, A. Graph databases. *Artif. Intell. Res. Lab. Dep. Comput. Sci. Iowa State Univ.* (2012).

20. Suchanek, F. M., Kasneci, G. & Weikum, G. Yago: a core of semantic knowledge. in *Proceedings of the 16th international conference on World Wide Web* 697–706 (Association for Computing Machinery, New York, NY, USA, 2007). doi:10.1145/1242572.1242667.
21. Mitchell, T. *et al.* Never-ending learning. *Commun. ACM* **61**, 103–115 (2018).
22. Bollacker, K., Evans, C., Paritosh, P., Sturge, T. & Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* 1247–1250 (2008).
23. Introducing the Knowledge Graph: things, not strings. *Google* <https://blog.google/products/search/introducing-knowledge-graph-things-not/> (2012).
24. Ernst, P., Siu, A. & Weikum, G. Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC Bioinformatics* **16**, 157 (2015).
25. Liekens, A. M. *et al.* BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biol.* **12**, R57 (2011).
26. Wishart, D. S. *et al.* DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* **46**, D1074–D1082 (2018).
27. Shen, Y. *et al.* KGDDS: A System for Drug-Drug Similarity Measure in Therapeutic Substitution based on Knowledge Graph Curation. *J. Med. Syst.* **43**, 92 (2019).
28. Davis, A. P. *et al.* Comparative Toxicogenomics Database (CTD): update 2023. *Nucleic Acids Res.* (2022) doi:10.1093/NAR/GKAC833.
29. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).

30. Wishart, D. S. *et al.* ChemFOnt: the chemical functional ontology resource. *Nucleic Acids Res.* (2022).
31. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29 (2000).
32. of Agriculture, A. R. S. FoodData Central, 2019.
33. Wishart, D. S. FooDB: the food database. Preprint at (2018).
34. GS1. GPC. Preprint at (2018).
35. (EFSA), E. F. S. A. The food classification and description system FoodEx 2 (revision 2). *EFSA Support. Publ.* **12**, 804E (2015).
36. Shinbo, Y. *et al.* KNApSAcK: a comprehensive species-metabolite relationship database. in *Plant metabolomics* 165–181 (Springer, 2006).
37. Rothwell, J. A. *et al.* Phenol-Explorer 3.0: a major update of the Phenol-Explorer database to incorporate data on the effects of food processing on polyphenol content. *Database* **2013**, bat070 (2013).
38. Eftimov, T., Ispirova, G., Potočnik, D., Ogrinc, N. & Seljak, B. K. ISO-FOOD ontology: A formal representation of the knowledge within the domain of isotopes for food science. *Food Chem.* **277**, 382–390 (2019).
39. Dooley, D. M. *et al.* FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. *Npj Sci. Food* **2**, 1–10 (2018).
40. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 1–9 (2016).

41. Kotova, E. E. & Pisarev, I. A. Automated Creation of Knowledge Bases for Intelligent Systems, taking into account Linguistic Uncertainty. in *2019 XXII International Conference on Soft Computing and Measurements (SCM)*) 149–152 (IEEE, 2019).
42. Xu, J. *et al.* Building a PubMed knowledge graph. *Sci. Data* **7**, 205 (2020).
43. Cenikj, G. *et al.* From language models to large-scale food and biomedical knowledge graphs. *Sci. Rep.* **13**, 7815 (2023).
44. Harnoune, A. *et al.* BERT based clinical knowledge extraction for biomedical knowledge graph construction and analysis. *Comput. Methods Programs Biomed. Update* **1**, 100042 (2021).
45. Diaz Gonzalez, A. D., Hughes, K. S., Yue, S. & Hayes, S. T. Applying BioBERT to Extract Germline Gene-Disease Associations for Building a Knowledge Graph from the Biomedical Literature. in *2023 the 7th International Conference on Information System and Data Mining (ICISDM)* 37–42 (ACM, Atlanta USA, 2023). doi:10.1145/3603765.3603771.
46. Dang, L. D., Phan, U. T. P. & Nguyen, N. T. H. GENA: A knowledge graph for nutrition and mental health. *J. Biomed. Inform.* **145**, 104460 (2023).
47. Hausmann, S. *et al.* FoodKG: A Semantics-Driven Knowledge Graph for Food Recommendation. in *The Semantic Web – ISWC 2019* (eds. Ghidini, C. *et al.*) vol. 11779 146–162 (Springer International Publishing, Cham, 2019).
48. Ahmad, Z. *et al.* Active Learning Based Relation Classification for Knowledge Graph Construction from Conversation Data. in *Neural Information Processing* (eds. Yang, H. *et al.*) 617–625 (Springer International Publishing, Cham, 2020). doi:10.1007/978-3-030-63820-7_70.

49. Sun, L. *et al.* ASRC: A Knowledge Graph Relation Construction Model based on Active Learning and Semantic Recognition. in *2022 IEEE International Conference on Big Data (Big Data)* 6025–6029 (2022). doi:10.1109/BigData55660.2022.10020502.
50. Ren, P. *et al.* MKGB: A Medical Knowledge Graph Construction Framework Based on Data Lake and Active Learning. in *Health Information Science* (eds. Siuly, S., Wang, H., Chen, L., Guo, Y. & Xing, C.) vol. 13079 245–253 (Springer International Publishing, Cham, 2021).
51. Bach, N. & Badaskar, S. A review of relation extraction. *Lit. Rev. Lang. Stat. II* **2**, 1–15 (2007).
52. Jiang, H. *et al.* Complex relation extraction: Challenges and opportunities. *ArXiv Prepr. ArXiv201204821* (2020).
53. Bosselut, A. *et al.* Comet: Commonsense transformers for automatic knowledge graph construction. *ArXiv Prepr. ArXiv190605317* (2019).
54. Wu, X. *et al.* Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. in *2019 IEEE International Conference on Data Mining (ICDM)* 1540–1545 (2019).
55. White, J. PubMed 2.0. *Med. Ref. Serv. Q.* **39**, 382–387 (2020).
56. Roberts, R. J. PubMed Central: The GenBank of the published literature. *Proc. Natl. Acad. Sci.* **98**, 381–382 (2001).
57. West, R. *et al.* Knowledge base completion via search-based question answering. in *Proceedings of the 23rd international conference on World wide web* 515–526 (2014).
58. Tamae, C. *et al.* Determination of antibiotic hypersensitivity among 4,000 single-gene-knockout mutants of *Escherichia coli*. *J. Bacteriol.* **190**, 5981–5988 (2008).

59. Palmieri, V. *et al.* The graphene oxide contradictory effects against human pathogens. *Nanotechnology* **28**, 152001 (2017).
60. Gebser, M. *et al.* Repair and prediction (under inconsistency) in large biological networks with answer set programming. in *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning* (2010).
61. Nickel, M., Murphy, K., Tresp, V. & Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proc. IEEE* **104**, 11–33 (2015).
62. Monnet, D. L. & Frimodt-Møller, N. Antimicrobial-drug use and methicillin-resistant *Staphylococcus aureus*. *Emerg. Infect. Dis.* **7**, 161–163 (2001).
63. Soo, V. W. C., Hanson-Manful, P. & Patrick, W. M. Artificial gene amplification reveals an abundance of promiscuous resistance determinants in *Escherichia coli*. *Proc. Natl. Acad. Sci.* **108**, 1484–1489 (2011).
64. Getoor, L. & Taskar, B. *Introduction to Statistical Relational Learning*. (MIT Press, 2007).
65. Lao, N. & Cohen, W. W. Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* **81**, 53–67 (2010).
66. Lao, N., Mitchell, T. & Cohen, W. Random walk inference and learning in a large scale knowledge base. in *Proceedings of the 2011 conference on empirical methods in natural language processing* 529–539 (2011).
67. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. & Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **26**, (2013).

68. Wang, Z., Zhang, J., Feng, J. & Chen, Z. Knowledge graph embedding by translating on hyperplanes. in *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 28 (2014).
69. Sun, Z., Deng, Z.-H., Nie, J.-Y. & Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. Preprint at <http://arxiv.org/abs/1902.10197> (2019).
70. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Prepr. ArXiv181004805* (2018).
71. Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. *ArXiv Prepr. ArXiv190711692* (2019).
72. Yao, L., Mao, C. & Luo, Y. KG-BERT: BERT for knowledge graph completion. *ArXiv Prepr. ArXiv190903193* (2019).
73. Zha, H., Chen, Z. & Yan, X. Inductive Relation Prediction by BERT. *ArXiv Prepr. ArXiv210307102* (2021).
74. Wang, B. *et al.* Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. in *Proceedings of the Web Conference 2021* 1737–1748 (2021).
75. Cucerzan, S. Large-scale named entity disambiguation based on Wikipedia data. in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* 708–716 (2007).

76. Youn, J., Naravane, T. & Tagkopoulos, I. Using Word Embeddings to Learn a Better Food Ontology. *Front. Artif. Intell.* **3**, (2020).
77. Guarino, N., Oberle, D. & Staab, S. What Is an Ontology? in *Handbook on Ontologies* (eds. Staab, S. & Studer, R.) 1–17 (Springer, Berlin, Heidelberg, 2009). doi:10.1007/978-3-540-92673-3_0.
78. Nickel, M. & Kiela, D. *Poincaré Embeddings for Learning Hierarchical Representations*. *papers.nips.cc*.
79. Mahmoud, N., Elbeh, H. & Abdlkader, H. M. Ontology Learning Based on Word Embeddings for Text Big Data Extraction. in *2018 14th International Computer Engineering Conference (ICENCO)* 183–188 (2018).
80. Jayawardana, V. *et al.* Semi-supervised instance population of an ontology using word vector embedding. in *2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer)* 1–7 (2017).
81. Barabási, A.-L., Menichetti, G. & Loscalzo, J. The unmapped chemical complexity of our diet. *Nat. Food* 1–5 (2019).
82. Reh, R. & Sojka, P. Software Framework for Topic Modelling with Large Corpora. in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* 45–50 (ELRA, Valletta, Malta, 2010).
83. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. in *Advances in neural information processing systems* 3111–3119 (2013).

84. Pennington, J., Socher, R. & Manning, C. D. Glove: Global vectors for word representation. in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* 1532–1543 (2014).
85. Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. Bag of tricks for efficient text classification. *ArXiv Prepr. ArXiv160701759* (2016).
86. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. *Enriching Word Vectors with Subword Information*. MIT Press.
87. Whetzel, P., Noy, N., ... N. S.-N. acids & 2011, undefined. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *academic.oup.com*.
88. Keet, C. M. A formal theory of granularity. *Free Univ. Bozen-Bolzano* (2008).
89. Noy, N. F., McGuinness, D. L. & others. Ontology development 101: A guide to creating your first ontology. Preprint at (2001).
90. Gangemi, A., Catenacci, C., Ciaramita, M. & Lehmann, J. Ontology evaluation and validation: an integrated formal model for the quality diagnostic task. *-Line Httpwww Loa-Cnr ItFilesOntoEval4OntoDevFinal Pdf* (2005).
91. Blanchard, E., Harzallah, M., Briand, H. & Kuntz, P. A Typology Of Ontology-Based Semantic Measures. *EMOI-INTEROP* **160**, (2005).
92. Maaten, L. van der & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
93. Toutanova, K. *et al. Representing Text for Joint Embedding of Text and Knowledge Bases*. *aclweb.org* (2015).

94. Grover, A. & Leskovec, J. node2vec: Scalable Feature Learning for Networks. *dl.acm.org* **13-17-August-2016**, 855–864 (2016).
95. Haussmann, S. *et al.* FoodKG: A Semantics-Driven Knowledge Graph for Food Recommendation. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 11779 LNCS 146–162 (Springer, 2019).
96. Niwattanakul, S., Singthongchai, J., Naenudorn, E. & Wanapu, S. Using of Jaccard coefficient for keywords similarity. in *Proceedings of the international multicongference of engineers and computer scientists* vol. 1 380–384 (2013).
97. Hamming, R. W. Error detecting and error correcting codes. *Bell Syst. Tech. J.* **29**, 147–160 (1950).
98. Youn, J., Rai, N. & Tagkopoulos, I. Knowledge integration and decision support for accelerated discovery of antibiotic resistance genes. *Nat. Commun.* **13**, 2360 (2022).
99. Maryam, L., Usmani, S. S. & Raghava, G. P. S. Computational resources in the management of antibiotic resistance: Speeding up drug discovery. *Drug Discov. Today* **26**, 2138–2151 (2021).
100. Gupta, S. K. *et al.* ARG-ANNOT, a new bioinformatic tool to discover antibiotic resistance genes in bacterial genomes. *Antimicrob. Agents Chemother.* **58**, 212–220 (2014).
101. Wang, X., Rai, N., Pereira, B. M. P., Eetemadi, A. & Tagkopoulos, I. Accelerated knowledge discovery from omics data by optimal experimental design. *Nat. Commun.* **11**, 1–9 (2020).

102. Kim, M. & Tagkopoulos, I. Data integration and predictive modeling methods for multi-omics datasets. *Mol. Omics* **14**, 8–25 (2018).
103. Kim, K.-J. & Tagkopoulos, I. Application of machine learning in rheumatic disease research. *Korean J. Intern. Med.* **34**, 708–722 (2019).
104. Barone, L., Williams, J. & Micklos, D. Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators. *PLoS Comput. Biol.* **13**, e1005755 (2017).
105. Li, Y. & Chen, L. Big biological data: challenges and opportunities. *Genomics Proteomics Bioinformatics* **12**, 187 (2014).
106. kumar Kaliyar, R. Graph databases: A survey. in *International Conference on Computing, Communication & Automation* 785–790 (2015).
107. da Silva, W. M. C., Werceles, P., Walter, M. E. M. T., Holanda, M. & Br\`igido, M. Graph databases in molecular biology. in *Brazilian Symposium on Bioinformatics* 50–57 (2018).
108. Fabregat, A. *et al.* Reactome graph database: Efficient access to complex pathway data. *PLoS Comput. Biol.* **14**, e1005968 (2018).
109. Silvescu, A., Caragea, D. & Atramentov, A. Graph databases. *Artif. Intell. Res. Lab. Dep. Comput. Sci. Iowa State Univ.* (2012).
110. Dumontier, M. *et al.* Bio2RDF release 3: a larger connected network of linked data for the life sciences. in *Proceedings of the 2014 International Conference on Posters & Demonstrations Track* vol. 1272 401–404 (2014).
111. Hasan, S. M. S. *et al.* Knowledge graph-enabled cancer data analytics. *IEEE J. Biomed. Health Inform.* **24**, 1952–1967 (2020).

112. Sheng, M. *et al.* CLMed: A Cross-lingual Knowledge Graph Framework for Cardiovascular Diseases. in *International Conference on Web Information Systems and Applications* 512–517 (2019).
113. Jia, B. *et al.* CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. *Nucleic Acids Res.* gkw1004 (2016).
114. Liu, B. & Pop, M. ARDB—antibiotic resistance genes database. *Nucleic Acids Res.* **37**, D443--D447 (2009).
115. Lakin, S. M. *et al.* MEGARes: an antimicrobial resistance database for high throughput sequencing. *Nucleic Acids Res.* **45**, D574--D580 (2016).
116. Scaria, J., Chandramouli, U. & Verma, S. K. Antibiotic Resistance Genes Online (ARGO): A Database on vancomycin and β -lactam resistance genes. *Bioinformatics* **1**, 5 (2005).
117. Nichols, R. J. *et al.* Phenotypic landscape of a bacterial cell. *Cell* **144**, 143–156 (2011).
118. Zhou, L., Lei, X.-H., Bochner, B. R. & Wanner, B. L. Phenotype microarray analysis of *Escherichia coli* K-12 mutants with deletions of all two-component systems. *J. Bacteriol.* **185**, 4956–4972 (2003).
119. Shaw, K. J. *et al.* Comparison of the changes in global gene expression of *Escherichia coli* induced by four bactericidal agents. *J. Mol. Microbiol. Biotechnol.* **5**, 105–122 (2003).
120. Louie, B., Mork, P., Martin-Sanchez, F., Halevy, A. & Tarczy-Hornoch, P. Data integration and genomic medicine. *J. Biomed. Inform.* **40**, 5–16 (2007).

121. Tshitoyan, V. *et al.* Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95–98 (2019).
122. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv Prepr. ArXiv181004805* (2018).
123. Begley, C. G. & Ioannidis, J. P. A. Reproducibility in science: improving the standard for basic and preclinical research. *Circ. Res.* **116**, 116–126 (2015).
124. McNutt, M. Journals unite for reproducibility. *Science* **346**, 679 (2014).
125. Anderson, N. R. *et al.* Issues in biomedical research data management and analysis: needs and barriers. *J. Am. Med. Inform. Assoc.* **14**, 478–488 (2007).
126. Skjærven, L., Yao, X.-Q., Scarabelli, G. & Grant, B. J. Integrating protein structural dynamics and evolutionary analysis with Bio3D. *BMC Bioinformatics* **15**, 399 (2014).
127. Organization, W. H. *Antimicrobial Resistance: Global Report on Surveillance*. (WHO Press, 2014).
128. Burnham, C.-A. D., Leeds, J., Nordmann, P., O’Grady, J. & Patel, J. Diagnosing antimicrobial resistance. *Nat. Rev. Microbiol.* **15**, 697 (2017).
129. Zankari, E. *et al.* Identification of acquired antimicrobial resistance genes. *J. Antimicrob. Chemother.* **67**, 2640–2644 (2012).
130. Arango-Argoty, G. *et al.* DeepARG: a deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome* **6**, 1–15 (2018).
131. Moradigaravand, D. *et al.* Prediction of antibiotic resistance in Escherichia coli from large-scale pan-genome data. *PLoS Comput. Biol.* **14**, e1006258 (2018).

132. Sang, S. *et al.* SemaTyP: a knowledge graph based literature mining method for drug discovery. *BMC Bioinformatics* **19**, 1–11 (2018).
133. Segler, M. & Waller, M. P. Chemical Discovery as a Knowledge Graph Completion Problem. *AITP 2017* (2017).
134. Hassani-Pak, K. & Rawlings, C. Knowledge discovery in biological databases for revealing candidate genes linked to complex phenotypes. *J. Integr. Bioinforma.* **14**, (2017).
135. Santos, A. *et al.* Clinical knowledge graph integrates proteomics data into clinical decision-making. *bioRxiv* (2020).
136. Jha, A., Khan, Y., Sahay, R. & d'Aquin, M. Metastatic Site Prediction in Breast Cancer using Omics Knowledge Graph and Pattern Mining with Kirchhoff's Law Traversal. *bioRxiv* (2020).
137. Nickel, M., Murphy, K., Tresp, V. & Gabrilovich, E. A review of relational machine learning for knowledge graphs. *Proc. IEEE* **104**, 11–33 (2016).
138. Quinlan, J. R. Learning logical definitions from relations. *Mach. Learn.* **5**, 239–266 (1990).
139. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. & Yakhnenko, O. Translating embeddings for modeling multi-relational data. in *Advances in neural information processing systems* 2787–2795 (2013).
140. Wang, Z., Zhang, J., Feng, J. & Chen, Z. Knowledge graph embedding by translating on hyperplanes. in *Twenty-Eighth AAAI conference on artificial intelligence* (2014).

141. Yao, L., Mao, C. & Luo, Y. KG-BERT: BERT for knowledge graph completion. *ArXiv Prepr. ArXiv190903193* (2019).
142. Wang, B. *et al.* Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. in *Proceedings of the Web Conference 2021* 1737–1748 (2021).
143. Keseler, I. M. *et al.* The EcoCyc database: reflecting new knowledge about Escherichia coli K-12. *Nucleic Acids Res.* **45**, D543--D550 (2016).
144. Tomasulo, P. ChemIDplus-super source for chemical and drug information. *Med. Ref. Serv. Q.* **21**, 53–59 (2002).
145. Pasternack, J. & Roth, D. Knowing what to believe (when you already know something). in *Proceedings of the 23rd International Conference on Computational Linguistics* 877–885 (2010).
146. Lao, N., Mitchell, T. & Cohen, W. W. Random walk inference and learning in a large scale knowledge base. in *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 529–539 (2011).
147. Socher, R., Chen, D., Manning, C. D. & Ng, A. Reasoning with neural tensor networks for knowledge base completion. in *Advances in neural information processing systems* 926–934 (2013).
148. Glorot, X., Bordes, A. & Bengio, Y. Deep sparse rectifier neural networks. in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* 315–323 (2011).

149. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
150. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *ArXiv Prepr. ArXiv14126980* (2014).
151. Freund, Y. & Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**, 119–139 (1997).
152. Dong, X. *et al.* Knowledge vault: A web-scale approach to probabilistic knowledge fusion. in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* 601–610 (2014).
153. Chawla, N. V, Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
154. Baba, T. *et al.* Construction of Escherichia coli K-12 in-frame, single-gene knockout mutants: the Keio collection. *Mol. Syst. Biol.* **2**, 8–2006 (2006).
155. Datsenko, K. A. & Wanner, B. L. One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products. *Proc. Natl. Acad. Sci.* **97**, 6640–6645 (2000).
156. Dingsdag, S. A. & Hunter, N. Metronidazole: an update on metabolism, structure-cytotoxicity and resistance mechanisms. *J. Antimicrob. Chemother.* **73**, 265–279 (2018).
157. Löfmark, S., Edlund, C. & Nord, C. E. Metronidazole is still the drug of choice for treatment of anaerobic infections. *Clin. Infect. Dis.* **50**, S16--S23 (2010).

158. Rodriguez, M. A. & Neubauer, P. A path algebra for multi-relational graphs. in *2011 IEEE 27th International Conference on Data Engineering Workshops* 128–131 (2011).
159. Consortium, G. O. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res.* **47**, D330--D338 (2019).
160. Liu, A. *et al.* Antibiotic sensitivity profiles determined with an Escherichia coli gene knockout collection: generating an antibiotic bar code. *Antimicrob. Agents Chemother.* **54**, 1393–1403 (2010).
161. Freund, Y., Schapire, R. & Abe, N. A short introduction to boosting. *J.-Jpn. Soc. Artif. Intell.* **14**, 1612 (1999).
162. Ji, G., He, S., Xu, L., Liu, K. & Zhao, J. Knowledge graph embedding via dynamic mapping matrix. in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* 687–696 (2015).
163. Kendall, M. G. A new measure of rank correlation. *Biometrika* **30**, 81–93 (1938).
164. Webber, W., Moffat, A. & Zobel, J. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst. TOIS* **28**, 1–38 (2010).
165. Samaluru, H., SaiSree, L. & Reddy, M. Role of SufI (FtsP) in cell division of Escherichia coli: evidence for its involvement in stabilizing the assembly of the divisome. *J. Bacteriol.* **189**, 8044–8052 (2007).
166. Ko, M. & Park, C. H-NS-dependent regulation of flagellar synthesis is mediated by a LysR family protein. *J. Bacteriol.* **182**, 4670–4672 (2000).
167. Krin, E., Danchin, A. & Soutourina, O. Decrypting the H-NS-dependent regulatory cascade of acid stress resistance in Escherichia coli. *BMC Microbiol.* **10**, 1–9 (2010).

168. Djoko, K. Y. *et al.* Interplay between tolerance mechanisms to copper and acid stress in *Escherichia coli*. *Proc. Natl. Acad. Sci.* **114**, 6818–6823 (2017).
169. Tani, T. H., Khodursky, A., Blumenthal, R. M., Brown, P. O. & Matthews, R. G. Adaptation to famine: a family of stationary-phase genes revealed by microarray analysis. *Proc. Natl. Acad. Sci.* **99**, 13471–13476 (2002).
170. Ferrario, M. *et al.* The leucine-responsive regulatory protein of *Escherichia coli* negatively regulates transcription of *ompC* and *micF* and positively regulates translation of *ompF*. *J. Bacteriol.* **177**, 103–113 (1995).
171. Gul, N. & Poolman, B. Functional reconstitution and osmoregulatory properties of the ProU ABC transporter from *Escherichia coli*. *Mol. Membr. Biol.* **30**, 138–148 (2013).
172. Kim, I.-K. *et al.* Crystal structure of a new type of NADPH-dependent quinone oxidoreductase (QOR2) from *Escherichia coli*. *J. Mol. Biol.* **379**, 372–384 (2008).
173. Piek, S. *et al.* The role of oxidoreductases in determining the function of the neisserial lipid A phosphoethanolamine transferase required for resistance to polymyxin. *PLoS One* **9**, e106513 (2014).
174. Al Mamun, A. A. M. *et al.* Identity and function of a large gene network underlying mutagenic repair of DNA breaks. *Science* **338**, 1344–1348 (2012).
175. Zhao, X. & Lam, J. S. WaaP of *Pseudomonas aeruginosa* is a novel eukaryotic type protein-tyrosine kinase as well as a sugar kinase essential for the biosynthesis of core lipopolysaccharide. *J. Biol. Chem.* **277**, 4722–4730 (2002).

176. Yethon, J. A. *et al.* Salmonella enterica Serovar Typhimurium waaP Mutants Show Increased Susceptibility to Polymyxin and Loss of Virulence In Vivo. *Infect. Immun.* **68**, 4485–4491 (2000).
177. Alcock, B. P. *et al.* CARD 2020: antibiotic resistance surveillance with the comprehensive antibiotic resistance database. *Nucleic Acids Res.* **48**, D517--D525 (2020).
178. Merchel Piovesan Pereira, B., Wang, X. & Tagkopoulos, I. Biocide-Induced Emergence of Antibiotic Resistance in Escherichia coli. *Front. Microbiol.* **12**, 335 (2021).
179. Toutanova, K. & Chen, D. Observed versus latent features for knowledge base and text inference. in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality* 57–66 (2015).
180. Dettmers, T., Minervini, P., Stenetorp, P. & Riedel, S. Convolutional 2d knowledge graph embeddings. in *Thirty-second AAAI conference on artificial intelligence* (2018).
181. Feng, J. *et al.* Knowledge graph embedding by flexible translation. in *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning* (2016).
182. Wang, M., Qiu, L. & Wang, X. A Survey on Knowledge Graph Embeddings for Link Prediction. *Symmetry* **13**, 485 (2021).
183. Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. *ArXiv Prepr. ArXiv190711692* (2019).
184. Sun, Z., Deng, Z.-H., Nie, J.-Y. & Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *ArXiv Prepr. ArXiv190210197* (2019).

185. Yang, B., Yih, W., He, X., Gao, J. & Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *ArXiv Prepr. ArXiv14126575* (2014).
186. Wang, Q., Mao, Z., Wang, B. & Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**, 2724–2743 (2017).
187. Ji, S., Pan, S., Cambria, E., Marttinen, P. & Philip, S. Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* (2021).
188. Lee, J. *et al.* BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **36**, 1234–1240 (2020).
189. Yeh, P., Tschumi, A. I. & Kishony, R. Functional classification of drugs by properties of their pairwise interactions. *Nat. Genet.* **38**, 489 (2006).
190. Suzuki, S., Horinouchi, T. & Furusawa, C. Prediction of antibiotic resistance by gene expression profiles. *Nat. Commun.* **5**, 5792 (2014).
191. Weiss, S. J., Mansell, T. J., Mortazavi, P., Knight, R. & Gill, R. T. Parallel mapping of antibiotic resistance alleles in *Escherichia coli*. *PloS One* **11**, e0146916 (2016).
192. Cho, K. *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv Prepr. ArXiv14061078* (2014).
193. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
194. Sybrandt, J., Tyagin, I., Shtutman, M. & Safro, I. AGATHA: Automatic Graph Mining And Transformer based Hypothesis Generation Approach. in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* 2757–2764 (2020).

195. Raccuglia, P. *et al.* Machine-learning-assisted materials discovery using failed experiments. *Nature* **533**, 73–76 (2016).
196. Gunning, D. Explainable artificial intelligence (xai). *Def. Adv. Res. Proj. Agency DARPA Nd Web* **2**, (2017).
197. Holzinger, A., Biemann, C., Pattichis, C. S. & Kell, D. B. What do we need to build explainable AI systems for the medical domain? *ArXiv Prepr. ArXiv171209923* (2017).
198. Huynh, L., Tsoukalas, A., Köppe, M. & Tagkopoulos, I. SBROME: a scalable optimization and module matching framework for automated biosystems design. *ACS Synth. Biol.* **2**, 263–273 (2013).
199. Youn, J. & Tagkopoulos, I. KGLM: Integrating Knowledge Graph Structure in Language Models for Link Prediction. in *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)* (eds. Palmer, A. & Camacho-collados, J.) 217–224 (Association for Computational Linguistics, Toronto, Canada, 2023). doi:10.18653/v1/2023.starsem-1.20.
200. Shen, J., Wang, C., Gong, L. & Song, D. Joint Language Semantic and Structure Embedding for Knowledge Graph Completion. Preprint at <https://doi.org/10.48550/arXiv.2209.08721> (2022).
201. Rossi, A., Barbosa, D., Firmani, D., Matinata, A. & Merialdo, P. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data TKDD* **15**, 1–49 (2021).
202. Dettmers, T., Minervini, P., Stenetorp, P. & Riedel, S. Convolutional 2d knowledge graph embeddings. in *Thirty-second AAAI conference on artificial intelligence* (2018).
203. Miller, G. A. *WordNet: An Electronic Lexical Database*. (MIT press, 1998).

204. Toutanova, K. & Chen, D. Observed versus latent features for knowledge base and text inference. in *Proceedings of the 3rd workshop on continuous vector space models and their compositionality* 57–66 (2015).
205. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. *ArXiv Prepr. ArXiv171105101* (2017).
206. Rasley, J., Rajbhandari, S., Ruwase, O. & He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 3505–3506 (2020).
207. Rajbhandari, S., Rasley, J., Ruwase, O. & He, Y. Zero: Memory optimizations toward training trillion parameter models. in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis* 1–16 (IEEE, 2020).
208. Wolf, T. *et al.* Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv Prepr. ArXiv191003771* (2019).
209. Barabási, A.-L., Menichetti, G. & Loscalzo, J. The unmapped chemical complexity of our diet. *Nat. Food* **1**, 33–37 (2020).
210. Elmadfa, I. & Meyer, A. L. Importance of food composition data to nutrition and public health. *Eur. J. Clin. Nutr.* **64**, S4–S7 (2010).
211. Diana, M., Quílez, J. & Rafecas, M. Gamma-aminobutyric acid as a bioactive compound in foods: a review. *J. Funct. Foods* **10**, 407–420 (2014).
212. Reboledo-Rodríguez, P. *et al.* State of the Art on Functional Virgin Olive Oils Enriched with Bioactive Compounds and Their Properties. *Int. J. Mol. Sci.* **18**, 668 (2017).

213. Gan, J., Siegel, J. B. & German, J. B. Molecular annotation of food – Towards personalized diet and precision health. *Trends Food Sci. Technol.* **91**, 675–680 (2019).
214. Eetemadi, A. *et al.* The Computational Diet: A Review of Computational Methods Across Diet, Microbiome, and Health. *Front. Microbiol.* **11**, (2020).
215. Eetemadi, A. & Tagkopoulos, I. Methane and fatty acid metabolism pathways are predictive of Low-FODMAP diet efficacy for patients with irritable bowel syndrome. *Clin. Nutr. Edinb. Scotl.* **40**, 4414–4421 (2021).
216. McKillop, K., Harnly, J., Pehrsson, P., Fukagawa, N. & Finley, J. FoodData Central, USDA's Updated Approach to Food Composition Data Systems. *Curr. Dev. Nutr.* **5**, 596 (2021).
217. Anses. Ciquel French food composition table. (2020).
218. Kapsokefalou, M. *et al.* Food Composition at Present: New Challenges. *Nutrients* **11**, 1714 (2019).
219. Scalbert, A. *et al.* The food metabolome: a window over dietary exposure. *Am. J. Clin. Nutr.* **99**, 1286–1308 (2014).
220. Wishart, D. FooDB Version 1.0.
221. Rakhi, N. K., Tuwani, R., Mukherjee, J. & Bagler, G. Data-driven analysis of biomedical literature suggests broad-spectrum benefits of culinary herbs and spices. *PLOS ONE* **13**, e0198030 (2018).
222. Afendi, F. M. *et al.* KNApSACk Family Databases: Integrated Metabolite–Plant Species Databases for Multifaceted Plant Research. *Plant Cell Physiol.* **53**, e1 (2012).
223. Duke, J. & Bogenschutz, M. J. *Dr. Duke's Phytochemical and Ethnobotanical Databases.* (USDA, Agricultural Research Service Washington, DC, 1994).

224. Neveu, V. *et al.* Phenol-Explorer: an online comprehensive database on polyphenol contents in foods. *Database* **2010**, bap024 (2010).
225. Rothwell, J. A. *et al.* Phenol-Explorer 2.0: a major update of the Phenol-Explorer database integrating data on polyphenol metabolism and pharmacokinetics in humans and experimental animals. *Database* **2012**, bas031 (2012).
226. Silva, A. B. da *et al.* PhytoHub V1.4: A new release for the online database dedicated to food phytochemicals and their human metabolites. in np (2016).
227. Schoch, C. L. *et al.* NCBI Taxonomy: a comprehensive update on curation, resources and tools. *Database J. Biol. Databases Curation* **2020**, baaa062 (2020).
228. Allot, A. *et al.* LitSense: making sense of biomedical literature at sentence level. *Nucleic Acids Res.* **47**, W594–W599 (2019).
229. Lee, J. *et al.* BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* btz682 (2019) doi:10.1093/bioinformatics/btz682.
230. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Preprint at <https://doi.org/10.48550/arXiv.1810.04805> (2019).
231. Liu, Y. *et al.* RoBERTa: A Robustly Optimized BERT Pretraining Approach. Preprint at <https://doi.org/10.48550/arXiv.1907.11692> (2019).
232. Brown, T. *et al.* Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
233. National Food Institute, Technical University of Denmark. Food data (frida.fooddata.dk), version 4.2, 2022.

234. Rossi, A., Barbosa, D., Firmani, D., Matinata, A. & Merialdo, P. Knowledge Graph Embedding for Link Prediction: A Comparative Analysis. *ACM Trans. Knowl. Discov. Data* **15**, 14:1-14:49 (2021).
235. Ali, M. *et al.* PyKEEN 1.0: a Python library for training and evaluating knowledge graph embeddings. *J. Mach. Learn. Res.* **22**, 82:3723-82:3728 (2021).
236. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. & Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. in *Advances in Neural Information Processing Systems* vol. 26 (Curran Associates, Inc., 2013).
237. Dong, X. *et al.* Knowledge vault: a web-scale approach to probabilistic knowledge fusion. in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* 601–610 (Association for Computing Machinery, New York, NY, USA, 2014). doi:10.1145/2623330.2623623.
238. Yang, B., Yih, W., He, X., Gao, J. & Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. Preprint at <https://doi.org/10.48550/arXiv.1412.6575> (2015).
239. Ji, G., He, S., Xu, L., Liu, K. & Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* 687–696 (Association for Computational Linguistics, Beijing, China, 2015). doi:10.3115/v1/P15-1067.
240. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E. & Bouchard, G. Complex Embeddings for Simple Link Prediction. in *Proceedings of The 33rd International Conference on Machine Learning* 2071–2080 (PMLR, 2016).

241. Youn, J. & Tagkopoulos, I. KGLM: Integrating Knowledge Graph Structure in Language Models for Link Prediction. Preprint at <https://doi.org/10.48550/arXiv.2211.02744> (2022).
242. Kim, S. *et al.* PubChem Protein, Gene, Pathway, and Taxonomy Data Collections: Bridging Biology and Chemistry through Target-Centric Views of PubChem Data. *J. Mol. Biol.* **434**, 167514 (2022).
243. Hooton, F., Menichetti, G. & Barabási, A.-L. Exploring food contents in scientific literature with FoodMine. *Sci. Rep.* **10**, 16191 (2020).
244. Reifen, R., Karlinsky, A., Stark, A. H., Berkovich, Z. & Nyska, A. α -Linolenic acid (ALA) is an anti-inflammatory agent in inflammatory bowel disease. *J. Nutr. Biochem.* **26**, 1632–1640 (2015).
245. Singer, P. *et al.* Effects of dietary oleic, linoleic and alpha-linolenic acids on blood pressure, serum lipids, lipoproteins and the formation of eicosanoid precursors in patients with mild essential hypertension. *J. Hum. Hypertens.* **4**, 227–233 (1990).
246. Lawrence, G. D. & Sawyer, D. T. The chemistry of biological manganese. *Coord. Chem. Rev.* **27**, 173–193 (1978).
247. Schramm, V. L. *Manganese in Metabolism and Enzyme Function*. (Elsevier, 2012).
248. Aguirre, J. D. & Culotta, V. C. Battles with Iron: Manganese in Oxidative Stress Protection *. *J. Biol. Chem.* **287**, 13541–13548 (2012).
249. Bagnato, C. *et al.* Analysis of triglyceride synthesis unveils a green algal soluble diacylglycerol acyltransferase and provides clues to potential enzymatic components of the chloroplast pathway. *BMC Genomics* **18**, 223 (2017).

250. Nagao, A., Maeda, M., Lim, B. P., Kobayashi, H. & Terao, J. Inhibition of β -carotene-15,15'-dioxygenase activity by dietary flavonoids. *J. Nutr. Biochem.* **11**, 348–355 (2000).
251. Garcia-Mas, J. *et al.* The genome of melon (*Cucumis melo* L.). *Proc. Natl. Acad. Sci.* **109**, 11872–11877 (2012).
252. Joradon, P. *et al.* Ergosterol Content and Antioxidant Activity of Lion's Mane Mushroom (*Hericium erinaceus*) and Its Induction to Vitamin D2 by UVC-Irradiation: in *Proceedings of the 8th International Conference on Agricultural and Biological Sciences* 19–28 (SCITEPRESS - Science and Technology Publications, Shenzhen, China, 2022). doi:10.5220/0011594600003430.
253. Monroig, Ó. *et al.* Biosynthesis of Polyunsaturated Fatty Acids in *Octopus vulgaris*: Molecular Cloning and Functional Characterisation of a Stearoyl-CoA Desaturase and an Elongation of Very Long-Chain Fatty Acid 4 Protein. *Mar. Drugs* **15**, 82 (2017).
254. Shamsiev, A., Park, J., Olawuyi, I. F., Odey, G. & Lee, W. Optimization of ultrasonic-assisted extraction of polyphenols and antioxidants from cumin (*Cuminum cyminum* L.). *Korean J. Food Preserv.* **28**, 510–521 (2021).
255. Senanayake, U. M., Lee, T. H. & Wills, R. B. H. Volatile constituents of cinnamon (*Cinnamomum zeylanicum*) oils. *J. Agric. Food Chem.* **26**, 822–824 (1978).
256. Herzig, J., Nowak, P. K., Müller, T., Piccinno, F. & Eisenschlos, J. M. TAPAS: Weakly Supervised Table Parsing via Pre-training. in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* 4320–4333 (2020). doi:10.18653/v1/2020.acl-main.398.

257. Yao, L., Mao, C. & Luo, Y. KG-BERT: BERT for Knowledge Graph Completion. Preprint at <https://doi.org/10.48550/arXiv.1909.03193> (2019).
258. Zha, H., Chen, Z. & Yan, X. Inductive Relation Prediction by BERT. *Proc. AAAI Conf. Artif. Intell.* **36**, 5923–5931 (2022).
259. Albalat, R. & Cañestro, C. Evolution by gene loss. *Nat. Rev. Genet.* **17**, 379–391 (2016).
260. Hamre, K. Nutrition in cod (*Gadus morhua*) larvae and juveniles. *ICES J. Mar. Sci.* **63**, 267–274 (2006).
261. Maoka, T. Carotenoids in Marine Animals. *Mar. Drugs* **9**, 278–293 (2011).
262. Luterotti, S., Franko, M. & Bicanic, D. Ultrasensitive determination of β -carotene in fish oil-based supplementary drugs by HPLC-TLS. *J. Pharm. Biomed. Anal.* **21**, 901–909 (1999).
263. Crozier, A., Jaganath, I. B. & Clifford, M. N. Dietary phenolics: chemistry, bioavailability and effects on health. *Nat. Prod. Rep.* **26**, 1001–1043 (2009).
264. Kyngäs, H., Kääriäinen, M. & Elo, S. The Trustworthiness of Content Analysis. in *The Application of Content Analysis in Nursing Science Research* (eds. Kyngäs, H., Mikkonen, K. & Kääriäinen, M.) 41–48 (Springer International Publishing, Cham, 2020). doi:10.1007/978-3-030-30199-6_5.
265. Dooley, D. M. *et al.* FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. *Npj Sci. Food* **2**, 23 (2018).
266. Bodenreider, O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.* **32**, D267–D270 (2004).

267. Min, W., Liu, C., Xu, L. & Jiang, S. Applications of knowledge graphs for food science and industry. *Patterns* **3**, 100484 (2022).
268. Ławrynowicz, A., Wróblewska, A., Adrian, W. T., Kulczyński, B. & Gramza-Michałowska, A. Food Recipe Ingredient Substitution Ontology Design Pattern. *Sensors* **22**, 1095 (2022).
269. Chen, Y., Subburathinam, A., Chen, C.-H. & Zaki, M. J. Personalized Food Recommendation as Constrained Question Answering over a Large-scale Food Knowledge Graph. in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* 544–552 (ACM, Virtual Event Israel, 2021). doi:10.1145/3437963.3441816.
270. Degtyarenko, K. *et al.* ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res.* **36**, D344–D350 (2008).
271. Rodrigues, F. A. Network Centrality: An Introduction. in *A Mathematical Modeling Approach from Nonlinear Dynamics to Complex Systems* (ed. Macau, E. E. N.) 177–196 (Springer International Publishing, Cham, 2019). doi:10.1007/978-3-319-78512-7_10.
272. Wagner, G. P., Pavlicev, M. & Cheverud, J. M. The road to modularity. *Nat. Rev. Genet.* **8**, 921–931 (2007).
273. Abdel-Salam, O. M. E. *et al.* Cannabis-induced impairment of learning and memory: effect of different nootropic drugs. *EXCLI J.* **12**, 193–214 (2013).
274. Steigerwalt, R. D. *et al.* Mirtogenol potentiates latanoprost in lowering intraocular pressure and improves ocular blood flow in asymptomatic subjects. *Clin. Ophthalmol. Auckl. NZ* **4**, 471–476 (2010).

275. Zhang, G. *et al.* Therapeutic Efficiency of an External Chinese Herbal Formula of Mammary Precancerous Lesions by BATMAN-TCM Online Bioinformatics Analysis Tool and Experimental Validation. *Evid.-Based Complement. Altern. Med. ECAM* **2019**, 2795010 (2019).
276. Bernier, C., Goetz, C., Jubinville, E. & Jean, J. The New Face of Berries: A Review of Their Antiviral Properties. *Foods* **11**, 102 (2021).
277. Schultz, I. R., Skillman, A. & Woodruff, D. Domoic acid excretion in dungeness crabs, razor clams and mussels. *Mar. Environ. Res.* **66**, 21–23 (2008).
278. Yazdanparast, R. & Alavi, M. Antihyperlipidaemic and antihypercholesterolaemic effects of *Anethum graveolens* leaves after the removal of furocoumarins. *Cytobios* **105**, 185–191 (2001).
279. Kruse, A. B. *et al.* What is the impact of the adjunctive use of omega-3 fatty acids in the treatment of periodontitis? A systematic review and meta-analysis. *Lipids Health Dis.* **19**, 100 (2020).
280. Floridi, L. & Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.* **30**, 681–694 (2020).
281. Consortium, G. O. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res.* **45**, D331--D338 (2016).
282. Kitagawa, M. *et al.* Complete set of ORF clones of *Escherichia coli* ASKA library (A Complete Set of *E. coli* K-12 ORF Archive): Unique Resources for Biological Research. *DNA Res.* **12**, 291–299 (2005).

283. Fang, X. *et al.* Global transcriptional regulatory network for *Escherichia coli* robustly connects gene expression to transcription factor activities. *Proc. Natl. Acad. Sci.* **114**, 10286–10291 (2017).
284. Girgis, H. S., Hottes, A. K. & Tavazoie, S. Genetic architecture of intrinsic antibiotic susceptibility. *PloS One* **4**, e5629 (2009).
285. Feunang, Y. D. *et al.* ClassyFire: automated chemical classification with a comprehensive, computable taxonomy. *J. Cheminformatics* **8**, 61 (2016).
286. Bollacker, K., Evans, C., Paritosh, P., Sturge, T. & Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* 1247–1250 (2008).
287. Li, Y. *et al.* A survey on truth discovery. *ACM Sigkdd Explor. Newsl.* **17**, 1–16 (2016).
288. Li, X., Dong, X. L., Lyons, K., Meng, W. & Srivastava, D. Truth finding on the deep web: Is the problem solved? *Proc. VLDB Endow.* **6**, 97–108 (2012).
289. Melas, I. N., Samaga, R., Alexopoulos, L. G. & Klamt, S. Detecting and removing inconsistencies between experimental data and signaling network topologies using integer linear programming on interaction graphs. *PLoS Comput. Biol.* **9**, e1003204 (2013).
290. Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *J. ACM JACM* **46**, 604–632 (1999).
291. Yin, X., Han, J. & Philip, S. Y. Truth discovery with multiple conflicting information providers on the web. *IEEE Trans. Knowl. Data Eng.* **20**, 796–808 (2008).

292. Liekens, A. M. L. *et al.* BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biol.* **12**, R57 (2011).
293. Pecina, P. Lexical association measures and collocation extraction. *Lang. Resour. Eval.* **44**, 137–158 (2010).
294. Carlson, A. *et al.* Toward an architecture for never-ending language learning. in *Twenty-Fourth AAAI Conference on Artificial Intelligence* (2010).
295. Ding, X., Zhang, Y., Liu, T. & Duan, J. Deep learning for event-driven stock prediction. in *Twenty-Fourth International Joint Conference on Artificial Intelligence* (2015).
296. Nickel, M., Rosasco, L. & Poggio, T. Holographic embeddings of knowledge graphs. in *Thirtieth Aaai conference on artificial intelligence* (2016).
297. Lin, Y., Liu, Z., Sun, M., Liu, Y. & Zhu, X. Learning entity and relation embeddings for knowledge graph completion. in *Twenty-ninth AAAI conference on artificial intelligence* (2015).
298. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901).
299. Han, X. *et al.* Openke: An open toolkit for knowledge embedding. in *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations* 139–144 (2018).
300. Kazemi, S. M. & Poole, D. Simple embedding for link prediction in knowledge graphs. *ArXiv Prepr. ArXiv180204868* (2018).
301. Balažević, I., Allen, C. & Hospedales, T. M. Tucker: Tensor factorization for knowledge graph completion. *ArXiv Prepr. ArXiv190109590* (2019).

302. Davis, J. & Goadrich, M. The relationship between Precision-Recall and ROC curves. in *Proceedings of the 23rd international conference on Machine learning* 233–240 (2006).
303. Vu, T., Nguyen, T. D., Nguyen, D. Q., Phung, D., & others. A capsule network-based embedding model for knowledge graph completion and search personalization. in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* 2180–2189 (2019).
304. Hooton, F., Menichetti, G. & Barabási, A.-L. Exploring food contents in scientific literature with FoodMine. *Sci. Rep.* **10**, 16191 (2020).
305. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Ser. B Methodol.* **57**, 289–300 (1995).
306. KEGG: Kyoto Encyclopedia of Genes and Genomes | Nucleic Acids Research | Oxford Academic. <https://academic.oup.com/nar/article/28/1/27/2384332>.
307. UniProt: the Universal Protein knowledgebase | Nucleic Acids Research | Oxford Academic. https://academic.oup.com/nar/article/32/suppl_1/D115/2505378.
308. BLAST: Basic Local Alignment Search Tool. <https://blast.ncbi.nlm.nih.gov/Blast.cgi>.
309. Senanayake, U. M., Lee, T. H. & Wills, R. B. H. Volatile constituents of cinnamon (*Cinnamomum zeylanicum*) oils. *J. Agric. Food Chem.* **26**, 822–824 (1978).

310. Sommer, K., Hillinger, M., Eigenmann, A. & Vetter, W. Characterization of various isomeric photoproducts of ergosterol and vitamin D₂ generated by UV irradiation. *Eur. Food Res. Technol.* **249**, 713–726 (2023).
311. Joradon, P. *et al.* Ergosterol Content and Antioxidant Activity of Lion's Mane Mushroom (*Hericium erinaceus*) and Its Induction to Vitamin D₂ by UVC-Irradiation: in *Proceedings of the 8th International Conference on Agricultural and Biological Sciences* 19–28 (SCITEPRESS - Science and Technology Publications, Shenzhen, China, 2022). doi:10.5220/0011594600003430.
312. Dahlqvist, A. *et al.* Phospholipid:diacylglycerol acyltransferase: An enzyme that catalyzes the acyl-CoA-independent formation of triacylglycerol in yeast and plants. *Proc. Natl. Acad. Sci. U. S. A.* **97**, 6487–6492 (2000).
313. Shamsiev, A., Park, J., Olawuyi, I. F., Odey, G. & Lee, W. Optimization of ultrasonic-assisted extraction of polyphenols and antioxidants from cumin (*Cuminum cyminum* L.). *Korean J. Food Preserv.* **28**, 510–521 (2021).
314. Nagao, A., Maeda, M., Lim, B. P., Kobayashi, H. & Terao, J. Inhibition of β -carotene-15,15'-dioxygenase activity by dietary flavonoids. *J. Nutr. Biochem.* **11**, 348–355 (2000).
315. Shi, X. *et al.* Identification and investigation of a novel NADP⁺-dependent secoisolariciresinol dehydrogenase from *Isatis indigotica*. *Front. Plant Sci.* **13**, (2022).
316. Cui, H. *et al.* Comparative analysis of nuclear, chloroplast, and mitochondrial genomes of watermelon and melon provides evidence of gene transfer. *Sci. Rep.* **11**, 1595 (2021).

317. Wishart, D. S. *et al.* PathBank: a comprehensive pathway database for model organisms. *Nucleic Acids Res.* **48**, D470–D478 (2020).
318. Wishart, D. S. *et al.* HMDB 5.0: the Human Metabolome Database for 2022. *Nucleic Acids Res.* **50**, D622–D631 (2022).
319. Wolf, T. *et al.* Transformers: State-of-the-Art Natural Language Processing. in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* 38–45 (Association for Computational Linguistics, Online, 2020). doi:10.18653/v1/2020.emnlp-demos.6.
320. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, Inc., 2019).
321. Loshchilov, I. & Hutter, F. Decoupled Weight Decay Regularization. Preprint at <https://doi.org/10.48550/arXiv.1711.05101> (2019).
322. Settles, B. *Active Learning Literature Survey*. <https://minds.wisconsin.edu/handle/1793/60660> (2009).
323. OpenAI. ChatGPT. <https://openai.com/blog/chatgpt>.
324. Technical University of Denmark. Frida Food Data, Version 4.2. <https://frida.fooddata.dk/?lang=en>.
325. *Entrez Programming Utilities Help*. (National Center for Biotechnology Information (US), 2010).

Appendix

Appendix A. Supplementary information

A.1. Knowledge integration and decision support for accelerated discovery of antibiotic resistance genes

A.1.1. Knowledge Graph Constructor

A.1.1.1. Data collection

To construct a comprehensive knowledge graph of *E. coli* antibiotic resistance, existing knowledge bases and literature were integrated. A summary of the 10 different sources we used in our work can be found in **Supplementary Table 1**. The following list describes each source in detail.

CARD¹¹³. From version 2.0.0 of the Antibiotic Resistance Ontology (ARO), two predicates '*targeted_by_drug*' and '*confers_resistance_to_drug*' for only the *E. coli* genes were extracted. This results in a total of 147 triplets between 72 *E. coli* genes as the subject and 33 antibiotics as an object, where predicates are renamed from '*targeted_by_drug*' and '*confers_resistance_to_drug*' to '*targeted by*' and '*confers resistance to antibiotic*', respectively.

Gene Ontology (GO) dataset²⁸¹. From version 2.4.26 of AmiGO 2, we downloaded annotation data for *E. coli* K-12. Note that a part of its knowledge originally comes from external sources like EcoCyc¹⁴³, but we still consider them as GO. We represent them using the following three types of triplets types: (*gene, has, molecular_function*), (*gene, is part of, cellular_component*), and (*gene, is involved in, biological_function*). This results in 17,739 triplets.

Liu et al.¹⁶⁰ In this work, among the 3,985 single-gene knockouts (KEIO) in *E. coli*, 283 strains showed susceptibility to 1 of 14 antibiotics. The work then extends to 8 more antibiotics to test the susceptibility of 283 screened strains. From the initial screening of 14 antibiotics, 3,985 knockout strains without a response of susceptibility to 14 antibiotics are considered negative results. In the second screening of 283 strains for 8 antibiotics, any strains without a response of susceptibility to 8 antibiotics are considered negative results. Any genes with positive results are considered to confer intrinsic resistance to a varied set of antibiotics, as their deletion renders the cell more sensitive than the wild type. We curate the facts discovered in this study using the triplet types (*gene, confers resistance to antibiotic after 18 hours, antibiotic*) for the positive results and (*gene, confers no resistance to antibiotic after 18 hours, antibiotic*) for the negative results. As a result, a total of 55,877 triplets are created from this study.

Tamae et al.⁵⁸ This work, conducted by the same group that published Liu et al.¹⁶⁰, presents strains that are susceptible to 1 of 7 antibiotics among 3,985 single-gene knockouts in *E. coli*. We apply the same representation used in the Liu et al. dataset. This process results in 26,926 triplets.

Shaw et al.¹¹⁹ This study presents genes with significant fold-change in expression (profiled with gene expression microarrays) 30 minutes after induction of 4 antibiotics (norfloxacin, kanamycin, rifampicin, and ampicillin) at a different drug concentration of each. From this, we collected genes with positive fold-change at the highest drug concentration of each. This results in a total of 145 facts about 139 genes upregulated by 4 antibiotics, and we represent them using the triplet type (*gene, upregulated by antibiotic after 30 mins, antibiotic*). They examined the expression levels of 3,913 genes. Among them, any genes that are not upregulated in this study are represented using the triplet type (*gene, not upregulated by antibiotic after 30 mins, antibiotic*). This results in a total of 15,327 triplets.

Nichols et al.¹¹⁷ A seminal work in chemical genomics in *E. coli* was published in which a library of over 4,000 Keio¹⁵⁴ knockout strains was screened under many different chemical and physical conditions using phenotype microarray. In their work, individual strains were plated robotically in a 1,536-well format, and colony size was investigated to determine fitness. We used the published normalized dataset of this raw data. Please note that we took antibiotics with the highest concentration to be conservative on the findings. Statistical testing was performed based on the description of the original article (FDR<0.05), and we only considered statistically significant results in the negative tail of fitness score distribution (*i.e.*, gene deletions that show increased susceptibility to an antibiotic over wild-type). Missing values are imputed with Random Forest before statistical testing. Among them, we only took gene IDs with clear mappings to the original gene symbol. There was a total of 51 antibiotics, and we identified 2,700 pairs of genes and antibiotics that are considered to confer resistance to the antibiotic. We represented

those positive findings using the triplet type (*gene, confers resistance to antibiotic after 15 hours, antibiotic*). Any of the genes with no positive findings for the 51 antibiotics are represented using the triplet type (*gene, confers no resistance to antibiotic after 15 hours, antibiotic*). This results in a total of 186,941 triplets.

Zhou et al.¹¹⁸ This work measures nearly 2,000 growth phenotypes in Phenotype Microarrays for *E. coli* K-12 mutants with individual deletions of all two-component systems (a total of 47 genes). Among them, there were a total of 31 antibiotics, and 78 positive findings of 28 genes (*i.e.*, mutants that show increased susceptibility to antibiotics over wild-type) were identified, and we represent them using the triplet type (*gene, confers resistance to antibiotic after 36 hours, antibiotic*). Any of 47 genes with no positive findings for 31 antibiotics are represented using the triplet type (*gene, confers no resistance to antibiotic after 36 hours, antibiotic*). This results in 1,457 triplets.

Soo et al.⁶³ This work examined the effect of *E. coli* genes from the ASKA library²⁸² overexpressed on plasmids challenged by 237 toxic chemicals, among which results for the 44 antibiotics were extracted. In this study, it found genes conferring increased fitness (growth rates) in the presence of toxins compared to control, which we consider positive findings. A total of 59 positive findings of 32 genes were identified, which we represent using the triplet type (*gene, confers resistance to antibiotic after 7 days, antibiotic*). The rest of the genes with no positive findings of 44 antibiotics are represented using the triplet type (*gene, confers no resistance to antibiotic after 7 days, antibiotic*). This results in 188,936 triplets.

hiTRN²⁸³ The original hiTRN data has 6,754 gene-regulatory relations with 207 transcription factors (TFs). Among them, 2,159 gene-regulatory relations with 14 TFs

were from the ChIP experiments. We include the gene-regulatory relations in the antibiotic resistance knowledge base to use them in training the hypothesis generator. We considered any *E. coli* genes not reported in the ChIP experiments as negative facts with regard to binding with the 14 TFs. This creates negative triplet types (*gene, no activates, gene*) and (*gene, no represses, gene*). Along with positive triplets types (*gene, activates, gene*) and (*gene, represses, gene*), hiTRN data results in a total of 101,878 triplets.

Girgis et al.²⁸⁴ The authors exposed a transposon-mutagenized library of *E. coli* to each of 17 antibiotics, propagating the library for multiple generations. Then they determined the quantitative contribution of each gene to *E. coli*'s intrinsic antibiotic susceptibility using a microarray-based genetic foot-printing technique. From their resource, we found a total of 576 positive findings of 430 genes, which can be represented using the triplet type (*gene, confers resistance to antibiotic after 3 days, antibiotic*). After the exclusion of gene-antibiotic pairs with no available data, the rest of the *E. coli* genes with no positive findings of 17 antibiotics are represented using the triplet type (*gene, confers no resistance to antibiotic after 3 days, antibiotic*). This results in 63,636 triplets.

In addition to the information provided above, more detailed source characteristics are provided in **Supplementary Table 6**.

A.1.1.2. Knowledge inference

We manually generated the 15 sets of knowledge inference rules after careful inspection of the existing triplet types in the knowledge graph. Application of these knowledge inference rules generated 20,841 new triplets, therefore increasing the number of total triplets in the knowledge graph by 3.16% (from 658,726 to 679,567). However, such a

manual approach cannot guarantee complete coverage of all possible rules. We, therefore, consider using an automated approach by utilizing automatic knowledge graph construction methods like COMET⁵³. However, we ultimately decided to leave it as future work since more extensive analysis needs to be performed to ensure that such an automated approach does not create unwanted noise in the data, therefore negatively affecting the downstream performance of the hypothesis generators.

A.1.1.3. Analysis of the knowledge graph

Among the 7,917 nodes in the knowledge graph, 4,488 were *E. coli* genes (55.0%), 1,782 were molecular functions (22.5%), 1,522 were biological processes (19.2%), 152 were cellular components (1.9%), and 104 were antibiotics (1.3%) (**Figure 3.2d**). We then classified the 104 antibiotics in the knowledge graph into 6 different taxonomic groups using a chemical classification ontology²⁸⁵ (**Supplementary Figure 7**) and analyzed the distribution of the CRA triplets. The results show that the organoheterocyclic compounds group which contains a ring with at least one carbon atom and one non-carbon atom was the most prevalent group containing 28 antibiotics. The CRA triplets belonging to this antibiotic group were also the most well-explored ones with 86.02% of the whole data being already covered in the knowledge graph (**Supplementary Figure 8**).

A.1.2. Inconsistency Resolver

Multiple truth discovery methods have been proposed over the past decade and have been successfully applied in diverse domains. The primary application domain of these methods is conflict resolution between the web sources, where information conveyed on

a particular web page conflicts with that of other web pages²⁸⁶. In this setting, three popular approaches exist: iterative methods, optimization methods, and probabilistic methods^{287,288}. Recently, using a link prediction method to decide the truth among conflicts has been proposed¹³⁷. Some notable works in biological sciences include inconsistency repair in the *E. coli* gene regulatory network using answer set programming⁶⁰ and inconsistency resolution in signal transduction knowledge using integer linear programming²⁸⁹.

A.1.2.1. Resolution algorithms

In addition to the AverageLog¹⁴⁵ inconsistency resolution method described in the **Methods**, we tested 5 additional inconsistency resolution algorithms. The first algorithm is Voting¹⁴⁵, where the triplet asserted by most sources is selected. Other algorithms are briefly described below.

A.1.2.1.1. Sums²⁹⁰:

$$R^i(s) = \sum_{t \in T_s} B^{i-1}(t) \quad (1)$$

$$B^i(t) = \sum_{s \in S_t} R^i(s) \quad (2)$$

A.1.2.1.2. AverageLog¹⁴⁵:

$$R^i(s) = \log |T_s| \frac{\sum_{t \in T_s} B^{i-1}(t)}{|T_s|} \quad (3)$$

$$B^i(t) = \sum_{s \in S_t} R^i(s) \quad (4)$$

A.1.2.1.3. Investment¹⁴⁵:

$$R^i(s) = \sum_{t \in T_s} B^{i-1}(t) \frac{R^{i-1}(s)}{|T_s| \cdot \sum_{r \in S_t} \frac{R^{i-1}(r)}{|T_r|}} \quad (5)$$

$$B^i(t) = G \left(\sum_{s \in S_t} \frac{R^i(s)}{|T_s|} \right) \quad (6)$$

where $G(x) = x^g$ and $g = 1.2$ as chosen by the author of the method.

A.1.2.1.4. PooledInvestment¹⁴⁵:

$$R^i(s) = \sum_{t \in T_s} B^{i-1}(t) \frac{R^{i-1}(s)}{|T_s| \cdot \sum_{r \in S_t} \frac{R^{i-1}(r)}{|T_r|}} \quad (7)$$

$$B^i(t) = H^i(t) \cdot \frac{G(H^i(t))}{\sum_{d \in M_t} G(H^i(d))} \quad (8)$$

where $H^i(t) = \sum_{s \in S_t} \frac{R^i(s)}{|T_s|}$ and $g = 1.4$ as chosen by the author of the method.

A.1.2.1.5. TruthFinder²⁹¹:

$$R^i(s) = \frac{\sum_{t \in T_s} B^{i-1}(t)}{|T_s|} \quad (9)$$

$$B^i(t) = 1 - \prod_{s \in S_t} (1 - R^i(s)) \quad (10)$$

where the hyperparameters ρ and γ (not shown here; refer to the original paper²⁹¹ for implementation details) were set to 1.8 and 0.8, respectively, based on the empirical study (Supplementary Table 7).

A.1.2.2. Evaluation of the algorithms using a synthetic dataset

We use a synthetically generated dataset to evaluate the inconsistency resolution algorithms in a controlled setting. In this section, we describe how the synthetic dataset was constructed and how the resolution algorithms were evaluated.

A.1.2.2.1. Construction of the synthetic dataset

We constructed synthetic datasets from the hiTRN²⁸³ dataset where each synthetic dataset consists of triplets from multiple sources with a pre-determined error rate for each source. The performance of inconsistency correction methods is measured for each dataset. In a dataset, multiple sources exist where each source is comprised of triplets and the size of the source follows the normal distribution of $N(1,000,333)$. We also investigated the impact of the number of triplets per source on the accuracy of inconsistency correction (**Supplementary Table 8**). We varied the source size as some of the inconsistency correction methods we compared take it into account. Each source was falsified by replacing the predicate with its negative counterpart (e.g., replace ‘no activates’ with ‘activates’) at the specific error rate E , which follows the normal distribution of $N(E, E/3)$ (*i.e.*, certain triplets are incorrect, which creates inconsistency when compared to other source data). For the standard deviation of two normal distributions, a mean divided by 3 was selected to sample positive numbers at a 99.9% chance. We iterated this procedure S times, and therefore, creating S sources. That is, a dataset consists of S sources where average of source size is 1,000 and the average source error rate is E . We experimented with varying $E(0.1, 0.2, 0.3, 0.4)$ and $S(3, 5, 7, 9)$ to see how these variables affect the performance of inconsistency correction methods. For each E

and S , we created 1,000 datasets to get the statistics of inconsistency correction performance, thus resulting in 16,000 datasets. We verified that 1,000 sampled datasets were enough to approximate the true population of the two parameters (**Supplementary Figure 9**). We also had extra experiments by fixing the parameters of E and S , and there was no significant difference in accuracy across the six methods (**Supplementary Figure 10 ~ Supplementary Figure 12**).

A.1.2.2.2. Rules to identify inconsistencies in the synthetic dataset

The following rules were used to detect inconsistencies in the simulated dataset: 1) 'represses', 'no represses' and 2) 'activates', 'no activates'. That is, these rules detect conflicts in gene-regulatory relations where a protein (subject) either represses or does not repress the expression of a gene (object), and likewise, a protein (subject) either activates or does not activate the expression of a gene (object).

A.1.2.2.3. Performance metric

Accuracy was measured by the number of correctly resolved inconsistencies divided by the number of total inconsistencies. PCC (Pearson's correlation coefficient) was measured between the true and estimated trustworthiness of sources. The true trustworthiness of the source is essentially $1 - \text{error rate } (E)$ of the source.

A.1.2.2.4. Stopping criteria

In our simulated studies, we observed that the mean difference, δ , of trustworthiness between the previous iteration and present iteration is rapidly saturated within 10 iterations (**Supplementary Figure 13**). Therefore, the stopping criterion for the iterative inconsistency correction methods was when the number of iterations reached 10.

A.1.2.2.5. Evaluation results.

To investigate the feasibility of the computational correction of inconsistencies, we evaluated six algorithms of inconsistency resolution methods using the synthetic datasets created above. **Supplementary Figure 14** shows AverageLog, Investment, PooledInvestment, and TruthFinder outperform Sums and Voting overall. As expected, the accuracy of inconsistency correction monotonically increases when the number of sources increases, and when the average percentage of error per source decreases. The performance gap across methods becomes more distinguishable as the average percentage of error per source increases. Interestingly, PooledInvestment begins outperforming when the number of sources increases whereas its performance is suboptimal when the number of sources is few (e.g., 3). AverageLog is particularly accurate when the number of sources is a few. This observation is particularly clear when true and estimated source trustworthiness are compared (**Supplementary Figure 15**). Given those major conflicting facts in the *E. coli* antibiotic resistance knowledge base come from two sources, we have decided to use AverageLog.

A.1.2.3. Inconsistency resolution results

A.1.2.3.1. Level 1 inconsistency resolution

Using the inconsistency detection criteria discussed in the **Methods**, we initially identified 291 conflicting sets of triplets originating from the two sources Liu et al.¹⁶⁰ and Tamae et al.⁵⁸ between the two predicates '*confers resistance to antibiotic after 18 hours*' and '*confers no resistance to antibiotic after 18 hours*' (**Supplementary Figure 16**). Note that these two sources share identical characteristics such as exposure time, parent strain,

and media. However, we found that metronidazole, which is a pro-drug and is converted to the active-drug by bacteria only under anaerobic condition^{156,157}, was related to 55 sets of inconsistencies. Therefore, we decided to discard these metronidazole-related inconsistencies and only take into account 236 sets of inconsistencies for any further evaluation (more detail in **Appendix A.1.2.4**). We then applied the AverageLog inconsistency resolution method, which was chosen from experimenting with the synthetic dataset above, to resolve these 236 sets of inconsistencies. When compared with the ground truth wet-lab validation results, our computational resolution results had an F1 score of 0.24 and an accuracy of 0.86. However, performing an inconsistency resolution where only two conflicting sources exist leads to the problem that the belief $B(t)$ of all resolved triplets are equal. In other words, triplets from the source that have higher trustworthiness $R(s)$ (in our case Tamae et al. with 0.53) were chosen over the triplets from the source Liu et al with 0.40. We denote these as level 1 inconsistencies.

A.1.2.3.2. Level 2 inconsistency resolution

As we have learned using a synthetic dataset that more sources lead to better resolution performance in **Supplementary Figure 14**, we tested to see if increasing the number of conflicting sources would also translate to improved performance in the real-world scenario. To do this, we observed source characteristics and found that Nichols et al.¹¹⁷ share the same source characteristics as Tamae et al. and Liu et al. except for the shorter exposure time of 15 hours instead of 18 hours. Thus, by alleviating the inconsistency detection criteria to treat predicates '*confers (no) resistance to antibiotic after 15 hours*' and '*confers (no) resistance to antibiotic after 18 hours*' equally, a total of 1,096 sets of conflicting triplets (which we denote as level 2) were identified as shown in

Supplementary Figure 17. Among these, we only compared the original subset of 236 sets of inconsistencies with the ground truth wet-lab validation results from level 1. Results show that the F1 score increased by 75% from 0.24 to 0.42 and the accuracy increased by 6.98% from 0.86 to 0.92 when compared to the level 1 results.

A.1.2.3.3. Level 3 inconsistency resolution

Supported by experimental proof that more sources indeed lead to a better resolution, we alleviated the inconsistency detection criteria one more level (level 3) by ignoring all source characteristics including the exposure time. This process allowed us to increase the number of conflicting sources to 8, and a total of 2,131 sets of conflicting triplets were identified as shown in **Supplementary Figure 18**. Out of these 2,131 sets, we still compared the original 236 sets of inconsistencies that we have validated in level 1. The results show that the F1 score increased to 0.50 and accuracy increased to 0.94, a 108.30% and 9.30% increase, respectively when compared to the level 1 results. Although we found using the simulated datasets that PooledInvestment works the best when there are 8 sources (**Supplementary Figure 15**), AverageLog still performed the best among all the resolution methods.

A.1.2.4. Wet-lab validation using single-gene knockout strains of *E. coli* BW25113

To validate whether a gene confers resistance or not, wild-type Keio strain BW25113 and its derivative single-gene knockout (KO) strains were used¹⁵⁴. MIC values of the following antibiotics were measured: Amoxicillin (Sigma), Ampicillin (Roche Diagnostics), Apramycin (Alfa Aesar), Cephadrine (Alfa Aesar), Chloramphenicol (Calbiochem), Geneticin (Teknova), Hygromycin B (Calbiochem), Kanamycin (Acros Organics),

Levofloxacin (Chem-Impex), Norfloxacin (Sigma), Novobiocin (Calbiochem), Oxycarboxin (Sigma), Paromomycin (Chem-Impex), Rifampin (Alfa Aesar), Sisomicin (TCI), Spectinomycin (RPI), Streptomycin (Across Organics), Sulfanilamide (Alfa Aesar), Triclosan (Cayman Chemical Company), Troleandomycin (Enzo Life Sciences), and Vancomycin (VWR Life Science). Since KO strains had a kanamycin resistance gene, the kanamycin resistance gene was removed from the required KO strains¹⁵⁵ to measure the resistance in kanamycin. Antibiotics and strains were preserved at -80°C until used.

1 µL of the required preserved strain was inoculated in 200 µL LB broth and grown overnight in an incubator shaker (BioTek Synergy HTX) at 37°C. ~3 µL of grown culture was transferred, using a replicator, to LB agar plates containing different amounts of antibiotics, and plates were incubated overnight (~18 hours) at 37°C in an incubator. The next day, the absence and presence of colonies were monitored. The minimum concentration of antibiotic, at which no colonies were observed, was defined as minimum inhibitory concentration (MIC). In the case of metronidazole, colonies were observed at all concentrations. Metronidazole is a pro-drug and inactive but in anaerobic conditions, this is converted to an active form by the bacteria^{156,157}. The active form is toxic which leads to the killing of bacteria. As our experimental conditions were aerobic, metronidazole was converted to an active form, and we observed colonies at all concentrations. Subsequently, we removed metronidazole from our study.

A.1.3. Hypothesis Generator

There are multiple approaches when building statistical models over knowledge graphs. In this work, we implemented three types of hypothesis generator models PRA, MLP, Stacked, TransE, and TransD. We refer the interested reader to a review of the various machine learning techniques over knowledge graphs provided by Nickel et. al¹³⁷.

A.1.3.1. Preprocessing the knowledge graph

A.1.3.1.1. Use of the negative samples

We investigated how to utilize the negative samples when training different models of the hypothesis generator. The first option was to treat both 8 positive and 4 negative predicates (**Supplementary Table 4**) uniquely, but the key issue with this option was that the knowledge graph was now skewed to the negative predicate types since they were the majority of the edges. Next, we considered treating the negative samples as known negatives of their positive counterparts. For example, we could optimize against the cross-entropy loss when training the MLP using these known negatives. However, since there only existed known negatives for 4 of the 8 predicates (**Supplementary Table 4**), it was not clear how we should train the remaining predicate types. Ultimately, we decided to use alternate methods for training the hypothesis generator models. For the PRA, we decided to follow the original approach used in the paper of choosing the negative samples via the closed-world assumption. When the PRA leverages the closed-world assumption, it chooses the negative samples based on a selection and filtering strategy that helps to identify important samples to train on. For the MLP, we used the more standard training regimen of margin-based ranking loss which generates negatives through corruption (see **Methods**). For the Stacked, we did leverage the negative samples during training, since we trained the ensemble only on edges that consisted of

the CRA predicate. This was possible as a separate stacked ensemble was produced for each predicate type in the knowledge graph, and we were only predicting on the CRA predicate. Having said this, we never came to a concrete conclusion on whether using these negative samples could be beneficial. We believe there is still a potential opportunity to use these negatives during training.

A.1.3.1.2. Data split for the 5-fold cross-validation

We split the knowledge graph into 5-folds to train/evaluate different hypothesis generator models. As for the distribution of the positive samples, we allotted 72% of the positive CRA triplets to the training set, 8% to the validation set used to identify optimal thresholds for the models, and the rest 20% to the test set. The remaining positive samples (non-CRA triplets) were then distributed across the training set. As for the distribution of negative samples used for evaluation of the hypothesis generator models, for every positive CRA triplet in the knowledge graph, we sampled 49 negatives with the same antibiotic from our known negatives^{152,292}. We chose to have this uneven balance of negatives to positive samples to reflect the fact that a gene is far more likely to not confer resistance to a certain antibiotic. Since for some antibiotics, there would not be enough genes to produce negatives for a given edge, we limited the number of negative samples to 49. Having this ratio of negative to positive samples, the baseline average precision can be approximated to 2% when evaluating the performance of our models²⁹³. Note that in some cases, there were not enough known negatives for an edge to produce negative samples. In such cases, we used the local closed-world assumption to generate synthetic negatives. To construct these negatives for every positive edge that did not have enough

negatives, we randomly replaced the gene that was not already a known negative or known positive.

A.1.3.1.3. Removal of temporal information

As discussed in the **Methods** section of the main manuscript, we removed the temporal information from some of the predicates in the knowledge graph (**Supplementary Table 2** and **Supplementary Table 4**). This decision was to handle the lack of training data if we were to treat each predicate with varying temporal information (e.g., the positive predicate ‘CRA after 7 days’ only has 59 triplets). However, removing the temporal information has the side effect of creating potential inconsistencies. For example, although the two triplets (*cydX*, CRA after 15 hours, Vancomycin) and (*cydX*, -CRA after 18 hours, Vancomycin) supported by Nichols et al.¹¹⁷ and Tamae et al.⁵⁸, respectively, are not inconsistencies in their original form, they become inconsistencies after removing the temporal information. As described in **Appendix A.1.2.3.3**, removing the temporal information from the knowledge graph results in an increase of inconsistencies from 236 (level 1 inconsistency) to 2,131 (level 3 inconsistency).

A.1.3.2. Path Ranking Algorithm (PRA)

A.1.3.2.1. An observable graph feature model

Observable graph feature models extract features from the observed edges over the knowledge graph to predict the existence of a new edge, and the PRA^{65,146} is an example of such a model. Among others, the PRA has been used for link prediction over the Nell knowledge graph²⁹⁴ and the Knowledge Vault project¹⁵². Liekens et al.²⁹² also used a graph feature model like PageRank to predict genes causing disease. The advantage of

this type of approach is that the features are readily observable over the graph, therefore translating to useful reasons why the prediction was made. The models in this category are well suited for modeling local patterns in the data.

The PRA performs random walks over the graph at a bounded step size to identify the existence of new edges over the graph. The features of this model are the path probabilities of reaching an object entity from a subject entity. The PRA leverages the closed-world assumption to identify the negative training samples for the model. As the random walks are performed, paths will also result in the wrong object entities. These subject-object pairs can act as negative samples. The paths generated for each wrong object entity are scored against an untrained model using default initialized weights to rank these negative samples. A selection strategy is then used to choose which negative samples to train for the model. In our case, we chose to use all negative samples found. We leveraged the original Java implementation by the author of the PRA.

A.1.3.2.2. The path features obtained by the PRA

One advantage of the PRA is that it provides interpretable results. **Supplementary Table 9** and **Supplementary Table 10** show the path features and their corresponding weights trained by the PRA when generating the first and second iteration of hypotheses, respectively. In both iterations, the most important feature identified by the PRA was as follows:

gene $\xrightarrow{\text{is involved in}}$ *biological_process* $\xrightarrow{\text{is involved in}^{-1}}$ *gene* $\xrightarrow{\text{confers resistance to antibiotic}}$ *antibiotic*.

This path corresponds to a sequence of three predicates linking the gene-antibiotic pair. The inverse sign indicates the inverse direction that the random walker took in creating a

path. In other words, this path tells us that at least one other gene that is involved in the same biological process also confers resistance to the antibiotic of interest. One can take these paths as evidence behind the predictions.

A.1.3.3. Multilayer Perceptron (MLP)

A.1.3.3.1. Latent feature model

Another popular approach to this field of research is to generate latent features by using embeddings for the entities and/or predicates in the knowledge graph. The features generated from these types of models are called “latent” because they are not directly observable over the graph. Moreover, these types of relational models are well suited to modeling global patterns that exist over the graph¹³⁷. To produce these latent features, the entities in the knowledge graph are converted to numerical vectors or embeddings that are treated as learnable parameters by the model. The relationship between these entities is then derived from the interaction of their latent features in the respective model. The outputs of these models consist of a single score or confidence indicating whether an edge should exist between the two entities.

For instance, Ding et. al²⁹⁵ generated latent features by using the Neural Tensor Network¹⁴⁷ to perform event-driven stock market prediction. The Entity-Relation Multilayered Perceptron (ER-MLP)¹⁵², which has been shown to provide comparable results to the Neural Tensor Network while using significantly fewer parameters, was used to predict new facts over the Freebase knowledge graph for the Knowledge Vault project. More recently, the use of holographic embeddings has shown promising results²⁹⁶.

Another class of latent feature models involves predicting the existence of edges over a knowledge graph by measuring the similarity of the vector-spaced entity embeddings. For instance, TransE¹³⁹ identifies the score for a certain edge as the distance between the predicate-specific translations of two entity embeddings. The distance can be measured by using Euclidean distance. Although this type of model requires very few parameters, this is with the cost of modeling performance. Hence, the TransH¹⁴⁰ and TransR²⁹⁷ have been introduced to improve on this limitation by introducing additional parameters to improve the TransE performance.

A.1.3.3.2. Word embeddings form clusters based on their entity types

The MLP concatenates a single predicate embedding of size 50 and two entity embeddings of size 50 each to train the model. These embeddings consist of learnable parameters that capture a semantic representation after training. When we reduced the dimensions of the entity embeddings by performing the principal component analysis (PCA)²⁹⁸, we observed noticeable clusters forming depending on their entity type as shown in **Supplementary Figure 19**. Interestingly, the entity types (*i.e.*, gene, antibiotic, etc.) of these entities were never provided to the MLP during training. It simply learned the entity types on its own.

A.1.3.4. Stacked

It has been shown experimentally that neither the latent feature model nor the graph feature model can predict optimally on its own¹⁵². As they are well equipped to model different types of patterns in the knowledge graph, researchers have built combined models that incorporate both the global and local perspectives over the knowledge

graph^{152,179}. The fused prior models in this category, have shown to have state-of-the-art performance due to this dual nature in pattern recognition. Consequently, we decided to explore this option to automatically generate new hypotheses over the *E. coli* knowledge graph. This ensemble approach using AdaBoost¹⁶¹ leverages a sequence of one-depth decision trees. Each decision tree is trained on a modified version of the training set. After each iteration of training, when the classifier incorrectly classifies a sample, that sample is upweighted in importance to ensure that the classifier focuses its attention on correcting the mistake during the next iteration. The predictions of each weak learner are combined through a weighted majority vote to make the final prediction.

A.1.3.5. Other graph embedding methods

We tested graph embedding methods TransE¹³⁹ and TransD¹⁶² that model relationships between the entities by interpreting them as a translational operation. That is, the model optimizes the embeddings by enforcing the vector operation of the subject entity embedding plus the relation embedding to be close to the object entity embeddings. We used self-adversarial negative sampling with a temperature fixed to 1.0, optimized using Adam¹⁵⁰, fine-tuned the hyperparameters on the validation dataset, and performed early stopping. The range of the grid search used for hyperparameter search was as follows: negative samples $n \in \{25, 50, 100\}$, embedding dimension $d \in \{128, 256, 512, 1024\}$, margin $\gamma \in \{6.0, 12.0, 24.0\}$, and learning rate $\alpha \in \{0.001, 0.0001\}$. We used the open-source implementation of these models using the OpenKE toolkit²⁹⁹. For TransE, the best hyperparameters obtained were $n = 100$, $d = 256$, $\gamma = 12.0$, and $\alpha = 0.001$. For TransD, the best hyperparameters obtained were $n = 100$, $d = 256$, $\gamma = 24.0$, and $\alpha = 0.0001$. Results of these models are provided in **Supplementary Table 5**. We also tested two

additional graph embedding methods SimpleE³⁰⁰ and RotatE¹⁸⁴, but we were not able to find optimal set of hyperparameters that performs better than the PRA even after an extensive grid-search.

A.1.3.6. State-of-the-art knowledge graph completion methods

In addition to the five methods PRA, MLP, Stacked, TransE, and TransD considered in this work, we also tested more recent state-of-the-art methods that were introduced after the project was conceived. We tested factorization-based knowledge graph completion methods TuckER³⁰¹ and performed the hyperparameter search among the following combinations with the best setting marked in bold: learning rate $\in \{\mathbf{0.0002}, 0.0005, 0.001\}$, the decay rate $\in \{0.99, 0.995, \mathbf{1.0}\}$, entity embedding dimension $\in \{\mathbf{200}\}$, relation embedding dimension $\in \{\mathbf{30}, 200\}$, input dropout $\in \{\mathbf{0.2}, 0.3\}$, first hidden dropout $\in \{0.1, 0.2, \mathbf{0.4}\}$, second hidden dropout $\in \{0.2, 0.3, \mathbf{0.5}\}$, and label smoothing $\in \{0.0, \mathbf{0.1}\}$. We used a batch size of 128 and trained for 500 iterations with early stopping. As shown in **Supplementary Table 5**, Tucker has a 0.7% higher F1 score than the stacked model (30.1% vs. 30.8%; p -value: 0.65). For our future work, we expect to see higher discovery rates using such state-of-the-art knowledge graph completion methods.

A.1.3.7. Optimization and evaluation

A.1.3.7.1. Optimization criteria

A validation set was used to identify optimal thresholds for the PRA, MLP, TransE, TransD, and TuckER. We optimized using the F1-score. Additionally, since we were training new PRA and MLP models for each fold, we optimized the number of classifiers and the

learning rate for the Stacked model during each fold. For this case, we optimized using Average Precision.

A.1.3.7.2. ROC curve for evaluation

The receiver operating characteristic (ROC) considers the true negatives during evaluation. This metric contains information about how well our model was able to correctly identify a negative sample. Due to this attribute, the ROC is not an ideal metric to use for a highly unbalanced dataset like ours, where the number of positive test samples is significantly less than that of negative samples. Since this true negative metric is relatively unimportant to the positive samples, we produced precision-recall (PR) curves for each model. This metric only considers true positives, false positives, and false negatives. This is a somewhat harder metric since there are significantly fewer positive test samples than negatives. The PR curve has also been shown to provide a more informative metric for retrieval tasks when compared to the ROC curve³⁰².

A.1.3.8. Hypothesis generation on individual sources

We wanted to test if the hypothesis generator trained using our knowledge graph predicts better associations than the ones trained using individual sources. To do this, we treated each source as a unique knowledge graph to evaluate the three hypothesis generator models PRA, MLP, and stacked using 5-fold cross-validation (**Supplementary Table 11**). However, we were not able to train PRA on any single source knowledge graph as PRA requires at least two unique predicates to build path features. As we were not able to train the PRA on any single source knowledge graph, we also could not train the stacked model. For the MLP, we were able to get results for 5 sources. The combined AUCPR of the MLP

from these 5 individual sources was 0.26 ± 0.15 . This was greater than the AUCPR of the MLP trained on our knowledge graph (0.22 ± 0.01), although statistically insignificant (p -value: 0.61).

A.1.3.9. Multi-iteration hypotheses generation

For the first iteration, we generated 108,078 CRA hypotheses which account for 23.2% of all gene-antibiotic pairs in the knowledge graph (466,752 total possible pairs from 4,488 *E. coli* genes and 104 antibiotics). We then grouped the hypotheses into 5 bins (**Figure 3.5a**) and tested all hypotheses above the 20% probability that we have antibiotics in stock (105 hypotheses out of 149 hypotheses). For the hypotheses with a probability \leq 20%, we randomly selected 121 hypotheses that we have antibiotics in stock out of 107,929 hypotheses. We observe that 99.9% of the hypotheses (107,929 out of 108,078) belong to the lowest bin (\leq 20%) due to the tendency of genes to not confer resistance to antibiotics. In total, we validated 226 CRA hypotheses with varying probability among which 64 were validated as positives in the first iteration.

For the second iteration, we merged all 226 CRA hypotheses that we validated in the first iteration to expand the knowledge graph. This increased the size of the knowledge graph from 651,758 to 651,984. After training the hypothesis generator again using this updated knowledge graph, we generated 107,852 CRA hypotheses. We then followed the same validation process as discussed for the first iteration. In addition to binning the probability of the generated hypotheses from the two iterations into 5 bins (**Figure 3.5a**), **Supplementary Figure 20** shows the same analysis but with 10 bins.

A.1.3.10. Hypothesis generation results

From these two iterations of hypotheses generation, we computationally predicted and experimentally validated a total of 93 CRA hypotheses for 83 *E. coli* genes that confer resistance to one or more of 15 antibiotics (**Figure 3.5e**). Among these 93 CRA hypotheses, we found that 61 CRA hypotheses were inconsistencies that we did not merge during the inconsistency resolution process. In other words, we only validated and merged 236 inconsistencies out of 2,131 inconsistencies identified from level 3 (see **Appendix A.1.2.3.3**). As we did not merge the remaining 1,895 inconsistencies into the knowledge graph, the hypothesis generator generated hypotheses on these 1,895 inconsistencies. In future work, we expect to address this issue and produce better results.

A.1.3.11. Consistency of the KIDS-generated hypotheses

We tested the consistency of the hypotheses generated by KIDS using two metrics Kendall's tau¹⁶³ and rank-biased overlap (RBO)¹⁶⁴ that checks if two ranked lists are in agreement. Kendall's tau, a widely used correlation-based method, ranges between -1 and +1, where -1 denotes the complete disagreement and +1 denotes the complete agreement between the two ranked lists. However, some of its properties render its application to the KIDS-generated hypotheses less appropriate. For example, Kendall's tau requires two ranked lists to be of the same length, yet the number of hypotheses with probability > 0.20 changes for every run of hypotheses generation due to the stochastic nature of the hypotheses generation models. Moreover, it assigns equal weight to all the items in the ranked list, treating the agreement at the top of the lists (hypotheses with higher probability) as important as the agreement at the bottom of the lists (hypotheses with lower probability). RBO is an alternative method whose value ranges between 0 and 1, where 0 means complete disagreement and 1 means a complete agreement between

the two ranked lists. In contrast to Kendall's tau, RBO allows comparing two disjoint ranked lists of different lengths as well as putting more emphasis on the agreement at the top of the lists. In this work, we provide results for both metrics.

To this end, in addition to the original first iteration hypotheses, we generated 99 different versions of first iteration hypotheses each with unique random seeds. For Kendall's tau, as the length of the ranked lists needs to be identical, we found the rank of the original 149 original first iteration hypotheses with probability > 0.20 among the 99 versions of the hypotheses. Note that each ranked list contains 149 hypotheses. We then generated $\binom{100}{2} = 4,950$ pairs of ranked lists which was used to calculate Kendall's tau statistics. For the baseline, we randomly selected 149 hypotheses from 100 versions of randomly generated hypotheses, which were used to calculate Kendall's tau statistics similarly to above. For RBO, as the length of the ranked lists does not need to be identical, we used a more straightforward approach of extracting hypotheses with probability > 0.20 from each one of the 100 versions of first iteration hypotheses (including the original first iteration hypotheses). Similar to Kendall's tau approach, we generated $\binom{100}{2} = 4,950$ pairs of ranked lists which were used to calculate RBO statistics while setting the hyperparameter p to 0.99. For the baseline, we randomly selected varying numbers of hypotheses from 100 versions of randomly generated hypotheses. For example, if the number of hypotheses with probability > 0.20 in one version of the first iteration hypotheses was x , we also selected x number of hypotheses from the randomly generated hypotheses. The p -value between the KIDS-generated statistics (Kendall's tau or RBO) and the randomly generated hypotheses was calculated using the T-test with a two-sided alternative hypothesis. Finally, among the 2,907 that appeared at least once

among the 100 different versions of first iteration hypotheses with probability > 0.20, we identified 11 hypotheses that appeared in all 100 different versions from which 10 were validated to be a positive relationship.

A.1.3.12. Wet-lab validation

We profiled the antibiotic resistance response of 226 single-gene knockout strains of *E. coli* obtained from the Keio collection¹⁵⁴. Wild-type *E. coli* strain BW25113 was used as a control. For routine culturing, *E. coli* Wild-type cells were grown in LB medium, while knockouts were grown in media supplemented with 50 µg/ml kanamycin unless otherwise mentioned. To profile the antibiotic resistance response, fresh colonies of required strains were transferred to 96 well plates containing 200 µl of LB broth and grown for 8 hours at 37°C in an incubator shaker (BioTek HTX). Later, a fraction of the culture was transferred using a 96-pin replicator to a plate containing LB agar and the different amounts of antibiotics. The plate was incubated overnight at 37°C, and the next day absence or presence of colonies was recorded to identify the MICs. All experiments were performed in biological triplicate.

A.1.3.13. The similarity of novel ARGs to known ARGs

To show how similar the 6 novel ARGs (*ftsP*, *hdfR*, *Irp*, *proV*, *qorB*, and *rbsK*) are at the sequence level to already known ARGs, we downloaded the nucleotide sequence of the 4,577 ARGs from CARD¹¹³ (version 3.1.4) and performed BLASTN with our 6 ARGs (<https://blast.ncbi.nlm.nih.gov/Blast.cgi>). The highest-ranked matches are the following: CTX-M-204 of *Klebsiella pneumoniae* for *Irp* (100% identify, E-value=5.9); CMH-5 of *Enterobacter cloacae* for *rbsK* (88% identify, E-value=0.93); *cprS* of *Pseudomonas*

aeruginosa PAO1 for *qorB* (84.8% identify, E-value=0.86); MOX-9 of *Citrobacter freundii* for *hdfR* (100% identify, E-value=0.84); OXA-541 of *Pseudomonas putida* for *Irp* (91.7% identify, E-value=0.12); *ErmG* of *Bacteroides thetaiotaomicron* for *Irp* (88.9% identify, E-value=1.2). We also tested how similar the genes that we have predicted to confer no resistance using the hypothesis generator (probability range [0.0 and 0.2]) and further validated in the wet-lab to the ARGs in CARD. Out of the 129 genes, we found 11 genes that have a significant E-value (≤ 0.05).

A.1.3.14. Dissemination of novel ARGs across microbial communities

Using the MGnify service (<http://www.ebi.ac.uk/metagenomics>), we performed a protein sequence search of our 6 novel ARGs using HMMER (<http://hmmerr.org>) against their human digestive system microbiome database which contains 94,342 samples with a cutoff E-value set to ≤ 0.05 . We found how much dissemination these genes have in the database for 5 ARGs except for proV which ran into a server-side error (**Supplementary Table 12**).

A.1.3.15. Identification of bacteria harboring a maximum number of genes homologous to 6 novel CRA genes

We performed nucleotide mega-BLAST (<blast.ncbi.nlm.nih.gov/Blast.cgi>) to identify the genes homologous to 6 novel CRA genes in other bacterial genera. We found that *Salmonella* spp. had the maximum number of homologous genes. *Salmonella enterica* had 5 homologs *ftsP*, *Irp*, *proV*, *rbsK*, and *yifA* (*hdfR* in *E. coli*) with >78% similarity in nucleotide sequences, while homolog of *qorB* was not identified in *S. enterica*.

A.1.3.16. Construction of in-frame single-gene knockouts of the *S. enterica* LT2

We constructed 5 single-gene KO strains of *S. enterica* LT2 using λ -red recombinase system as described elsewhere¹⁵⁴. Briefly, we PCR amplified the kanamycin cassette from Keio-strain JW1869 using 5 sets of specially designed primers containing end sequences of kanamycin cassette and target genes (**Supplementary Table 13**). Electrocompetent *S. enterica* cells harboring pKD46 plasmid, temperature-sensitive and an ampicillin-resistant plasmid expressing λ -red recombinase system, were transformed with individual sets of PCR amplified kanamycin cassette to replace the 5 target genes with kanamycin marker. Cells were selected on an LB agar plate containing kanamycin and later pKD46 was removed from the cells by growing the cells in LB broth at 40°C¹⁵⁴. MICs of antibiotics cephadrine (for *ftsP* and *rbsK* KOs), geneticin (for *Irp* KO), chloramphenicol (for *proV* KO), and hygromycin B (for *yifA* KO) were measured as described in section 1.2.4. Wild-type *S. enterica* was always used as a reference strain.

A.2. Integrating knowledge graph structure in language models for link prediction

A.2.1. KG-BERT

KG-BERT⁷² is a fine-tuning method that utilizes the base version of the pre-trained language model BERT (BERT_{BASE})⁷⁰ as an encoder for entities and relations of the knowledge graph. Specifically, KG-BERT first converts a triplet (h, r, t) to a sequence of tokens $w^{(h,r,t)} = \langle [CLS]w_a^h[SEP]w_b^r[SEP]w_c^t[SEP]: a \in \{1..|h|\} \& b \in \{1..|r|\} \& c \in \{1..|t|\} \rangle$, where w_n denotes the n th token of either entity or relation, $[CLS]$ and $[SEP]$ are the special tokens, while $|h|$, $|r|$, and $|t|$ denote the number of tokens in the head entity, relation, and tail entity, respectively. This textual token sequence is then converted to a sequence of token embeddings $\mathbf{w}^{(h,r,t)} \in \mathbb{R}^{d \times (|h|+|r|+|t|+4)}$, where d is the dimension of the embeddings and 4 is from the special tokens. Then the segment embeddings $\mathbf{s}^{(h,r,t)} = \langle (\mathbf{s}_e)_{\times(|h|+2)} (\mathbf{s}_r)_{\times(|r|+1)} (\mathbf{s}_e)_{\times(|t|+1)} \rangle$, where \mathbf{s}_e and \mathbf{s}_r are used to differentiate entities from relations, respectively, as well as the position embeddings $\mathbf{p}^{(h,r,t)} = \langle \mathbf{p}_i: i \in \{1..(|h|+|r|+|t|+4)\} \rangle$ are added to the token embeddings $\mathbf{w}^{(h,r,t)}$ to form a final input representation $\mathbf{X}^{(h,r,t)} \in \mathbb{R}^{d \times (|h|+|r|+|t|+4)}$ that is fed to BERT as input. Then, the score of how likely a given triplet (h, r, t) is to be true is computed by

$$score_{KG-BERT}(h, r, t) = SeqCls(\mathbf{X}^{(h,r,t)}).$$

KG-BERT significantly improved the MR of the link prediction task compared to the previous state-of-the-art approach CapsE³⁰³ (97 compared to 719, an 86.5% decrease),

but suffered from poor hits@1 of 0.041 due to the entity ambiguity problem and lack of structural learning^{74,75}.

A.2.2. StAR

StAR⁷⁴ is a hybrid model that learns both the contextual and structural information of the knowledge graph by augmenting the structured knowledge in the encoder. It divides a triplet into two parts, (h, r) and (t) , and applies a Siamese-style transformer with a sequence classification head to generate $\mathbf{u} = Pool(\mathbf{X}^{(h,r)}) \in \mathbb{R}^{d \times (|h|+|r|+3)}$ and $\mathbf{v} = Pool(\mathbf{X}^{(t)}) \in \mathbb{R}^{d \times (|t|+2)}$, respectively, where $Pool(\cdot)$ is the output of the RoBERTa's pooling layer. The first scoring module focuses on classifying the triplet by applying a

$$score_{StAR}^c(h, r, t) = Cls([\mathbf{u}, \mathbf{u} \times \mathbf{v}; \mathbf{u} - \mathbf{u}; \mathbf{v}]),$$

where $Cls(\cdot)$ is a neural binary classifier with a dense layer followed by a softmax activation function. The second scoring module then adopts the idea of how translation-based graph embedding methods like TransE learn the graph structure by minimizing the distance between \mathbf{u} and \mathbf{v} as

$$score_{StAR}^d(h, r, t) = -\|\mathbf{u} - \mathbf{v}\|,$$

where $\|\cdot\|$ is the L_2 -normalization. During the training, StAR uses a weighted average of the binary cross entropy loss computed using $score_{StAR}^c(h, r, t)$ and the margin-based hinge loss computed using $score_{StAR}^d(h, r, t)$, whereas only the $score_{StAR}^c(h, r, t)$ is used for inference. This approach shows a new state-of-the performance over the metrics MR (51) and hits@10 (0.709), as well as significantly improving the hits@1 compared to the KG-BERT (0.041 to 0.243, a 492.7% increase).

A.3. Automated knowledge extraction of food and chemicals from the literature

A.3.1. Additional analysis

A.3.1.1. Percentage of associations with literature citations in FooDB

We downloaded *foodb_2020_04_07_csv/Content.csv* from the FooDB (<https://foodb.ca>) website and analyzed *citation_type* and *citation* columns, which represent a generalized description and a more specific identifier for a citation of a given food-chemical association. Among 5,145,532 FooDB associations, 3,273,562 are classified as *PREDICTED* by *PATHBANK* or *HMDB*. 955,962 associations are noted as *MANUAL* without a citation. Finally, 895,467 associations are linked to other databases without specific references to scientific literature. Therefore, approximately 20,541 associations have a scientific literature reference in FooDB, which is 0.4% of the total number of associations.

A.3.1.2. Estimation of the total number of triplets in PubMed Central

To estimate all food-chemical associations available to extract in scientific literature, we extrapolated the total number of triplets extractable from PubMed using the statistics of the FoodAtlas knowledge graph (FAKG). First, we estimated the expected number of unique triplets extracted from each publication, denoted as m , by having the unique number of triplets extracted from PH pairs in FAKG divided by the unique number of PMIDs. While calculating, since most triplets resulted from FoodAtlas entailment predictions, we weighted each triplet based on the prediction probabilities,

$$m = \frac{1}{|L|} \sum_{t \in T} 1 \cdot p_t = \frac{1}{146,507} \cdot 193,310.42 = 1.32,$$

where L is the set of PMIDs in FAKG, T is the set of triplets, and p_t is the entailment probability for the triplet t . Then, we counted the number of papers relevant to food in PubMed using the advanced search query “(food) OR (fruit) OR (vegetable)”, which returned 1,588,596 unique PMIDs. With this information, we estimated that our pipeline would be able to extract $1.32 \times 1,588,596 \approx 2.1$ M triplets from unstructured text alone, not including tables, supplementary files, etc.

A.3.1.3. Lit2KG generated triplets vs. external database triplets

The Lit2KG pipeline integrated, on average, 39% of the contains triplets from the external databases (323 out of 529 FDC triplets, 371 out of 1,055 Phenol-Explorer triplets, and 1,873 out of 8,375 Frida triplets were also discovered by Lit2KG pipeline) (**Figure 5.2d**). We also identified 219,854 medium-quality triplets that were not reported in external databases (**Figure 5.2e**).

A.3.1.4. Validation using FoodMine

To evaluate the chemical coverage of FAKG, we compared chemicals in FAKG to that in FoodMine³⁰⁴, a database exhaustively curating chemicals for cocoa and garlic from the literature. In this section, we describe the methodology of the validation.

A.3.1.4.1. Processing chemicals in FoodMine

FoodMine initially contained 598 and 289 unique chemicals for cocoa and garlic, respectively. However, due to different chemical indexing methods, the chemicals in FoodMine were not directly comparable to those in FAKG. In the FoodMine paper, the

authors describe their method for chemical disambiguation using *structural code*, *i.e.*, the first 14 characters of InChIKey³⁰⁴. To be comparable, we follow the same approach as FoodMine did. Since every chemical in FAKG has a PubChem CID, we retrieved InChiKey for each chemical and derived its structural code. Finally, we dropped chemicals without structural code in FoodMine, resulting in 301 and 176 unique chemicals for cocoa and garlic, respectively.

A.3.1.4.2. Processing chemicals in FAKG

The processing required for FAKG was more complicated than that for FoodMine. There were four types of evidence in FAKG: entailment annotation, entailment prediction, link prediction, and external databases. We treated entailment annotation and external database evidence as positive evidence supporting a food containing a chemical. For entailment prediction, if we filtered by the NCBI Taxonomy ID, FAKG had 1,289 and 1,376 chemicals with unique PubChem CIDs for cocoa and garlic, respectively. Then, as mentioned in the section above, we retrieved the structural code for each chemical, resulting in 499 and 691 unique structural codes for cocoa and garlic, respectively. For a fair comparison between FAKG and FoodMine, we manually annotated and excluded the false positives and NER errors of entailment prediction. To do that, we collected all the entailment prediction evidence in FAKG for all structural codes, resulting in 5,506 and 9,616 pieces of evidence for cocoa and garlic, respectively. Note that LitSense also tagged “chocolate” as cocoa, so 5,506 evidence were dropped. In this case, we were only interested in the existence of entailment prediction evidence regarding cocoa or garlic containing a structural code. Thus, once we encountered one piece of positive evidence, we skipped the rest of the evidence associated with that structural code, which means we

discarded a structural code only if all evidence was false positive. Eventually, FAKG contained 366 and 400 unique structure codes for cocoa and garlic, respectively, supported by the entailment prediction evidence. For link prediction evidence, we annotated the structural codes following **Appendix A.3.1.8**, which resulted in 14 and 8 unique chemicals for cocoa and garlic, respectively. After removing the duplicated structural codes between entailment and link prediction, we had 379 and 406 chemicals with unique structural codes.

A.3.1.4.3. Limitations

There were limitations in this comparison. First, FAKG had the advantage in this comparison because FoodMine mainly targets quantified chemicals, but FAKG curates any chemical identified in foods. Second, we adopted the same methodology of FoodMine to match the chemicals in FoodMine and FAKG. However, the method considered chemicals within the same isomeric species identical by only looking at the segment InChIKeys. Lastly, as mentioned in the last section, FAKG focuses on raw food, while FoodMine does not discriminate between raw and non-raw food.

A.3.1.5. Active learning performance

We calculated the performance of the entailment model at the final AL round ($r = 10$) by averaging all 400 models (100 random seeds for each of the 4 AL strategies trained on the same data). Compared to the initial round ($r = 1$), precision at the final round ($r = 10$) increased by 10.8% (0.74 ± 0.04 vs. 0.82 ± 0.03 , respectively, p -value = 1.3×10^{-155}), recall by 21.7% (0.69 ± 0.11 vs. 0.84 ± 0.07 , respectively, p -value = 3.7×10^{-89}), and F1

score by 16.9% (0.71 ± 0.05 vs. 0.83 ± 0.03 , respectively, p -value = 2.8×10^{-206}) (**Supplementary Figure 26** and **Supplementary Table 14**).

A.3.1.6. Understanding the entailment predictions

Table 5.1 shows selected entailment model predictions for each of the four outcome scenarios, true positive (TP), false positive (FP), true negative (TN), and false negative (FN), of the confusion matrix. For certain TP and TN ($p \approx 1$ and $p \approx 0$, respectively), PH pairs are usually straightforward, such as premises with simple sentence structure (index 1) or hypotheses with wrong food and food part matching (index 3). For certain FP and FN and uncertain predictions, we found that premises often require the models to use domain expertise to predict correctly (e.g., *essential nutrient* implies chemicals may not naturally occur in species as in index 5) or are titles of the paper that are hypotheses yet to be validated (index 6). Uncertain predictions ($p \approx 0.5$) had a higher standard deviation of the probability scores assigned by the 400 entailment models at the final round than the certain predictions (**Table 5.1**).

A.3.1.7. Entailment model performance comparison across literature sections

We compared how sentences from different literature sections affected the production entailment model performance. Since the production model no longer had a designated test set, we performed 10-fold cross-validation to get the prediction score for each PH pair using the same train and validation splits and best hyperparameter described in **Appendix A.3.2.3.2**. To further increase the robustness, we repeated the cross-validation 50 times with different weight initializations such that the prediction score for each PH pair was the average of 50 runs. **Supplementary Table 15** summarizes the confusion

matrices for each section. We found that for the PH pairs from the introduction section, the model achieved the highest precision, F1, and the third-best recall scores. Also, regarding prediction scores, the model predicted the PH pairs of the introduction section significantly differently from those of the other sections (vs. Discussion, p -value = 7.8×10^{-33} ; vs. Abstract, p -value = 3.2×10^{-104} ; vs. Methods, p -value = 6.7×10^{-206} ; vs. Results, p -value = 1.8×10^{-13} ; vs. Conclusion, p -value = 9.4×10^{-4} ; vs. Title, p -value = 4.3×10^{-18} ; vs. Table, p -value = 1.1×10^{-23} ; vs. Others, p -value = 1.0×10^{-34} . p -values were computed by the two-sided t-test with the Benjamini-Hochberg adjustment³⁰⁵).

A.3.1.8. Validation for link prediction

To validate the utility of link prediction in discovering novel food-chemical relations, we manually annotated the triplets predicted by the best link prediction (LP) model. Because the inputs to LP were merely triplets (i.e., no longer using premise-hypothesis pairs), finding and annotating the evidence for link prediction triplets proved challenging without domain expertise. Therefore, we seek help from a postdoctoral biochemistry expert to annotate the triplets for LP validation. For this section, we will discuss the procedure of query, search, and annotation.

A.3.1.8.1. Annotation datasets

To validate that the link predictor was well-calibrated, we annotated all 466 triplets with probability scores greater than 90% as well as 200 randomly sampled triplets with 20 triplets from each of the 10 equally spaced bins. The reason for selecting these two sets was because the former would allow the discovery of novel food-chemical associations by looking at the most likely triplets, while the latter would show how well our model is

calibrated based on a scrutinized literature search. However, note that we skipped some unwanted triplets, including triplets with entities with LitSense NER Error and triplets with chemical entities that were chemical groups rather than a specific chemical structure.

A.3.1.8.2. Query standard

We searched four sources, which are PubChem taxonomy²⁴², Bing Chat, Google Scholar, and Google. We queried PubChem taxonomy and Bing Chat because they returned more straightforward responses compared to Google Scholar and Google, which had better coverage.

For PubChem taxonomy, we queried the scientific name of the food entity of the triplet to retrieve the evidence. Since PubChem taxonomy retrieved evidence based on fuzzy search²⁴², we reviewed and verified the evidence.

For Bing Chat, we queried the search engine by prompting, “Does *Theobroma cacao* contain stearic acid?” where synonyms for foods and chemicals were also used to query independently. Specifically, for food entities, we used all the synonyms listed in the NCBI taxonomy, while for chemical entities, we tried a subset of synonyms that were most commonly used (**Supplementary Figure 31**). We checked the evidence links returned by Bing Chat and verified their correctness.

For Google Scholar and Google, we formulated the query by concatenating the food and chemical names, e.g., *Theobroma cacao* stearic acid. The same treatment for synonyms performed in Bing Chat was applied.

A.3.1.8.3. Annotation standard

If we found the evidence, we annotated the triplets as *Yes*, and the reference sources were recorded. The triplets were annotated as *Unknown* if we found no evidence since we could not conclude that the corresponding triplet was novel without exhaustively searching the entire internet. Finally, a few triplets were annotated as *No*, if food could not biologically contain a chemical. For example, a triplet with the chemical *1,1,1-Trichloroethane* was labeled as *No* because the chemical could only be synthesized in labs.

A.3.1.8.4. Validation for *Unknown* triplets

Unknown triplets, described in the last section, not being found on the internet could be either novel or wrong. Since experimentally validating triplets would be expensive, we describe a heuristic to decide if a triplet is likely true based on its metabolic and evolutionary viability, where we consider a triplet is likely to be true if it passes at least one of the tests. We start this section with a description of our tests, followed by a specific discussion for each triplet that has passed the test.

Metabolic viability test. A chemical is likely to present in a species if a metabolic pathway exists within the species to produce that chemical, and we utilized the following approaches to identify the pathways likely to exist within species. First, we checked if a species can synthesize enzymes known to produce the chemical of interest. To do this, we searched for the enzymes of the chemical in databases, like KEGG³⁰⁶, and literature. Then, we retrieved the enzyme amino acid sequence from UniProt³⁰⁷. Finally, we input the retrieved enzyme sequences and the species of interest in the NCBI BLAST³⁰⁸ query to find the homologous enzymes with similar sequences producible by the species. Based on the assumption that similar sequences share a similar function, we considered the

triplet passing the metabolomic viability test if the searched enzymes were homologous with the query sequence by hitting the *E*-value of 0, implying that the observed similarity between the query and subject sequences was highly unlikely to have arisen randomly or by chance. Note that not all chemical synthesis requires enzymes. In those cases, we looked for the chemical equation that produced the chemical and verified whether the required reactants and catalysts were found in the species. If found, we also considered the triplet passing the test.

Evolutionary viability test. The metabolic viability test could be challenging for some species because they were not found in KEGG or BLAST. In such cases, we resorted to an evolutionary viability test. Specifically, a triplet is considered to pass the test if there are any species under the same genus of the species of interest containing the chemical. For example, we could not find any metabolic pathway for phenol in *Cinnamomum aromaticum* (Chinese cinnamon), but we found phenol in *Cinnamomum zeylanicum* (Ceylon cinnamon)³⁰⁹, and hence the triplet passed the evolutionary viability test.

Metabolically viable triplet 1: *Hericium erinacerus* (bearded tooth) contains lumisterol. We found a recent source indicating that lumisterol can be naturally synthesized with ergosterol by UV irradiation (e.g., sunlight)³¹⁰, where the bearded tooth is known to contain ergosterol³¹¹.

Metabolically viable triplet 2: *Cicer arietinum* (chickpea) contains triglyceride. Through a literature search, we found an enzyme, phospholipid:diacylglycerol acyltransferase, that could synthesize triacylglycerol (synonym for triglyceride)³¹², and found its protein sequence (UniProt ID: Q9FNA9) for *Arabidopsis thaliana* (mouse-ear cress). Through BLAST (blastp), four phospholipid:diacylglycerol acyltransferase-like

enzymes in chickpeas with zero *E*-value and 78.4%, 76.1%, 76.7%, and 56.9% percentage of identity (RefSeq ID: XP_004510434.1, XP_004507978.1, XP_012573333.1, and XP_004506121.1) were retrieved.

Metabolically viable triplet 3: *Cuminum cyminum* (cumin) contains sodium caffeate.

Sodium caffeate can be synthesized naturally by caffeic acid and sodium through a neutralization process without enzymes. Caffeic acid can be found in cumin³¹³ while sodium is present in all plants.

Metabolically viable triplet 4: *Gadus morhua* (Atlantic cod) contains beta-carotene.

Through a literature search, we found beta-carotene-15,15'-dioxygenase, an enzyme in *Homo sapiens* (human) that catalyzes the reaction involved in beta-carotene³¹⁴. We used the corresponding protein sequence (UniProt ID: Q9HAY6) and identified two protein sequences through BLAST (blastp) with zero *E*-values and percentage of identities of 58.5% and 53.9% respectively (RefSeq ID: XP_030205463.1 and XP_030208886.1).

Evolutionarily viable triplet 1: *Cucumis melo* var. *dudaim* (Dudaim melon) contains matairesinol.

Secoisolariciresinol dehydrogenase is an enzyme that can produce matairesinol and is found in various plants³¹⁵. Although we could not find a source that directly indicates *Cucumis melo* var. *dudaim* contains this enzyme, we found that *Cucumis melo* (muskmelon) and *Cucumis melo* var. *makuwa* (oriental melon), two species sharing the same genus with *Cucumis melo* var. *dudaim*, contain secoisolariciresinol dehydrogenase (UniProt ID: A0A1S3CT49 and A0A5A7UNV0). Since the species *Cucumis melo* and a variant *Cucumis melo* var. *makuwa* (oriental melon) that shares the same parent subspecies (*Cucumis melo* subsp. *agrestis*) as *Cucumis melo*

var. dudaim contains secoisolariciresinol dehydrogenase, likely, *Cucumis melo var. dudaim* and all *Cucumis melo*'s subspecies and variants contain this enzyme.

While there is a reference genome in BLAST for *Cucumis melo var. dudaim*, currently the only source for this sequencing data analyzes whole chloroplast gene assembly and is currently unpublished (<https://www.ncbi.nlm.nih.gov/nuccore/1240947550>), meaning that the proteome and genome are currently incomplete. Additionally, using BLAST (TBLASTN) to search for the DNA sequence for secoisolariciresinol dehydrogenase (RefSeq ID: XP_008467261.1) in *Cucumis melo*, we found the top two non-predicted sequences with E-values of 10^{-180} and percent identities of 96.48% (RefSeq ID: LN681895.1 and LN713263.1). The first reference sequence is a scaffold, but the second says it is located on chromosome 9 of the *Cucumis melo* nuclear genome, and the chloroplast genome has little similarity to the nuclear genome, especially chromosome 9³¹⁶. Therefore, given the current sequencing data for *Cucumis melo var. dudaim*, we are not able to find secoisolariciresinol dehydrogenase because the current sequencing data is incomplete and the current genomic data does not contain the region that has secoisolariciresinol dehydrogenase.

Evolutionarily viable triplet 2: *Cinnamomum aromaticum* (Chinese cinnamon) contains phenol. Phenol can be found in *Cinnamomum zeylanicum* (Ceylon cinnamon)³⁰⁹, which shares the same genus, *Cinnamomum*, with Chinese cinnamon.

5 non-passing triplets. *Diospyros kaki* (Japanese persimmon) contains 3-Rhamnosyl-Glucosyl Quercetin, *Anthriscus cerefolium* (chervil) contains loxoprofen, *Allium schoenoprasum* (chive) contains salicylic acid, *Juglans cinerea* (butternut) contains lawsone, and *Curcuma longa* (turmeric) contains 3-Rhamnosyl-Glucosyl Quercetin.

A.3.1.9. Benchmark with FooDB.

We benchmarked with FooDB, the existing state-of-the-art food resource that aggregates many other popular food databases, and by comparing with FooDB, we would be able to estimate the overall coverage of FoodAtlas concerning all the existing food sources.

Supplementary Figure 25 shows the comparison between FoodAtlas and FooDB. For the two databases to be comparable, we only considered foods and chemicals associated with NCBI Taxonomy ID (NCBI ID) and PubChem CID.

For FoodAtlas, we first extracted all the triplets with *contains* relation, resulting in 243,231 unique triplets. Within them, we counted 536 and 11,908 unique NCBI IDs and CIDs, forming 126,082 triplets with unique NCBI ID-CID pairs. The count of unique pairs was smaller than the number of triplets because different food parts were considered different triplets while sharing the same NCBI ID. Among them, 106,082 are exclusively included by FoodAtlas. To further investigate the distribution of quality of the 106,082 triplets, we retrieved all the evidence stored in FoodAtlas for the associated triplets and assigned each triplet with the highest evidence quality. For example, if a triplet had high- and medium-quality evidence simultaneously, the triplet was considered a high-quality triplet. This gave us 2,091 high-, 94,095 medium-, and 9,896 low-quality triplets, adding to 106,082.

For FooDB, we downloaded the entire database with 5,007,500 food-chemical associations. Out of 992 original foods, 600 unique NCBI IDs were found. For 70,477 original chemicals in FooDB, however, CIDs were not reported in the downloadable content. To accommodate, we used InChIKeys reported in FooDB to identify the corresponding CIDs with PucChem Identifier Exchange Service

(<https://pubchem.ncbi.nlm.nih.gov/docs/identifier-exchange-service>). This process returned 64,125 unique CIDs in FooDB. We then extracted food-chemical pairs that were associated with using these 600 and 64,125 NCBI IDs and CIDs in FooDB, resulting in 2,480,768 associations out of the original 5,007,500. After dropping NCBI IDs and CIDs without any association, FooDB had 598 and 51,130 unique NCBI IDs and CIDs, forming 2,480,768 associations.

To make a more fine-grained comparison, we also evaluated FooDB without associations predicted via metabolic pathways. This included associations imported from PathBank³¹⁷ and HMDB³¹⁸. This resulted in 651,680 unique associations compared to the previous 2,480,768.

A.3.1.10. Knowledge extraction using GPT3.5 and GPT4

We tested the performance of GPT3.5 and GPT4 on the knowledge extraction task. Instead of using the original BioBERT entailment model scheme which uses the premise (sentence) as evidence to predict the entailment label True or False, the GPT-based method uses an engineered prompt along with the sentence directly to extract knowledge as a chat completion. To benchmark the performance of GPT-extracted knowledge, we created two datasets. Both versions of the datasets contain randomly selected 100 unique sentences from the pool of 4,120 Lit2KG premise-hypothesis pairs. However, the first version did not contain any concentration value, whereas the second version did.

Benchmarking without concentration value. For sentences without concentration values, we used the following single-shot prompt to extract knowledge using GPT3.5 and GPT4.

Given a sentence, extract in CSV format the following: food, food part, and chemical. Food and chemical must exist to be a valid entry. Food part can be left empty if not found. Do not return anything if none found. Oil should be included in the food, not food part. For example, given a sentence "Olive leaf extract contains phenols and flavonoids." extract the following: \n olive, leaf, phenols\n olive, leaf, flavonoids\n \n\nSentence:

We then compared the extracted results to the manually annotated tuples of format (food, food part, chemical) as well as to the Lit2KG-extracted triplets. We assigned three outcomes TP, FP, and FN. TP is a case when the prediction (Lit2KG, GPT3.5, or GPT4) strictly matched the ground-truth (GT) annotation, FP is a case when the prediction did not match the GT annotation, and FN is a case when the GT annotation was not matched to any predictions. Note that we do not have any TN cases.

Benchmarking with concentration value. For sentences with concentration values, we used the following single-shot prompt to extract knowledge using GPT3.5 and GPT4.

Given a sentence, extract in CSV format the following: food, food part, chemical, and chemical concentration. Food and chemical must exist to be a valid entry. Food part and chemical concentration can be left empty if not found. Do not return anything if none found. Oil should be included in the food, not food part. For example, given a sentence "Total phenols and flavonoids in the olive leaf extract were 169.10 ± 0.57 mg/g and 98.15 ± 0.7 mg/g, respectively." extract the following: \nolive, leaf, phenols, 169.10 ± 0.57 mg/g\nolive, leaf, flavonoids, 98.15 ± 0.7 mg/g\n \n\nSentence:

The key difference of this benchmark dataset was that the manual annotation included chemical concentration fields as follows: (food, food part, chemical, chemical concentration). The evaluation process was the same as the dataset without concentration values. This was a harder task for Lit2KG model prediction as the Lit2KG did not extract concentration values, and even the true prediction (without concentration) was considered an FP.

A.3.2. Methods

A.3.2.1. Food name collection and LitSense query

We collected 650 unique NCBI Taxonomy IDs (NCBI IDs) from FooDB (<https://foodb.ca>), FDC²¹⁶, Phenol-Explorer^{37,224,225}, and Frida (<https://frida.fooddata.dk>). Note that the FoodAtlas framework requires each food item to have an NCBI ID; thus, we discarded any food items in these databases without NCBI IDs. After downloading the entire NCBI Taxonomy database from the FTP site (https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/new_taxdump; accession time: 9:53 AM on November 30th, 2022), we found the common name, Genbank name, and scientific name entries of the 650 NCBI IDs in the downloaded file, *names.dmp*, which resulted in 1,959 food names.

A.3.2.2. Data annotation

We deployed an annotation platform with a graphical user interface on Label Studio. We had two annotators labeled each premise-hypothesis (PH) pair as one of *entails* (i.e., the premise supports the hypothesis), *does not entail* (i.e., the premise does not support the

hypothesis), and *skip* (which will be defined later in the paragraph). The annotators did a preliminary annotation session and reported some observations, which we used to create basic annotation standards for annotators to follow for the following annotation sessions. We decided that for a PH pair, (a) if LitSense returned an incorrect NCBI Taxonomy ID or MeSH ID for the hypothesis, then the PH pair should be *skip*, and (b) the annotation should not rely on sources other than premise, i.e., even though it is well-known that lemon contains vitamin C, but if the premise does not support the claim, the PH pair should be *does not entail*. Note that we desired the annotation standard to be as general as possible to avoid biasing the subsequent training of the entailment model. We ensured the annotation quality by only using the PH pairs whose annotated labels were the consensus among annotators.

A.3.2.3. Entailment model

A.3.2.3.1. Model configuration

We used BioBERT²²⁹, a pre-trained language model pre-trained on the biomedical corpus, and the English Wikipedia and BooksCorpus, as our entailment model. Specifically, we imported the pre-trained model, dmis-lab/biobert-v1.1, from HuggingFace³¹⁹, with the default setup, except that we changed the pad token type ID to 1 to accommodate sentence-pair classification schema. We implemented the entailment model with PyTorch³²⁰, a deep-learning framework. We performed the grid search hyperparameter tuning over the batch size, learning rate of the AdamW³²¹ optimizer, and the number of epochs. When an input sentence pair exceeded the maximum length of 512, we truncated the longer sentence, which was always the first sentence, i.e., premise. We relied on four Nvidia RTX A5000 GPUs for the training and inference.

A.3.2.3.2. Production model

Once we completed the evaluation for entailment models with different active learning strategies, we deployed the production entailment model, where we used the entire labeled dataset (*i.e.*, training, validation, and test sets) to train it. Then, to configure the optimal hyperparameters for the production model, we performed a 10-fold cross-validation, where each fold contained a unique set of premises and a unique set of hypotheses. Again, we used the same grid search space, *i.e.*, learning rate = $\{2 \times 10^{-5}, 5 \times 10^{-5}\}$, epochs = $\{3, 4\}$, and batch size = $\{16, 32\}$, described in the main method section, where the one with the best mean precision across ten folds was chosen for the production model. Finally, using the best hyperparameter set, we trained our production model, an ensemble system, by training 100 entailment models with different random initializations using the entire dataset (*i.e.*, all ten folds). Given one input premise-hypothesis (PH) pair, the prediction of the production model was then the mean of predicted scores of the 100 entailment models.

A.3.2.3.3. Active learning

The available scientific literature online is too large to annotate manually. Thus, we experimented with active learning, a subfield of machine learning that aims the models to choose the training data to be labeled for the models to achieve better performance with less labeled data³²². To compare different active learning strategies without overly burdensome manual annotation labor, we employed a *pool-based active learning* scenario, which assumes a small subset of labeled data exists in a closed pool of data, allowing the model to select new training samples from the pool³²². Here, we describe more details regarding each active learning strategy.

Maximum Likelihood Active Learning. The maximum likelihood active learning continuously sampled the PH pairs with the most certainly positive probability scores. Because we only added the positives to the knowledge graph, so we hypothesized that the knowledge graph would grow the fastest with this active learning. Specifically, we sorted unsampled PH pairs based on their predicted probability scores. We then added the highest 412 PH pairs to the sampled data for the next active learning round training.

Maximum Entropy Active Learning. The maximum entropy active learning sampled the PH pairs that the model was most uncertain about. The PH pairs with uncertain predictions were close to the decision boundary of the model, so we hypothesized that the entailment model would benefit from learning these data. For binary classification, this means,

$$uncertainty = \min(1 - p, p),$$

where p is the probability score associated with the entailment prediction. We can observe that the uncertainty score is the highest when $p = 0.5$.

Stratified Active Learning. The stratified active learning split the data pool into ten bins of equal interval (i.e., $[0, 0.1]$, $[0.1, 0.2]$, ..., $[0.9, 1.0]$), and the entailment models selected the same number of samples from each bin randomly. However, especially for the late active learning rounds, the number of samples in a bin might be less than the number of samples the model required to draw. Therefore, we ensured to train the model with the same amount of data in each round by using the following simple algorithm for each stratified active learning round sampling:

1. Initialize the number of bins $B = 10$ and the total number of samples drawn per round $N = 412$.

2. Split the data pool into B equal-interval bins.
3. Check if, for all B bins, there exists at least (N / B) samples:
 - a. If true, randomly draw (N / B) samples from each bin.
 - b. If false, update $B \leftarrow B - 1$, and start from step 2.

Random Sampling. Random sampling is our baseline equivalent to no active learning.

Every PH pair has an equal chance to be sampled by the model.

A.3.2.3.4. Pseudocode

Algorithm 1. Training an entailment model with pool-based active learning (AL).

Input: $\{X_A, y_A, X_T, H\}$, where X_A is a set of annotated PH pairs for training the model, y_A is an array of labels of X_A , X_T is a set of PH pairs predicted by the trained model, and H is the hyperparameter search space.

Output: $\{X_T, \widehat{y}_T\}$, where X_T is the same as the input, and \widehat{y}_T is an array of labels of X_T predicted by the model across multiple AL rounds.

```

1:   $(X_{train}, y_{train}), (X_{val}, y_{val}) \leftarrow (X_A, y_A)$            // Split the PH pairs
2:  for  $j = 1$  to  $R$  do                                           // Run  $R$  AL rounds
3:    if  $j == 1$  do
4:       $X_s, y_s \leftarrow sample\_randomly(X_{train}, y_{train})$        // Section 1.1.3.3
5:       $X_r, y_r \leftarrow (X_{train}, y_{train}) \setminus (X_s, y_s)$    // Get the remaining set
6:    else do
7:       $X'_s, y'_s \leftarrow sample\_actively(model, X_r, y_r)$        // Section 1.1.3.3
8:       $X_s, y_s \leftarrow (X_s, y_s) \cup (X'_s, y'_s)$              // Update sampled training set
9:       $X_r, y_r \leftarrow (X_r, y_r) \setminus (X_s, y_s)$            // Get the remaining set
10:   end if
11:    $Model \leftarrow tuning\_heldout(X_s, y_s, X_{val}, y_{val}, H)$    // Select the best model
12:    $\widehat{y}_{T_j} \leftarrow Model(X_T)$ 
13: end for
14:  $\widehat{y}_T \leftarrow \{\widehat{y}_{T_1}, \widehat{y}_{T_2}, \dots, \widehat{y}_{T_R}\}$ 
15: return  $\{X_T, \widehat{y}_T\}$ 

```

Algorithm 2. Training the production entailment model and predicting unannotated PH pairs.

Input: $\{X_A, y_A, X_U, H\}$, where X_A and y_A are the annotated PH pairs and their labels, X_U is the unannotated PH pairs, and H is the hyperparameter search space.

Output: $\{X_U, \widehat{y}_U\}$, where X_U and \widehat{y}_U are the unannotated PH pairs and their corresponding predictions by the production model.

```
1:  hparam  $\leftarrow$  tuning_10_fold( $X_A, y_A, H$ )           // Run grid search
2:  for  $i = 1$  to  $S$  do                               // Run  $S$  different seeds
3:     $\theta_i \leftarrow$  initialize_random_weights(hparam)
4:     $Model_i \leftarrow$  train_model( $X_A, y_A, \theta_i$ )
5:     $\widehat{y}_{U_i} \leftarrow Model_i(X_U)$                 // Predict unannotated input
6:  end for
7:   $\widehat{y}_U \leftarrow$  get_elementwise_mean( $[\widehat{y}_{U_1}, \widehat{y}_{U_2}, \dots, \widehat{y}_{U_S}]$ ) // Do ensemble predictions
8:  Return  $\{X_U, \widehat{y}_U\}$ 
```

A.3.2.4. Link prediction

A.3.2.4.1. Model selection dataset generation

To evaluate different versions of the FAKG in **Figure 5.5b** fairly, we created a held-out validation and test set by randomly sampling the triplets of type (food, contains, chemical) from the annotated PH pairs used for the entailment model. Note that we only added the (food, contains, chemical) triplet type, but not the (food part, contains, chemical) triplet type, as we are interested in generating hypotheses of the former. This results in the validation set with 445 positives and the test set with 447 triplets. As for the training data, although the data source varies for each version of the FAKG, the generation process is identical for all. We start by making sure that the 892 (food, contains, chemical) triplets in the validation and test set are removed from the available data. Next, we also remove any (food part, contains, chemical) in the training set that shares the same NCBI

taxonomy ID as food entities in the validation and test set. For example, if (strawberry, contains, Ascorbic Acid) is in the validation set, we remove any (strawberry {*part*}, contains, Ascorbic Acid) triplets in the training set. We chose to do this exclusion as we wanted to prevent the model from learning the chemical composition of foods only from their food part chemical composition.

A.3.2.4.2. Hyperparameter optimization

We performed the hyperparameter optimization of the six link prediction models, TransE²³⁶, ER-MLP²³⁷, DistMult²³⁸, TransD²³⁹, ComplEx²⁴⁰, and RotatE⁶⁹, using the metric MR on the validation set. For each model and each version of the dataset in **Figure 5.5b**, we tested 50 different sets of hyperparameters, where each hyperparameter set was drawn randomly from the default pool of hyperparameters of the PyKEEN library²³⁵ that were chosen from the best-reported values in each model's original paper. Early stopping was also used during the hyperparameter optimization process.

A.3.2.4.3. OpenAI GPT

We tested how the vanilla GPT-3.5 model (text-davinci-003)³²³ performs without any training data and how it compares to the graph-embedding models. We used the (food, contains, chemical) triplets in the test set to ask the question “Does {*food*} contains {*chemical*}?” using the text completion endpoint, where the scientific name was used for food, and the PubChem name was used for the chemical (MeSH name was used for chemical in case PubChem entry did not exist). We prompted the model to generate five different versions of the answer to test the statistical significance, similar to what we did

with graph-embedding link prediction models. Surprisingly, GPT had a precision of 64.8%, although the recall and F1 score was lower at 31.8% and 42.7%, respectively.

A.3.3. FoodAtlas Knowledge Graph

A.3.3.1. Entity types

The current version of FoodAtlas KG (FAKG) supports three entity types: *cellular organism (food)*, *food part*, and *chemical* (**Supplementary Table 17**). All entities in FAKG are assigned a unique FoodAtlas ID, a sequentially assigned numerical ID prefixed with the letter *e*. In this section, we describe each entity type in more detail.

A.3.3.1.1. cellular organism (food)

In addition to the unique FoodAtlas ID, *cellular organism* entities have one unique NCBI Taxonomy ID (NCBI ID) and part ID of *p0*, corresponding to the whole food, in contrast to the food parts in the next section. We require all organisms to have a valid NCBI ID to merge external databases and resolve synonyms. Although we interchangeably use the terminology *cellular organism* and *food*, it is worth noting that not all organisms are foods. For example, an entity *strawberry* with a rank *species* in the taxonomic lineage has a parent entity *Fragaria* with a rank *genus* in the knowledge graph. Although both entities are *cellular organism* entities, we call only the entity *strawberry* a *food* entity. Note that in FAKG, this relationship is encoded using a triplet (*Fragaria*, *hasChild*, *strawberry*). While all *food* entities have an outgoing relationship, *contains*, to a *chemical* entity, *cellular organism* entities have no outgoing *contains* relationship but only a *hasChild* relationship to another *cellular organism* or *food* entity.

A.3.3.1.2. food part

Like the *cellular organism (food)* entities, each *food part* entity has one unique NCBI ID identical to the corresponding *food* entity (e.g., entities *carrot root* and *carrot* have the same NCBI ID of 4039). In addition, each *food part* entity also has a unique part ID (e.g., *carrot root* has a part ID of *p49*, and *carrot* has a part ID of *p0*). Note that we strictly use the terminology *food part* and not *cellular organism part*, as we derive all *food part* entities from *food* entities, not other *cellular organism* entities.

A.3.3.1.3. chemical

In addition to the unique FoodAtlas ID, each *chemical* entity has either one unique PubChem CID (CID) or MeSH ID. To be specific, a chemical has a unique MeSH ID if we could not find the PubChem ID. We used CID to merge other *chemical* entities from external databases and resolve the synonyms. We prioritized PubChem ID over the MeSH ID as the chemical indexer in FAKG because PubChem is the most comprehensive, used by most external databases, and prevents the dilution of chemicals when merging. The dilution problem happens when multiple identifiers are used to merge the entities. For example, according to PubChem, eight different PubChem entries for *fumaric acid* (CIDs: 6076814, 101823788, 444972, 6364607, 6433510, 6440849, 723, 9793847) all point to a single MeSH entry (MeSH ID: C032005) as a synonym. To make matters worse, each CID can point to more than one MeSH entry or CAS entry. For example, a CID 6076814 has two MeSH entries (MeSH IDs: C032005, C030272) and three CAS entries (CAS IDs: 18016-19-8, 5873-57-4, 7704-73-6). Therefore, if we use more than one unique identifier (PubChem ID in our case) for merging entities and resolving synonyms, we can end up with a chemical entry with multiple chemicals merged (diluted) into one.

A.3.3.2. Relation types

The current version of FoodAtlas KG supports four relation types: *contains*, *isA*, *hasChild*, and *hasPart* (**Supplementary Table 17**). In addition, all relations in FAKG are assigned a unique FoodAtlas ID, a sequentially assigned numerical ID prefixed with the letter *r*. In this section, we describe each relation type in more detail.

A.3.3.2.1. contains

The *contains* relation type has two possible triplet types (*food*, *contains*, *chemical*) and (*food part*, *contains*, *chemical*), as shown in **Figure 5.2a**. These *contains* triplets are either from the FoodAtlas framework or are from three external databases Frida³²⁴, Phenol-Explorer^{37,224,225}, and FDC²¹⁶.

A.3.3.2.2. isA

The *isA* relation type has one possible triplet type (*chemical*, *isA*, *chemical*) (**Figure 5.2a**), which encodes the ontological relationship of the chemical entities. This ontological relationship is taken directly from the MeSH Tree. Note that the head entity is a child node of the tail entity in the MeSH Tree (e.g., (*Catechin*, *isA*, *Flavonoids*)).

A.3.3.2.3. hasChild

The *hasChild* relation type has one possible triplet type (*cellular organism*, *hasChild*, *cellular organism*) (**Figure 5.2a**), which encodes the taxonomical relationship of the *cellular organism* entities and is taken directly from the NCBI Taxonomy. In contrast to the *isA* entity type, the tail entity of the *hasChild* triplet is the child node of the head entity (e.g., (*Fragaria*, *hasChild*, *strawberry*)).

A.3.3.2.4. hasPart

The *hasPart* relation type has one possible triplet type (*cellular organism, hasPart, food part*) (**Figure 5.2a**). In the current version of FAKG, the only source of such triplets is the FoodAtlas active learning pipeline. Note that the entailment model only predicts (*food, contains, chemical*) or (*food part, contains, chemical*) triplet types, and if a triplet (*food part, contains, chemical*) is either annotated or predicted as positive by the FoodAtlas pipeline, we automatically extract (*food, hasPart, food part*) triplet from it and mark it as a positive.

A.3.3.3. Knowledge injection

We first inject *contains* and *hasPart* triplets into the FoodAtlas KG, followed by *isA* triplets and *hasChild* triplets from the MeSH Tree and NCBI Taxonomy, respectively. This section describes the exact procedures for this knowledge injection process.

A.3.3.3.1. FoodAtlas active learning pipeline

Three triplet types, (*food, contains, chemical*), (*food part, contains, chemical*), and (*food, hasPart, food part*), are injected into FAKG from the FoodAtlas active learning (AL) pipeline. The *food* and *chemical* entities are originally tagged with NCBI IDs and MeSH IDs, respectively, by the LitSense API. To inject these triplets into the FoodAtlas KG, a preprocessing step of looking up the corresponding PubChem CIDs (CIDs) for the *chemical* entities is necessary, as CID is used to merge the entities (**Appendix A.3.3.1.3**). For example, a *chemical* entity of *oleic acid* from the triplet (*food, contains, oleic acid*) has a unique MeSH ID of D019301. We search this unique MeSH ID in the PubChem database and find all PubChem compound entries that list the MeSH ID as a synonym. Then, for *oleic acid*, three CIDs (445639, 5460221, 965) all list the MeSH ID of D019301

as a synonym. Therefore, the original triplet (*food, contains, oleic acid*) is expanded into three triplets, each with a unique CID, before being injected into the FoodAtlas KG.

A.3.3.3.2. External databases

We inject the (*food, contains, chemical*) triplets from three external DBs, Frida, Phenol-Explorer, and FDC, into FAKG. We describe the exact steps for processing data from these external DBs in detail in **Appendix A.3.3.4**. The *food* entities from these external DBs are already tagged with NCBI taxonomy, whereas the *chemical* entities are tagged with either CID or CAS ID but not MeSH ID. If a *chemical* entity has a CID, we use it to search the PubChem database for synonym MeSH IDs. We then associate the found MeSH IDs to the CID, which are further used to merge the entities as described in **Appendix A.3.3.1.3**. However, if a *chemical* entity does not have a CID but only has a CAS ID, we look up this CAS ID on PubChem and, in turn, retrieve the MeSH IDs. Like MeSH, a single CAS ID can be a synonym for multiple PubChem entries. In such cases, we perform the same process of expanding the original triplet to multiple triplets, each with a unique CID. Note that chemicals added from external DBs may not always have a MeSH ID since some PubChem entries do not have a related MeSH ID.

A.3.3.3.3. MeSH

Once all *chemical* entities are added to FAKG, we add the ontological relationships of these chemicals using the MeSH Tree. For all *chemical* entities in the KG, we find the subset of entities with a unique MeSH ID (19,645 out of 19,722) and then search the MeSH database for these MeSH IDs. There are two types of MeSH entries: *MeSH descriptor data*, which has a prefix of the letter *D* in its unique ID (e.g., D002392 for

catechin), and *MeSH supplementary concept data*, which has a prefix of the letter *C* in its unique ID (e.g., C024603 for Geraniin). All descriptor data with the prefix *D* has one or more MeSH Tree numbers. For example, catechin (MeSH ID: D002392) has 4 MeSH Tree numbers, as shown in **Supplementary Figure 29a**. Each MeSH Tree is converted to a triplet by connecting the tree's nodes with the *isA* relation type. For example, catechin with MeSH Tree number D03.383.663.283.240.190 in **Supplementary Figure 29a** will be expanded into five triplets (*Catechin, isA, Chromans*), (*Chromans, isA, Benzopyrans*), (*Benzopyrans, isA, Pyrans*), (*Pyrans, isA, "Heterocyclic Compound, 1-Ring"*), (*"Heterocyclic Compounds, 1-Ring", isA, Heterocyclic Compounds*). All supplementary concept data with the prefix *C* are mapped to one or more descriptor data with the prefix *D*. Geraniin (MeSH ID: C024603), for example, is mapped to two descriptor data Glucosides (MeSH ID: D005960) and Hydrolyzable Tannins (MeSH ID: D047348). We encode this relationship with a triplet (*Geraniin, isA, Glucosides*) and (*Geraniin, isA, "Hydrolyzable Tannins"*), respectively. The MeSH tree of the two descriptor entities Geraniin mapped to is also translated into triplets and injected into FAKG.

A.3.3.3.4. NCBI Taxonomy

For all *food* entities in the KG, their corresponding NCBI Taxonomy IDs are used to search the NCBI Taxonomy to retrieve the taxonomic lineage information. For example, the complete lineage data of strawberry (NCBI:txid3747) with rank *species* are shown in **Supplementary Figure 30**, where the cellular organism (NCBI:txid131567) is the root entity of the lineage, and the *Fragaria* (NCBI:txid3746) is the immediate parent entity of rank *genus*. This lineage is converted to triplets by connecting the entities in the lineage

with *hasChild* relationship as (*cellular organism, hasChild, Eukaryota*), (*Eukaryota, hasChild, Viridiplantae*), ..., and (*Fragaria, hasChild, Fragaria x ananassa*).

A.3.3.4. Triplets from external databases

This section will discuss how we integrate triplets extracted from Frida, FDC, and Phenol-Explorer. Note that we have considered other food chemical composition databases. However, to integrate with FAKG, the databases need to have standardized food and chemical indexers consistent with FAKG; Specifically, a food must have an NCBI ID, and a chemical must have a CID. Most databases, however, do not necessarily suffice this requirement; For example, Frida does not provide food NCBI IDs, and FDC does not provide CIDs. In the next paragraph, we will discuss how to rely on our metadata retrieval pipeline to retrieve the NCBI IDs if the database provides scientific names. However, if a database lacks CIDs, we need to contact the authors of the databases to receive chemical IDs that can be linked to CIDs internally. The databases integrated into FAKG either contain or have provided CIDs or IDs linked to CIDs to us internally. We are actively working with the authors of other databases, including FoodDB, to expand FAKG in the following versions.

To integrate triplets from FDC, Frida, and Phenol-Explorer into FAKG with a consistent standard, we have developed a semi-automatic method relying on NCBI Entrez³²⁵ to retrieve NCBI IDs, MeSH IDs, and PMIDs with scientific names, other chemical IDs, and publication titles, respectively (**Appendix A.3.3.4.1**).

Another property of FoodAtlas is its focus on raw foods: Food products (e.g., chocolate) or processed foods (e.g., roasted chicken) may contain inconsistent chemical additives,

which introduces noise to FAKG. Thus, we relied on a rule-based filterer to remove the food-chemical relations that were non-raw foods for each external source (**Appendix A.3.3.4.2**).

For the rest of this section, we will discuss the general methodology of metadata retrieval and relation filtering pipelines in the first two subsections. Then, we will summarize the specific treatment and the result for each external database.

A.3.3.4.1. Metadata Retrieval

Retrieving NCBI IDs. If the scientific name of a food is available, we may use NCBI Entrez to query the NCBI ID with the following steps:

- Make the scientific name lowercase.
- Report for manual validation if the input contains the “x” or “u'\xd7” term.
 - Note: Crossbreeds are hard to parse but rare, so they are dealt with manually.
- Drop the input with less than two terms.
 - Note: It cannot be species, subspecies, or varieties.
- Drop the input with “.” in the first two terms.
 - Note: Scientific names for species, subspecies, or varieties have no abbreviations in the first two terms.
- Report for manual validation if the input contains “convar” or “convar.”.
 - Note: Convarieties are hard to parse but rare, so they are dealt with manually.
- Report for manual validation if the input contains more than two terms and does not contain “.”.
 - Note: The majority of the scientific names have lengths of 2. If more than 2, the scientific names either contain the abbreviated authority, e.g., “L.”, or variety terms, such as “subsp.” and “var.”. NCBI Entrez does strict string matching for NCBI ID queries, so this step ensures that data sources are not omitting “.” by accident.
- If the scientific name has a length of 2, query the NCBI ID.
- If the scientific name has a length longer than 2, format the variety terms, remove the authority terms, and query the NCBI ID.
 - Note: Some external databases use a variety of terms that are not accepted by NCBI Entrez, e.g., “ssp.” instead of “subsp.”, which needs to be formatted accordingly. The authority terms are optional for the NCBI Entrez API. However, if authority terms are contained, they must be

correct for the API to return the NCBITaxon ID. Thus, it is much easier to remove them.

- Report for manual validation if NCBI Entrez does not return an ID. Fix if the error is due to minor typos.

This procedure ensured that the automated part of our method was exact (i.e., if NCBI Entrez returned, it was correct) and minimized the need for manual validation.

Retrieving PMID. If the publication title of a reference is available, we may use NCBI Entrez to query the PMID with the following steps:

- Make the publication title lowercase.
- Remove all stop words and reformat the input into the advanced search query format defined in NCBI Entrez API³²⁵.
 - Note: Specifically, the underlined terms of “uptake and metabolism of epicatechin and its access to the brain after oral ingestion” are removed first, and then the sentence is transformed to “uptake[Title] AND metabolism[Title] AND epicatechin[Title] AND access[Title] AND brain[Title] AND after[Title] AND oral[Title] AND ingestion[Title]”. We have empirically tested and found that this approach maximizes the hit rate.
- Query with the formatted input and retrieve its PMID(s).
- Select and verify the correctness of the PMID with the following steps.
 - Use the retrieved PMID to query and retrieve the publication title from PubMed.
 - Make the retrieved publication title lowercase.
 - Remove all punctuations and whitespaces in the input publication title and the retrieved publication title.
 - Compare the input publication title and the retrieved publication title. Verification succeeds if and only if the two titles are strictly matching. If not, report for manual validation.
 - If multiple PMIDs are returned for a single input publication title, report for manual validation if none of the PMIDs pass the verification.

Similar to retrieving NCBITaxon ID, this procedure ensured that our method’s automated part was exact (i.e., if it passed verification, it was correct) while minimizing manual validation.

A.3.3.4.2. Relation Filtering

Due to the inconsistency of food naming conventions among external databases, relation filtering has been implemented specifically for each external database to remove relations associated with non-raw foods. The primary mechanism of the filtering is to detect either specific substring of food names or specific food groups if available. If the concentration value associated with a relation is zero, we do not add that relation to the KG.

A.3.3.4.3. FDC

Foods. FDC is a massive reservoir of food. However, after consulting with researchers working on FDC, since the SR Legacy foods samples mixed different species/subspecies, we decided not to use them. Therefore, we only focused on Foundation Foods, where 56 have NCBI IDs.

Chemicals. As of 02/03, the current database does not provide CIDs or CAS IDs for the chemicals in the database. We consulted the FDC researchers and received some work-in-progress ID mapping internally. With that, we retrieved CIDs for 62 chemicals.

A.3.3.4.4. Frida

Foods. Frida contains 1,249 unique foods with common and scientific names. To exclude foods that are food products or processed (i.e., with additives), we only included foods with the term *raw* in their food names, excluding most non-raw foods. In addition, we ignored some food groups (**Supplementary Table 17**) for further food cleaning. Lastly, we retrieved NCBI IDs for these foods using their scientific names, which resulted in 200 usable raw foods.

Chemicals. Frida contains 205 unique *parameters*, where a parameter can be either a micronutrient or a macronutrient. We dropped most macronutrients due to their lack of

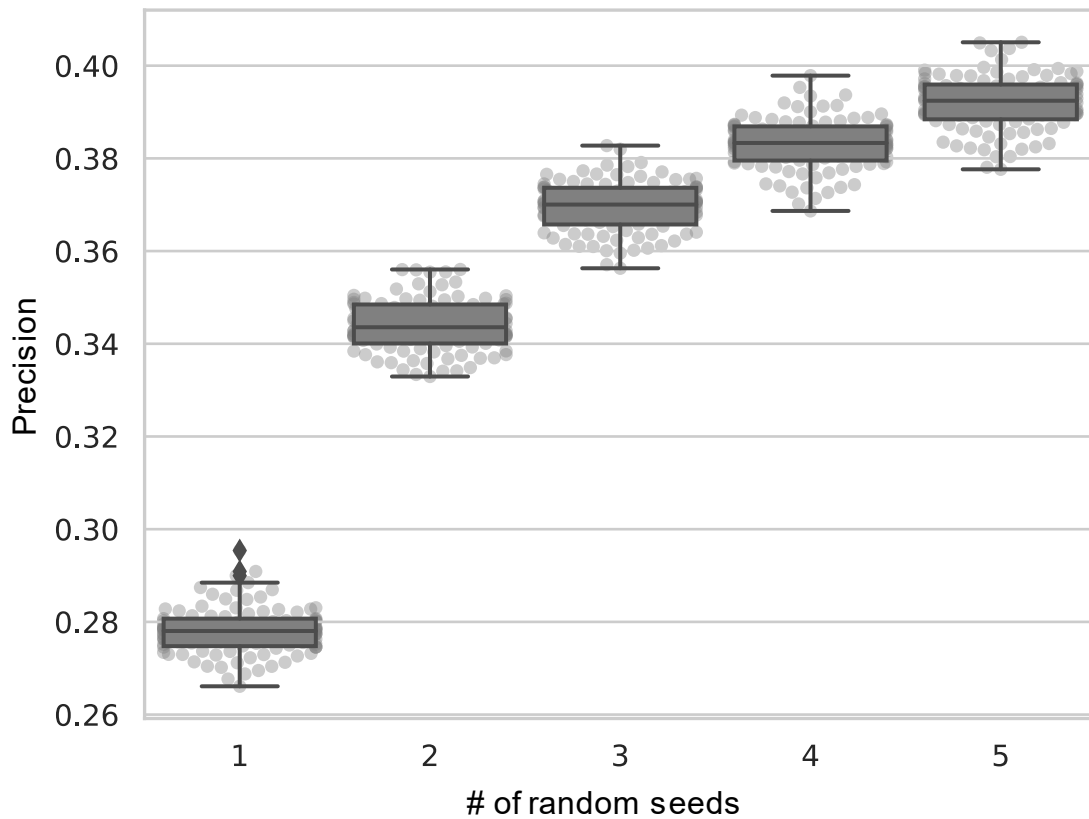
CIDs. Because the original Frida database does not contain CIDs, we contacted the correspondent of the Frida team and internally received the CIDs and CAS IDs for its parameters. Finally, we collected 133 chemicals compatible with FAKG.

A.3.3.4.5. Phenol-Explorer

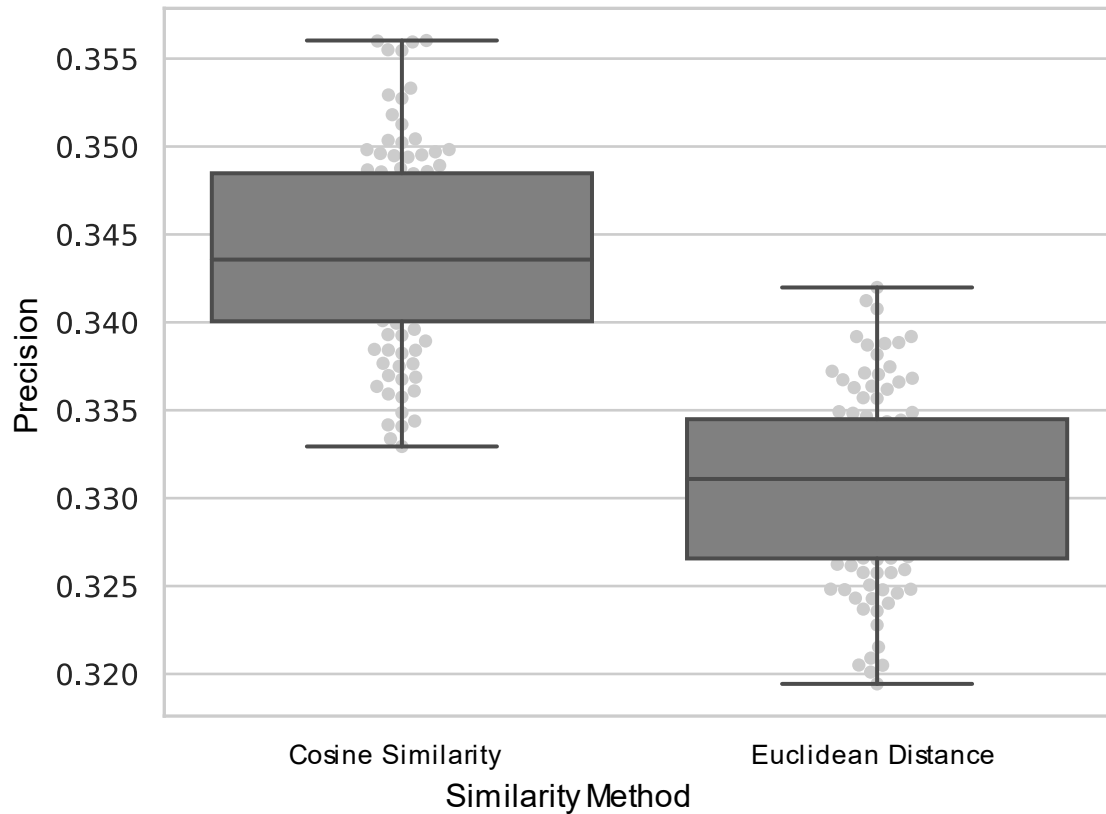
Foods. Phenol-Explorer initially contains 459 foods with common and scientific names. After removing products and processed foods (**Supplementary Table 17**), we retrieved NCBI IDs using the metadata retrieval pipeline. Consequently, we obtained 194 foods with IDs.

Chemicals. Phenol-Explorer contains 501 chemicals, each, if available, associated with CID and CAS ID. We used 155 chemicals that were at least associated with one of the IDs.

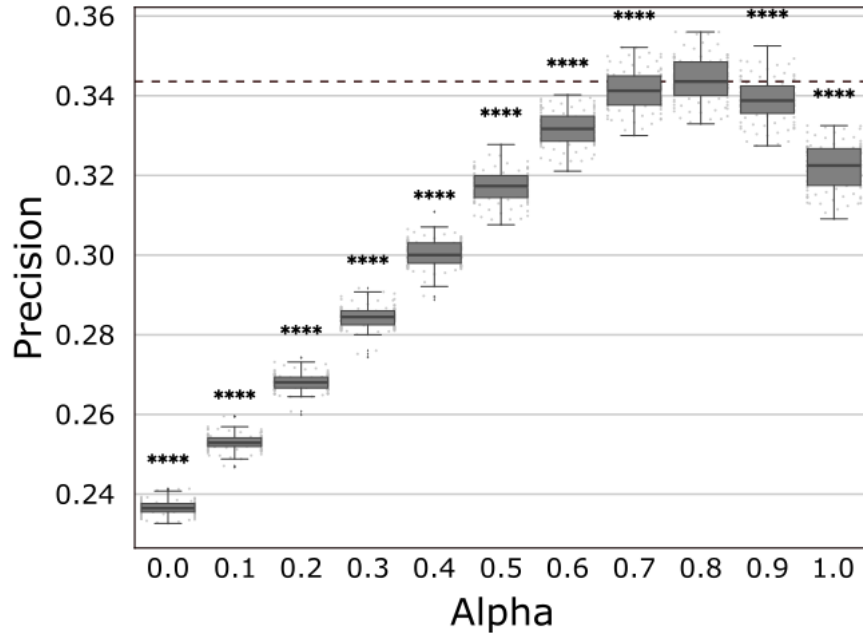
Appendix B. Supplementary figures



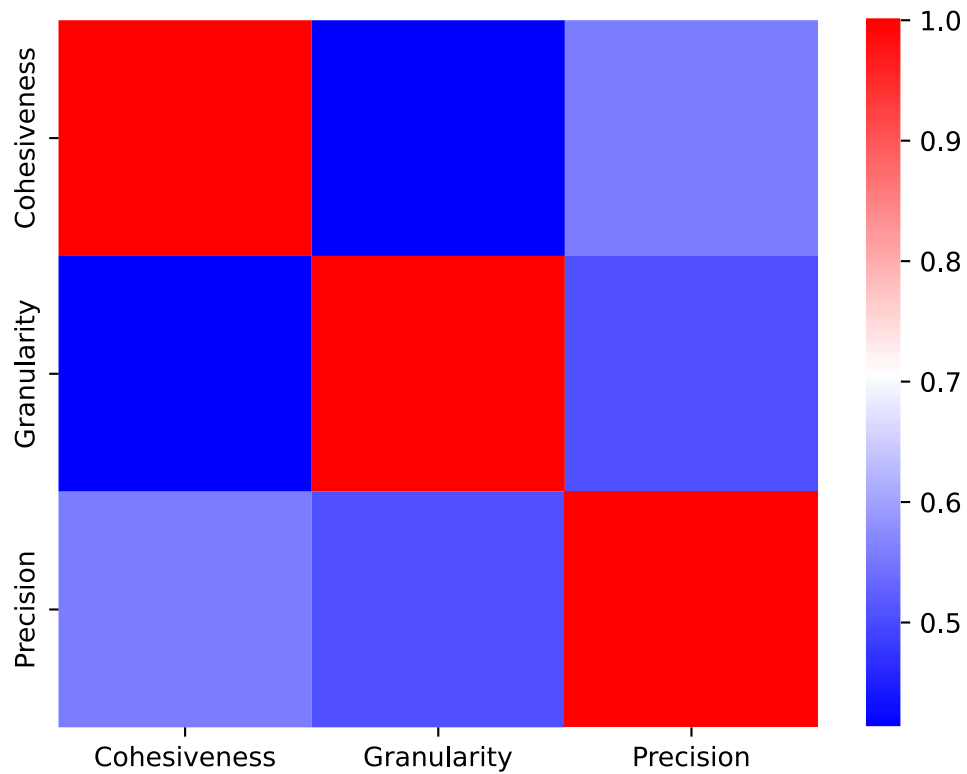
Supplementary Figure 1. Precision with respect to the number of random seeds used. The precision of FoodOn ontology mapping increases as the number of random seeds for each target class increases. In the case of FoodOn, we arbitrarily set $n_{seed} = 2$.



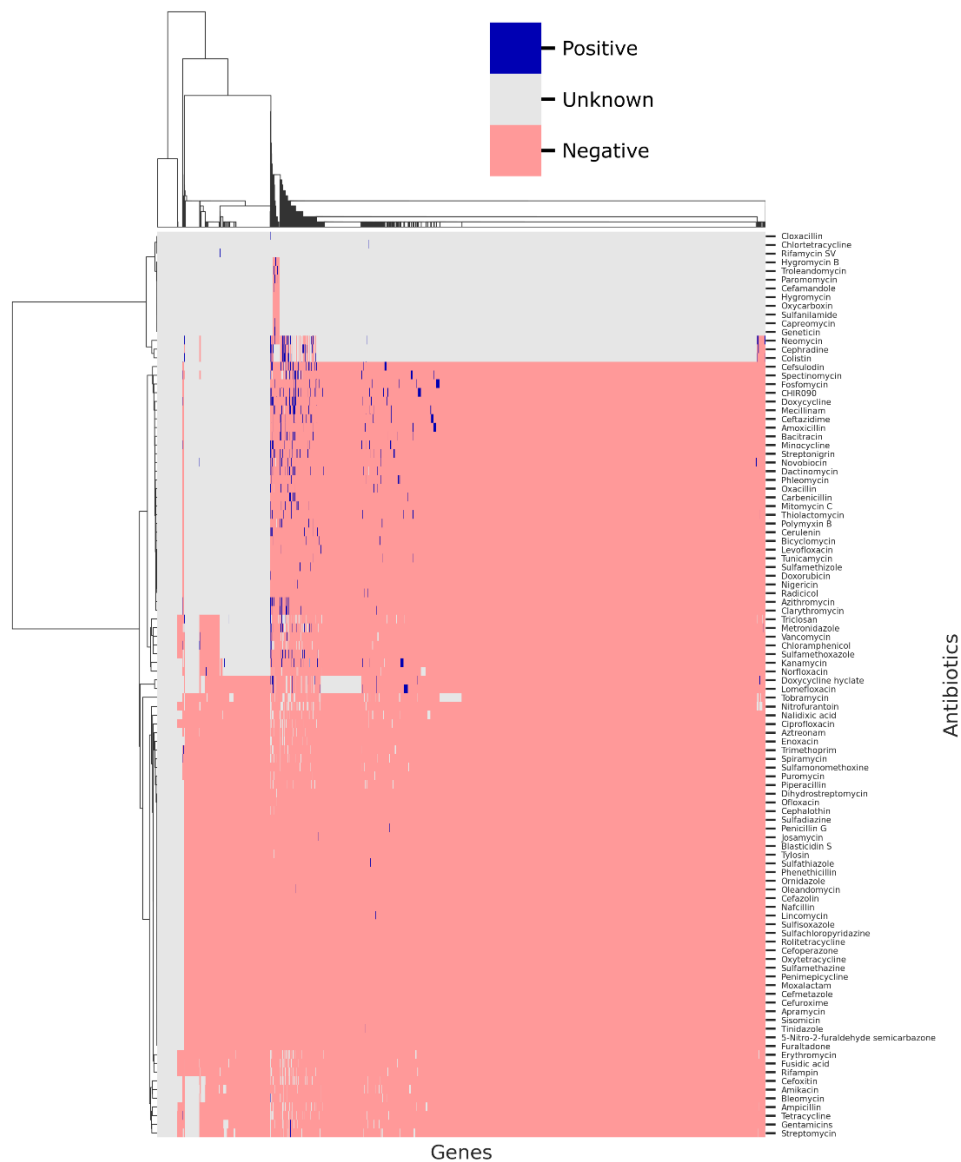
Supplementary Figure 2. Comparison of cosine similarity and Euclidean distance. For measuring similarity between the word embeddings, cosine similarity had better performance than the Euclidean distance (0.34 vs. 0.33, precision respectively; p -value = 8.1×10^{-54}).



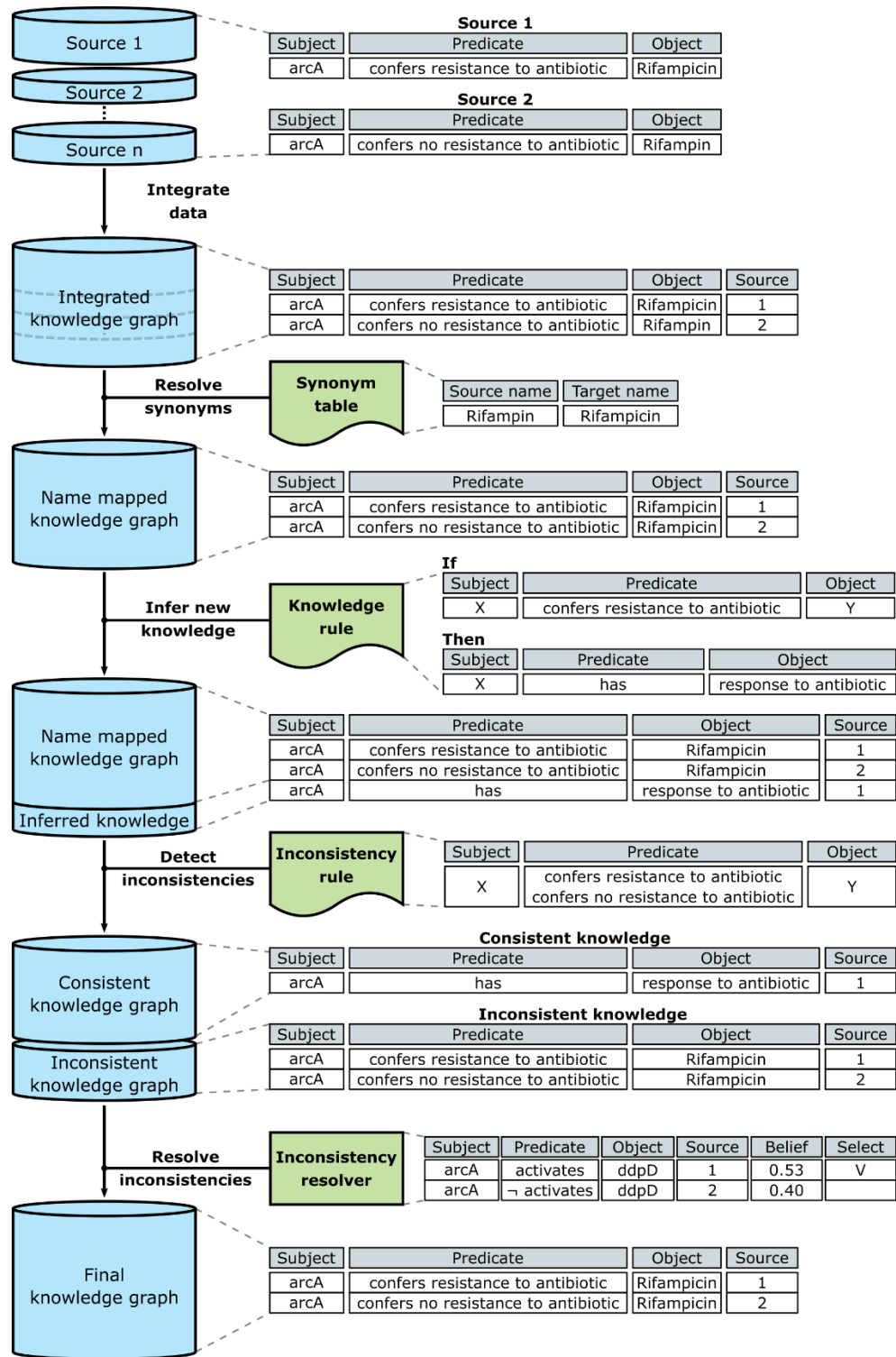
Supplementary Figure 3. Precision of ontology mapping with respect to the hyperparameter alpha. Alpha controls the balance between the similarity of the candidate entity with the target class or seed entities of the target class. The best performance is for alpha = 0.8 and the grey dots denote samples.



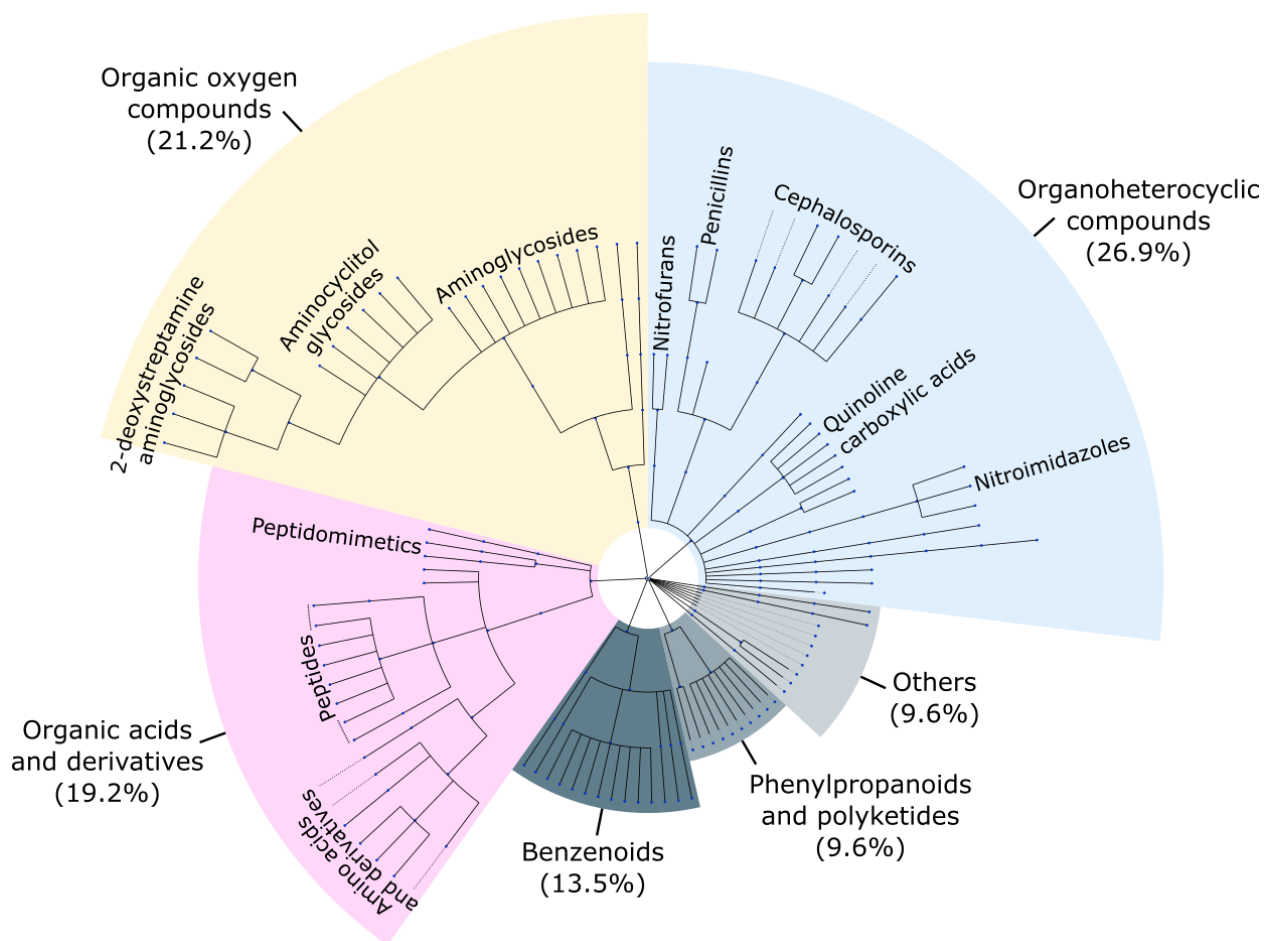
Supplementary Figure 4. Pairwise Pearson correlation between the artifacts of an ontology and the precision of ontology mapping. Both cohesiveness and granularity are positively correlated with the precision of ontology mapping (PCC of 0.56 and 0.51, respectively; p -value = 2.5×10^{-2} and 4.5×10^{-2} , respectively). Albeit not statistically significant, there also exists a positive correlation between cohesiveness and granularity (PCC of 0.41, p -value = 1.1×10^{-1}).



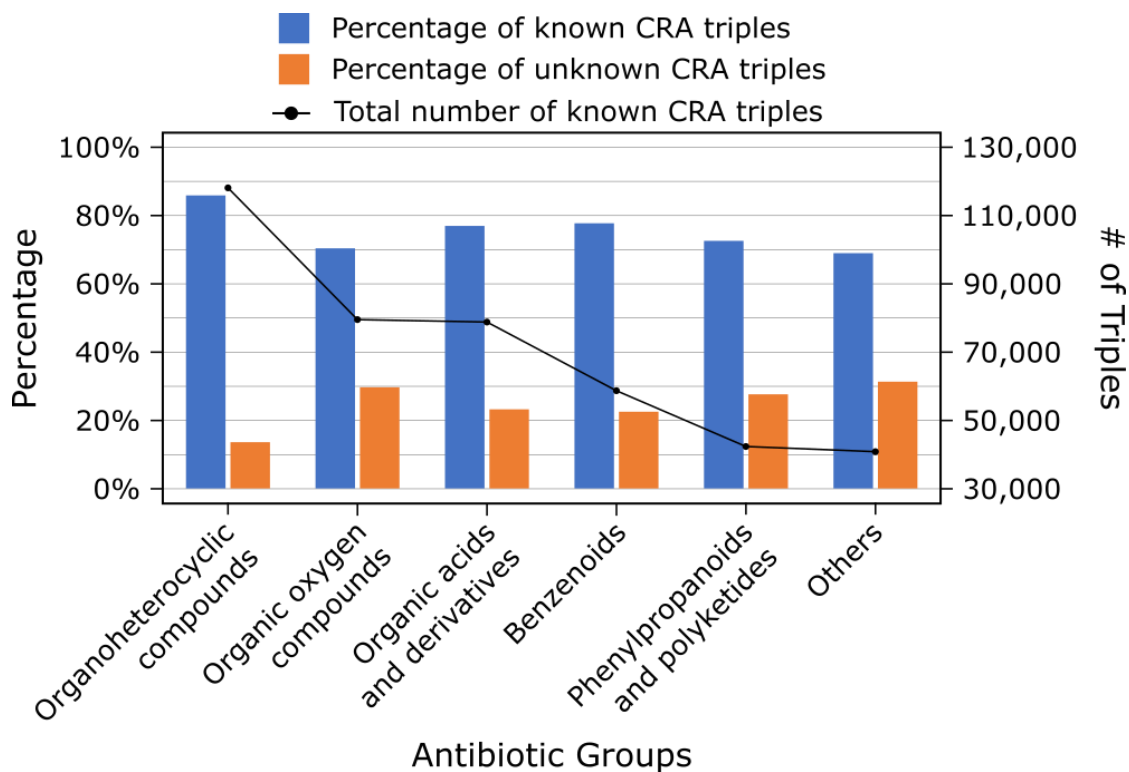
Supplementary Figure 5. Hierarchy-clustered heatmap showing the curation status of all pairwise combinations of genes and antibiotics. Positive (blue) and negative (red) cells denote there exist positive and negative CRA predicate, respectively, between the corresponding genes and antibiotics. Unknown (gray) cells denote possible candidates for either positive or negative CRA association.



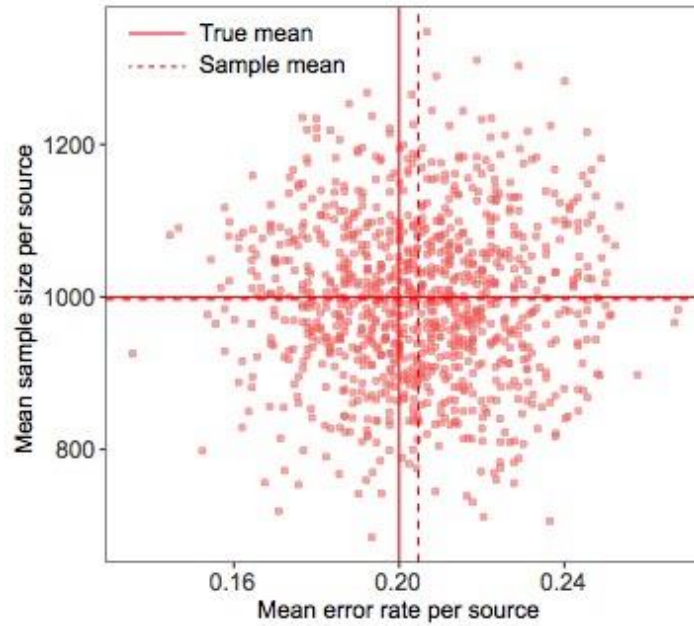
Supplementary Figure 6. Knowledge graph construction and inconsistency resolution process with examples.



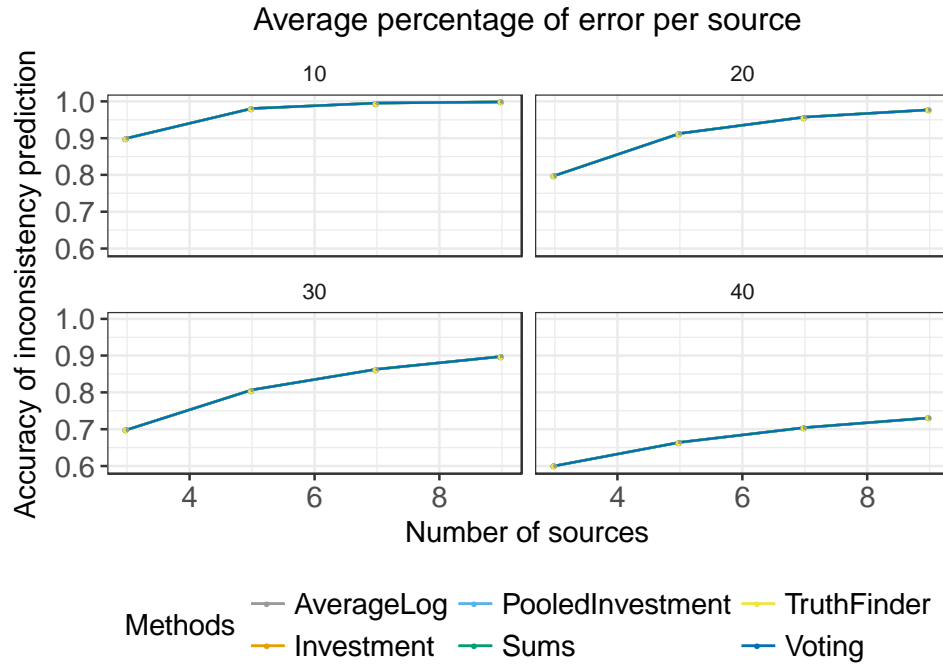
Supplementary Figure 7. Cladogram showing 6 unique taxonomic groups of antibiotics present in the knowledge graph. We classified the 104 antibiotics present in the knowledge graph into 6 taxonomic groups of antibiotics using the chemical classification ontology.



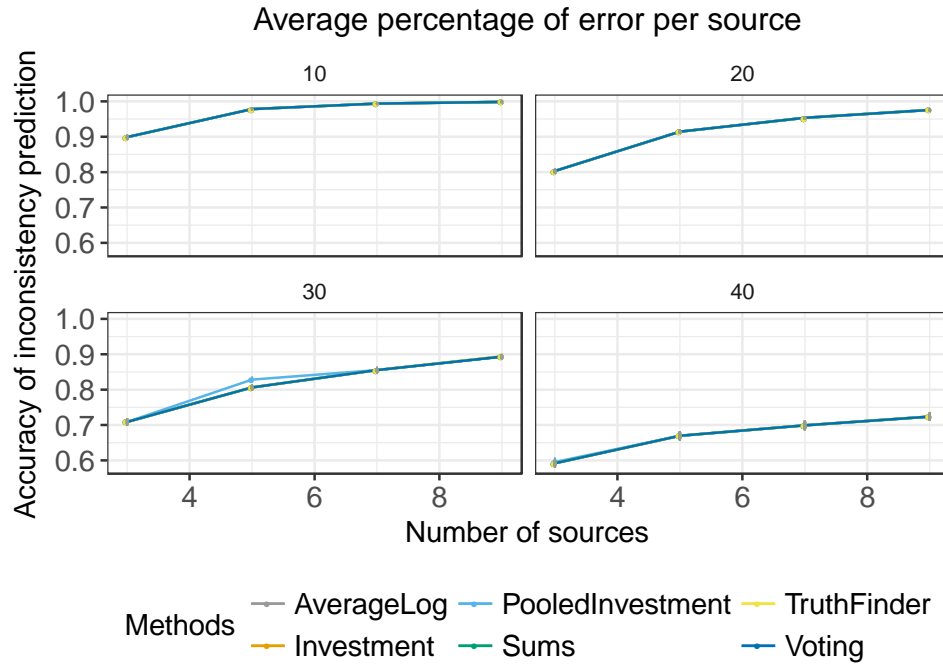
Supplementary Figure 8. The bar chart shows the percentage of known and unknown triplets with CRA predicate for each taxonomic group of antibiotics. Known triplets are used for training the hypothesis generator, whereas hypotheses will be generated for the unknown triplets. The solid line shows the actual number of CRA triplets for each group.



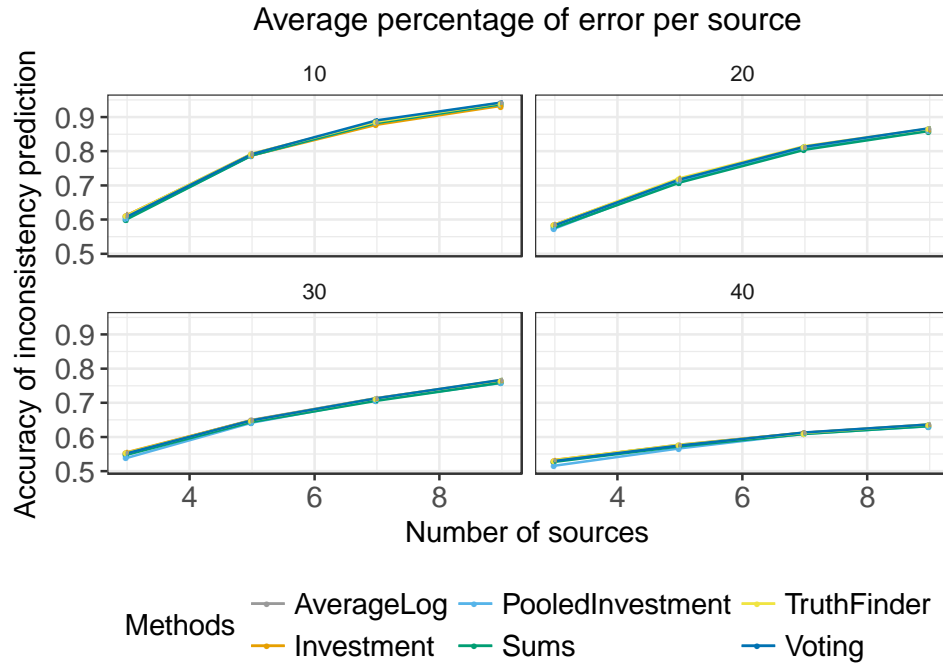
Supplementary Figure 9. Comparison of sampled mean and true mean for the two parameters (sample size per source and error rate per source) used in generating the synthetic datasets.



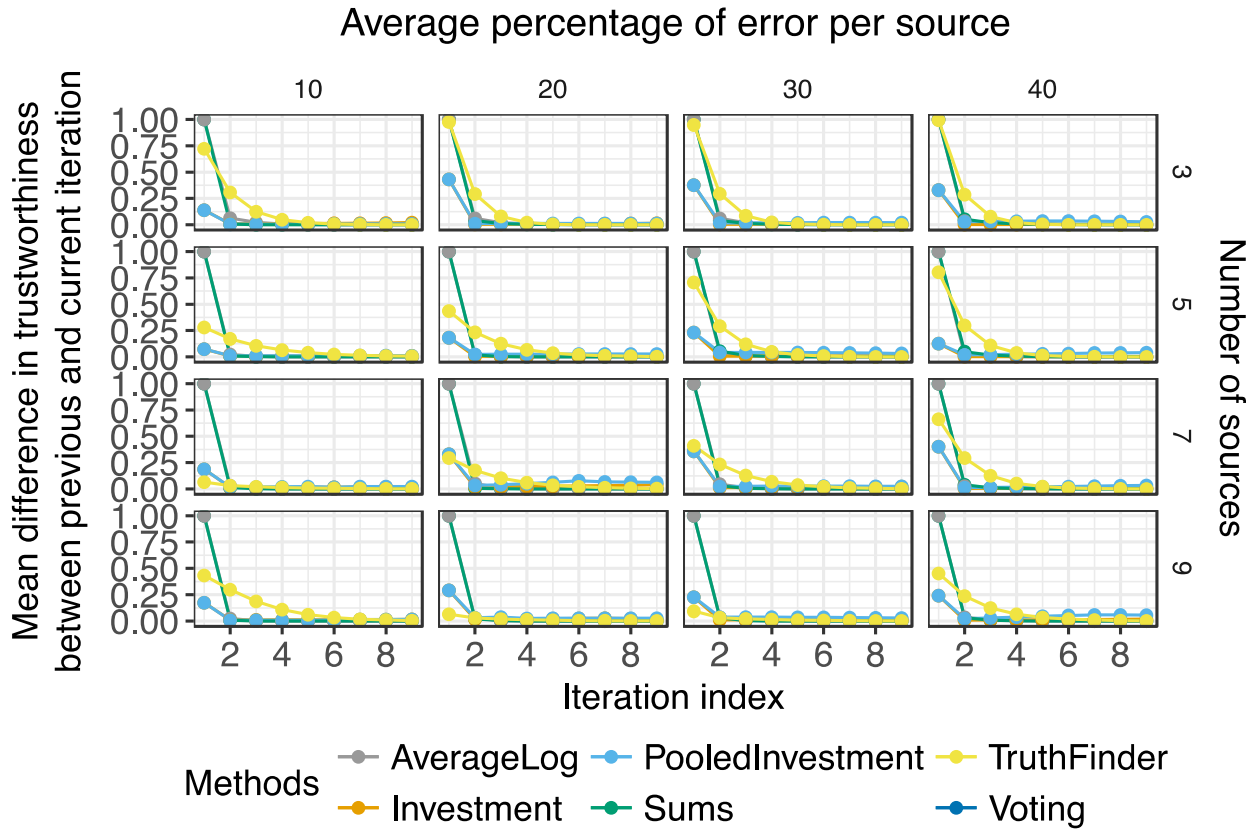
Supplementary Figure 10. Accuracy comparison of six inconsistency correction methods where the number of triplets per source and error rate per source is fixed.



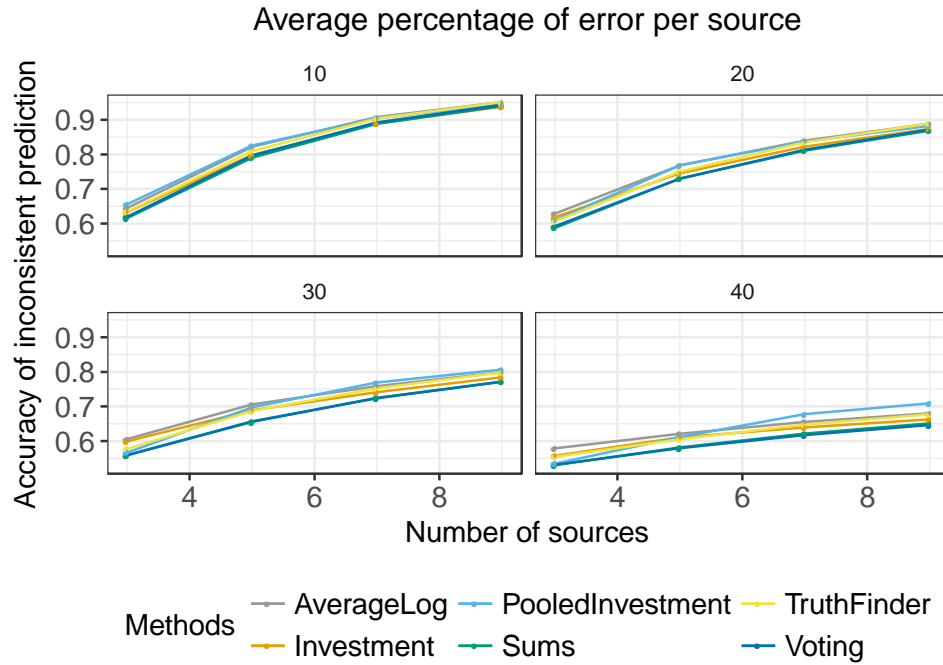
Supplementary Figure 11. Accuracy comparison of 6 inconsistency correction methods where the number of triplets per source and error rate per source is sampled.



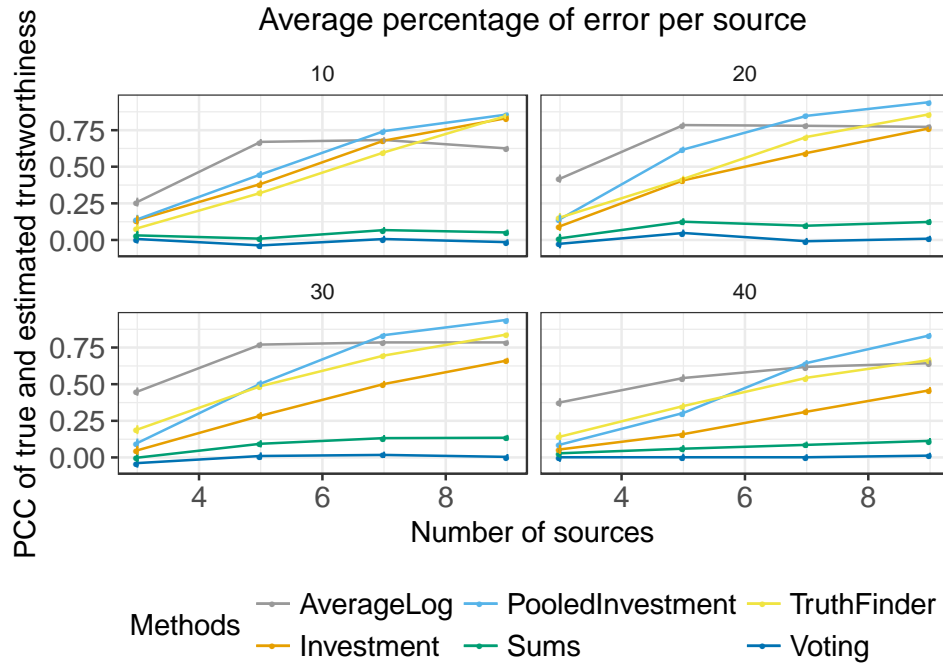
Supplementary Figure 12. Accuracy comparison of 6 inconsistency correction methods where the number of triplets per source is fixed and the error rate per source is sampled.



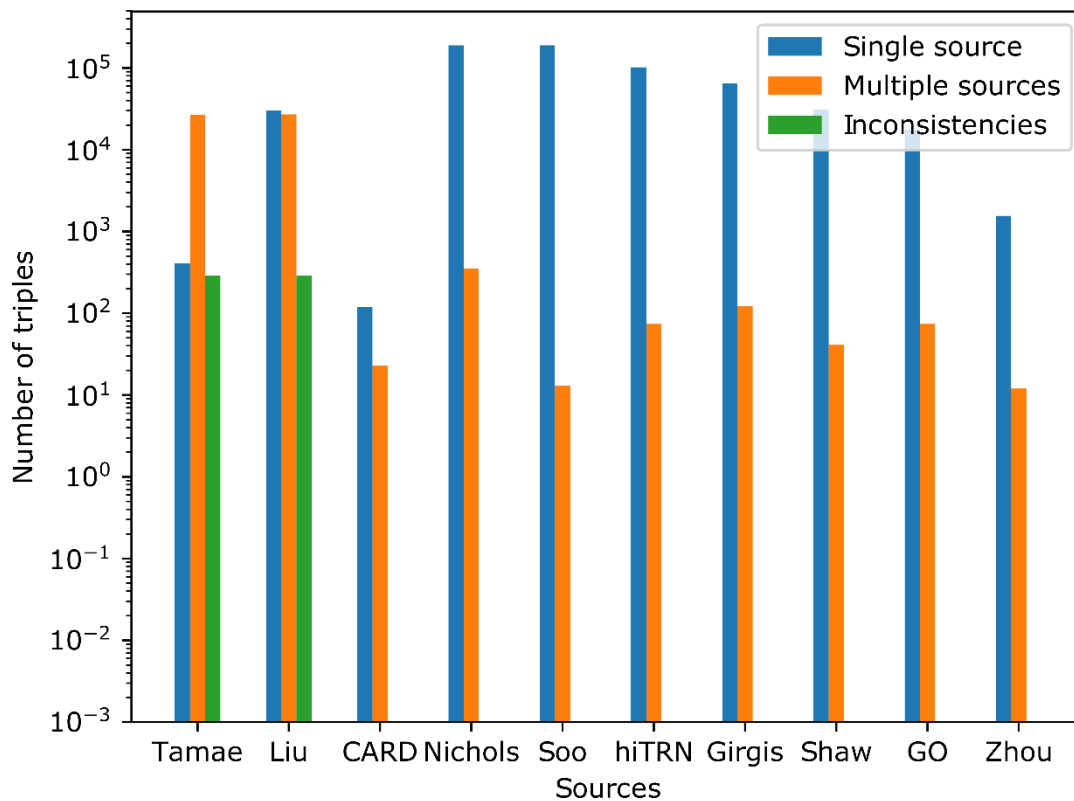
Supplementary Figure 13. The convergence of relative trustworthiness measured by inconsistency correction methods through multiple iterations.



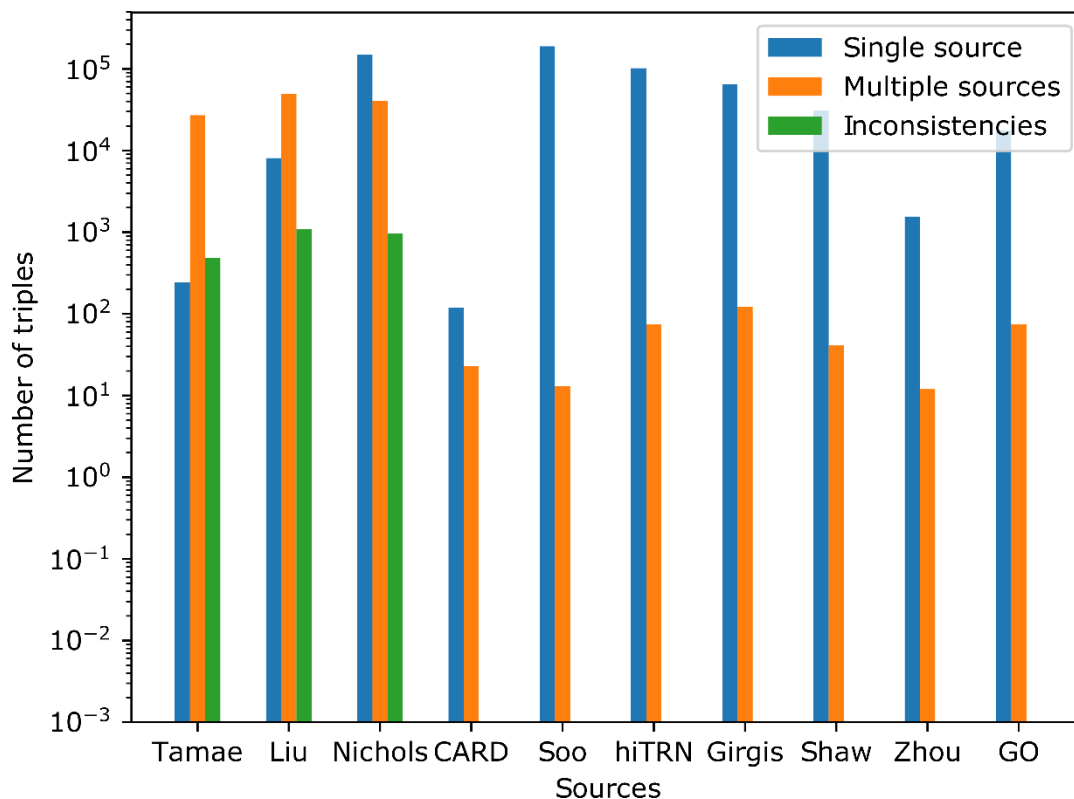
Supplementary Figure 14. Accuracy comparison of the 6 inconsistency correction methods number of triplets per source and error rate per source is sampled.



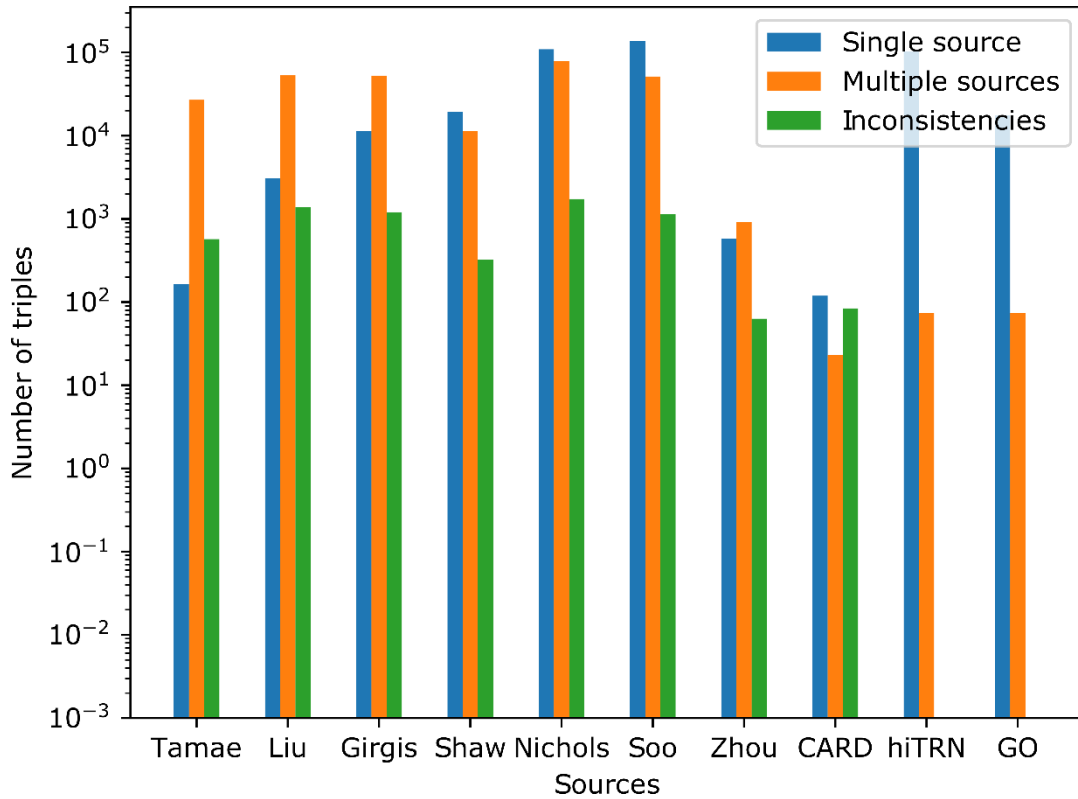
Supplementary Figure 15. The trustworthiness of sources estimated by 6 inconsistency correction methods and the ratio of true triplets per source. The number of triplets per source and error rate per source are sampled.



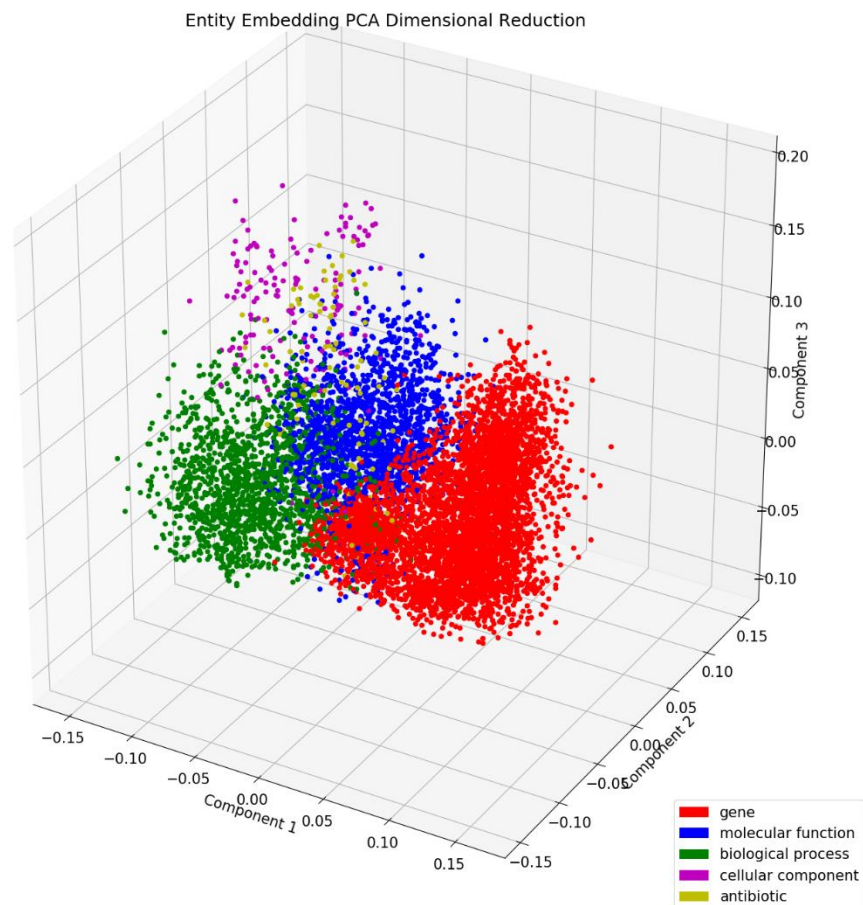
Supplementary Figure 16. The number of conflicting triplets for inconsistency resolution level 1. Distribution of triplets among different sources where 291 sets of inconsistencies (green) originate from Tamae and Liu between the two predicates: *'confers resistance to antibiotic after 18 hours'* and *'confers no resistance to antibiotic after 18 hours.'*



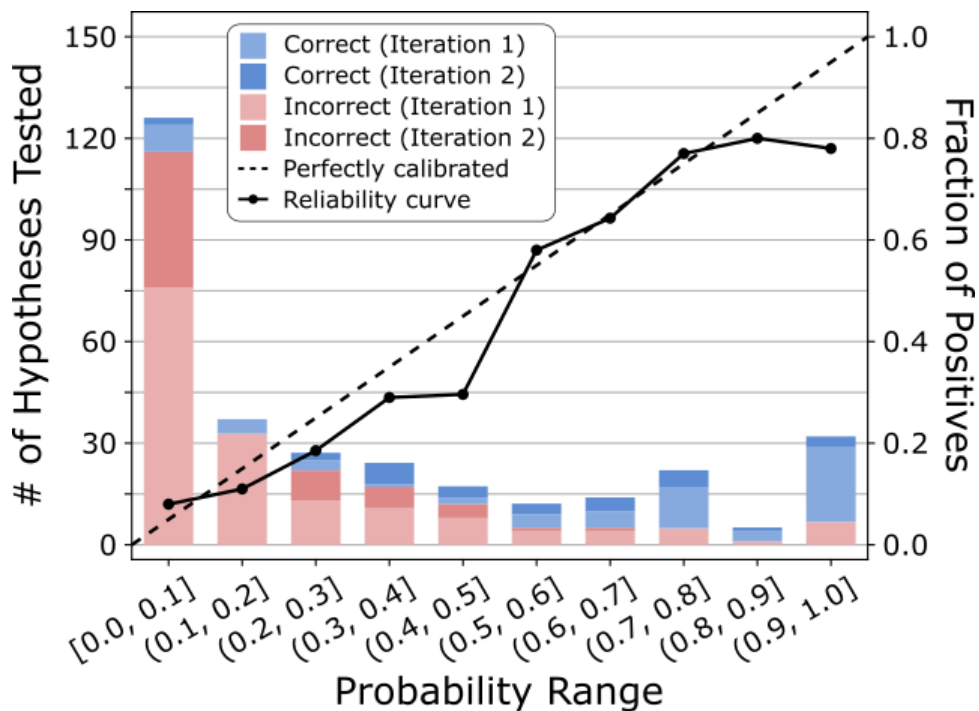
Supplementary Figure 17. The number of conflicting triplets for inconsistency resolution level 2. Distribution of triplets among different sources after alleviating the inconsistency detection criteria by considering antibiotic exposure time of 15 hours and 18 hours to be negligible. In this case, 1,096 sets of inconsistencies originate from Tamae et al., Liu et al., and Nichols et al. between the predicates: ‘confers (no) resistance to antibiotic after 15 hours’ and ‘confers (no) resistance to antibiotic after 18 hours.’



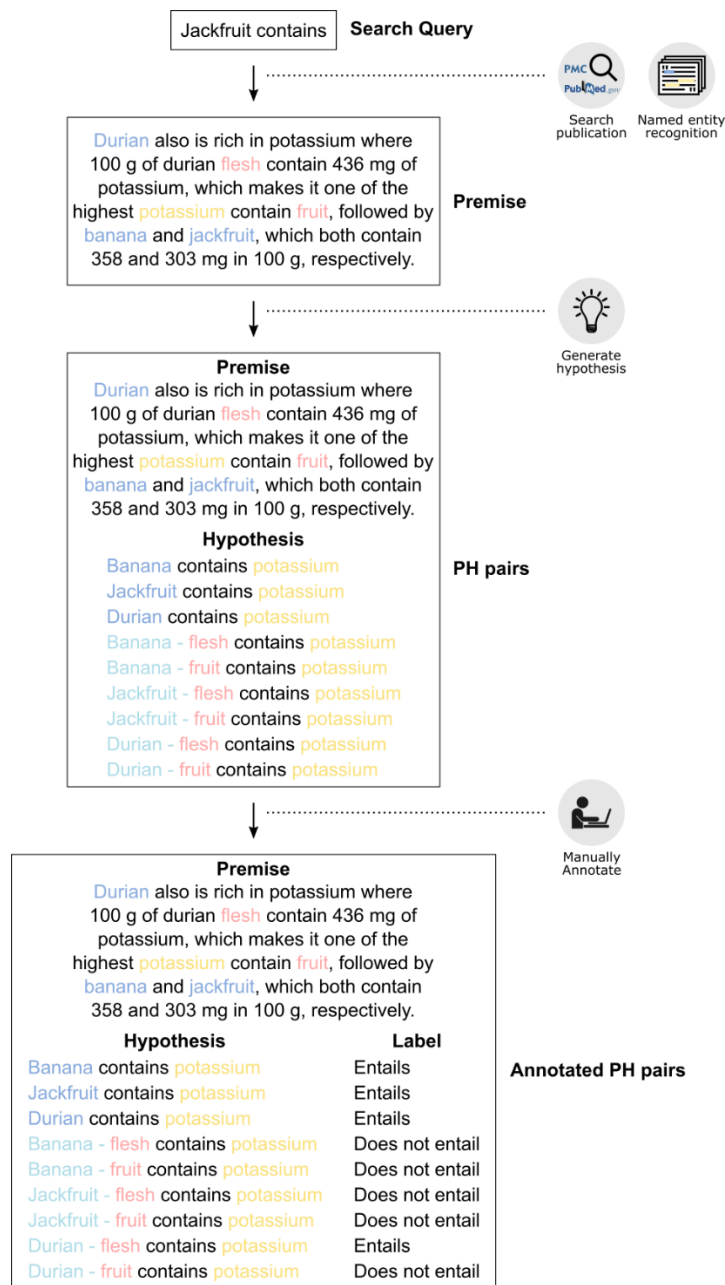
Supplementary Figure 18. The number of conflicting triplets for inconsistency resolution level 3. Distribution of triplets among different sources after expanding the inconsistency detection criteria to treat all CRA edges to be the same regardless of their source characteristics. In this case, 2,131 sets of inconsistencies originate from all sources but hiTRN and GO.



Supplementary Figure 19. Visualization of the dimensional reduction performed on the entity embeddings using principal component analysis. The initialization of these embeddings was random at the start of training; however, the noticeable clusters formed after training show that a semantic space was produced, with each entity type predominantly in their cluster. The interesting aspect of these clusters is that the MLP was never provided the entity types during training. It simply learned the entity types based on their relationship with other entities in the knowledge graph.



Supplementary Figure 20. Analysis of the validated hypotheses from the two iterations of hypotheses generation with 10 bins. Binning the probability of the 316 hypotheses from both iterations (226 and 90 from the first and second iterations, respectively) into 10 bins shows a high correlation between the probability assignment by the hypothesis generator and forward experimental validation ($R^2=0.92$).



Supplementary Figure 21. Example of PH pairs generation and the annotation process of the FoodAtlas framework. For a single premise shown above, 9 PH pairs are generated, among which four are annotated as being positives.

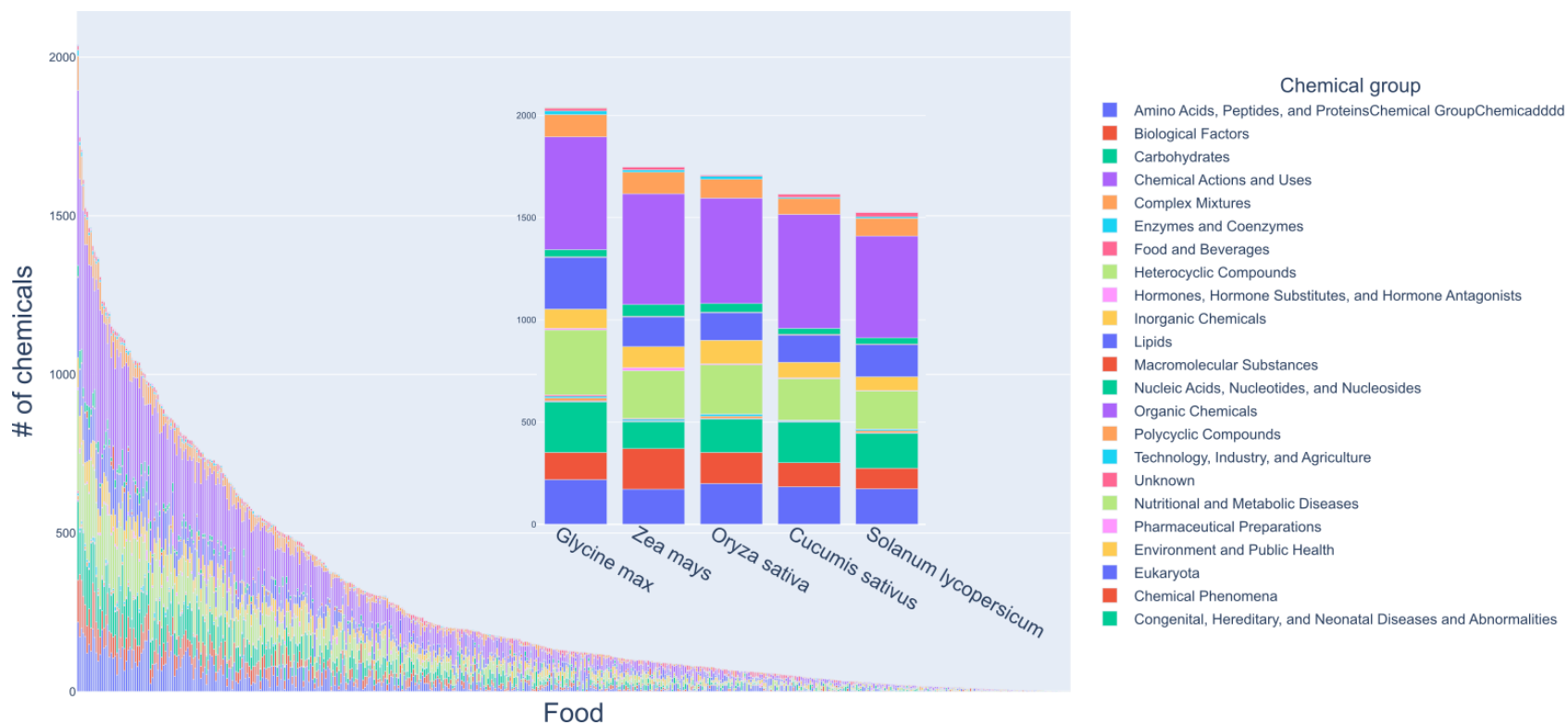
a		b		c	
PH pair	Probability	PH pair	Probability	PH pair	Probability
#1	0.99	#1	0.99	#1	0.99
#2	0.95	#2	0.95	#2	0.95
#3	0.86	#3	0.86	#3	0.86
#4	0.83	#4	0.83	#4	0.83
#5	0.72	#5	0.72	#5	0.72
#6	0.69	#6	0.69	#6	0.69
#7	0.66	#7	0.66	#7	0.66
#8	0.50	#8	0.50	#8	0.50
#9	0.42	#9	0.42	#9	0.42
#10	0.39	#10	0.39	#10	0.39
#11	0.25	#11	0.25	#11	0.25
#12	0.22	#12	0.22	#12	0.22
#13	0.15	#13	0.15	#13	0.15
#14	0.09	#14	0.09	#14	0.09
#15	0.02	#15	0.02	#15	0.02

d			e	
PH pair	Probability	Uncertainty	PH pair	Probability
#1	0.99	0.01	#1	0.99
#2	0.95	0.05	#2	0.95
#3	0.86	0.14	#3	0.86
#4	0.83	0.17	#4	0.83
#5	0.72	0.28	#5	0.72
#6	0.69	0.31	#6	0.69
#7	0.66	0.34	#7	0.66
#8	0.50	0.50	#8	0.50
#9	0.42	0.42	#9	0.42
#10	0.39	0.39	#10	0.39
#11	0.25	0.25	#11	0.25
#12	0.22	0.22	#12	0.22
#13	0.15	0.15	#13	0.15
#14	0.09	0.09	#14	0.09
#15	0.02	0.02	#15	0.02

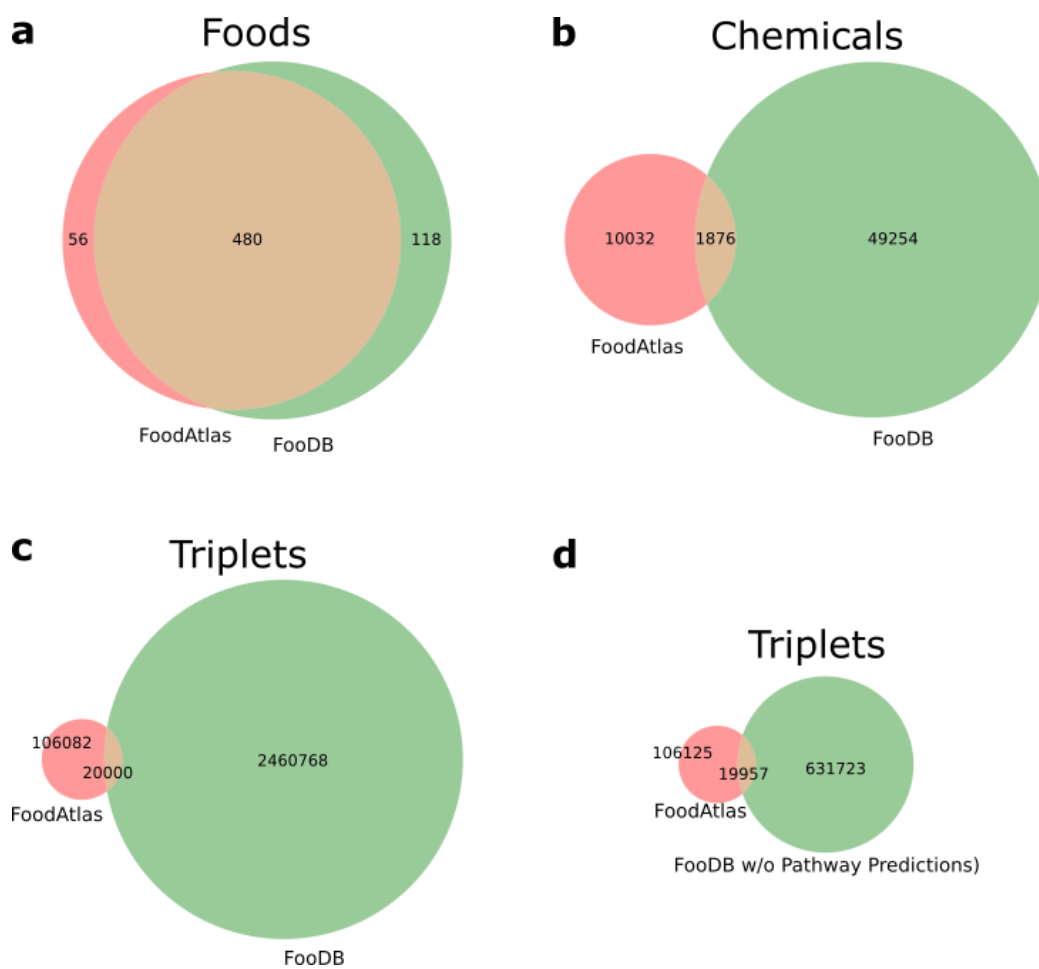
Supplementary Figure 22. Visualization of the active learning sampling strategies where three PH pairs (marked in bold) are to be chosen for each strategy. **a** A sample of 15 PH pairs for visualization purposes was ordered from high probability to low. **b** The *stratified* sampling strategy first bins the PH pairs into three equally sized bins, and one sample is drawn from each bin. Note that in actual implementation, we bin the PH pairs into ten equal-sized bins. **c** The *maximum likelihood* strategy selects the top three PH pairs with the highest probability. **d** The *maximum entropy* strategy selects the PH pairs with the highest uncertainty scores. **e** The *random* strategy selects PH pairs randomly.



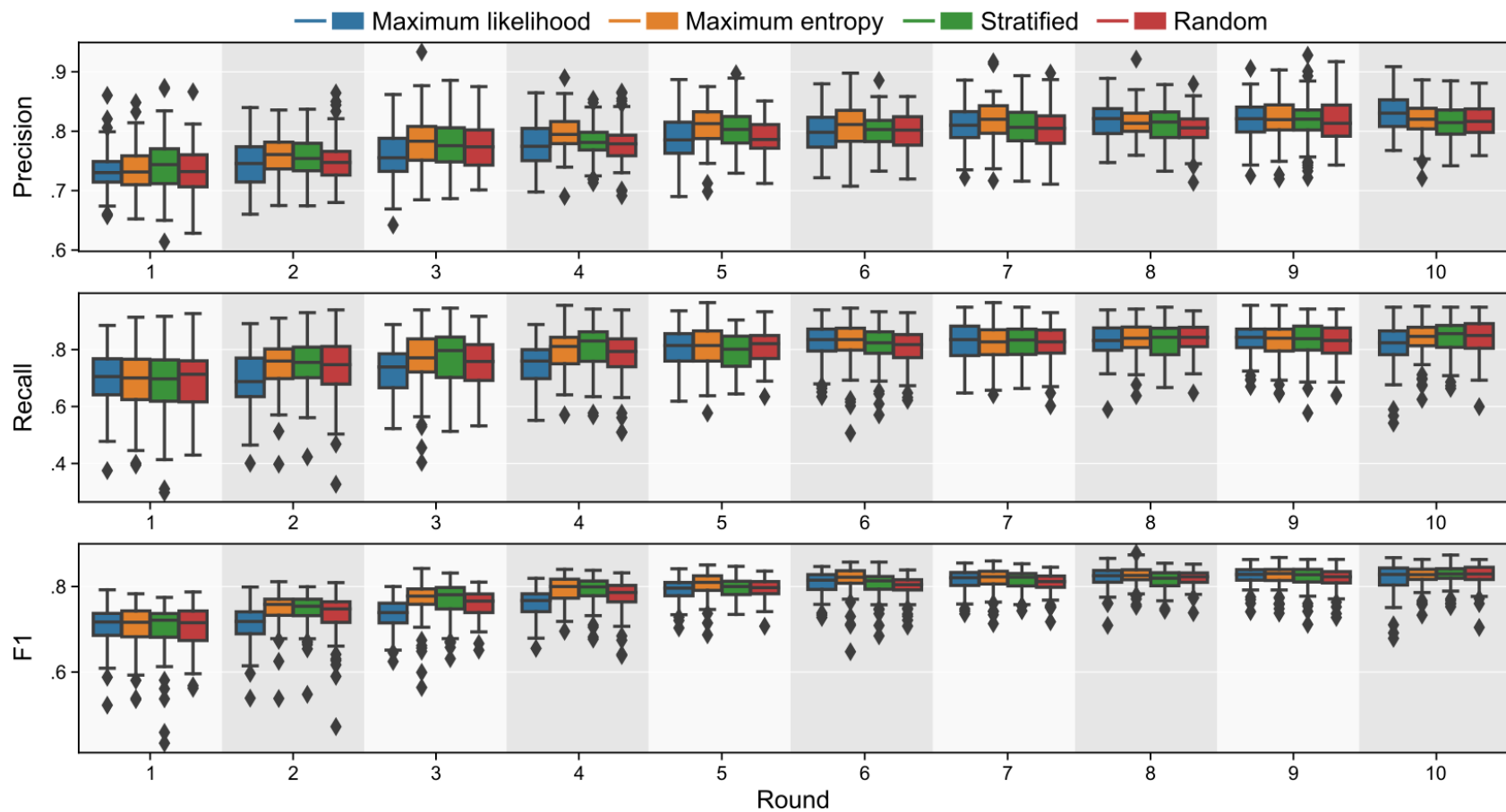
Supplementary Figure 23. Sample illustration of a Bing Chat search. In this example, Bing Chat returned two references to its claims. We made decisions by checking the validity of the referenced sources. Note that the conversation type of Bing Chat was set to *Balanced*.



Supplementary Figure 24. Foods in the FoodAtlas knowledge graph and their number of connections to the chemicals by the *contains* relationship. The chemicals are color-coded by their chemical group. The barplot inside the main bar plot shows the top 5 foods with the most *contains* relationship in the knowledge graph.

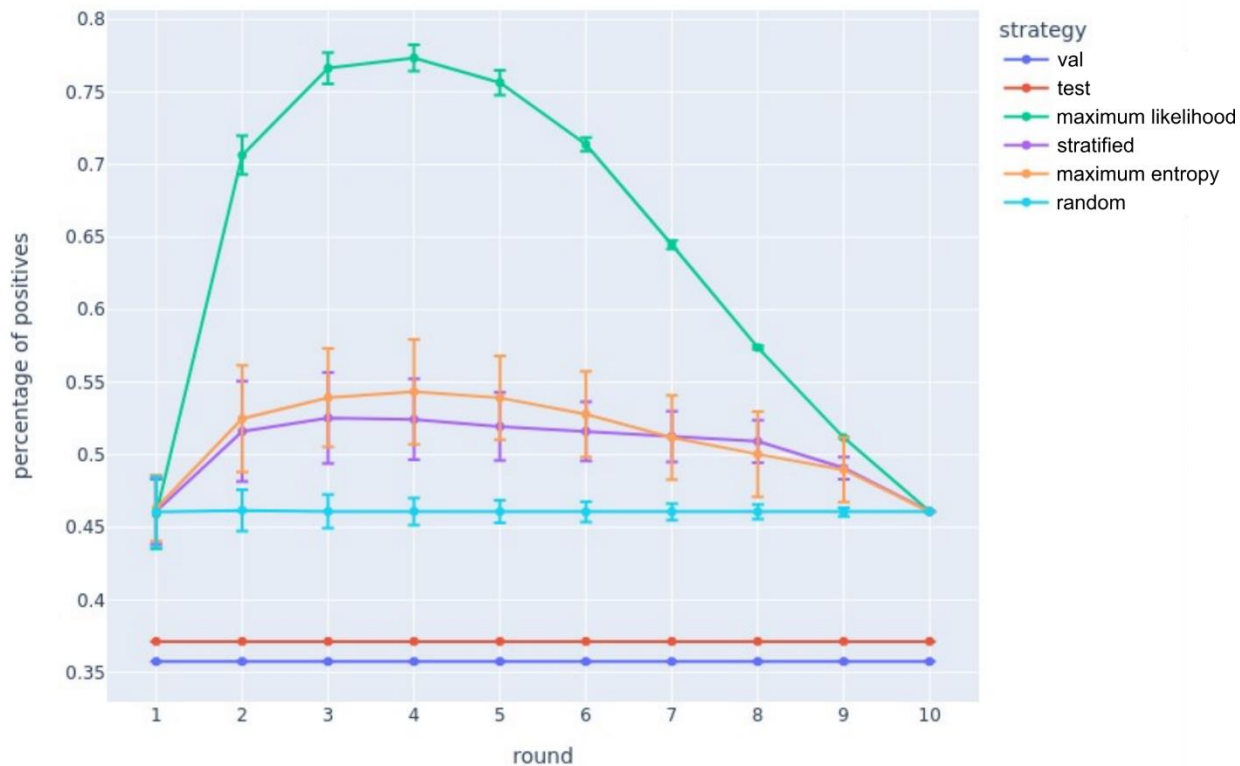


Supplementary Figure 25. Comparison of the number of entities in FAKG with that in FooDB. **a, b** Foods and chemicals were indexed based on NCBI Taxonomy IDs and PubChem CIDs, respectively, and the Venn diagrams show the coverage of unique IDs of the two databases. **c, d** Triplets were indexed based on pairs of NCBI Taxonomy IDs and PubChem CIDs, and the diagrams show the coverage of the unique paired IDs of the two databases. Note that a food with food part has the same NCBI Taxonomy ID as the corresponding food, and thus it was considered as one entity in the diagrams.

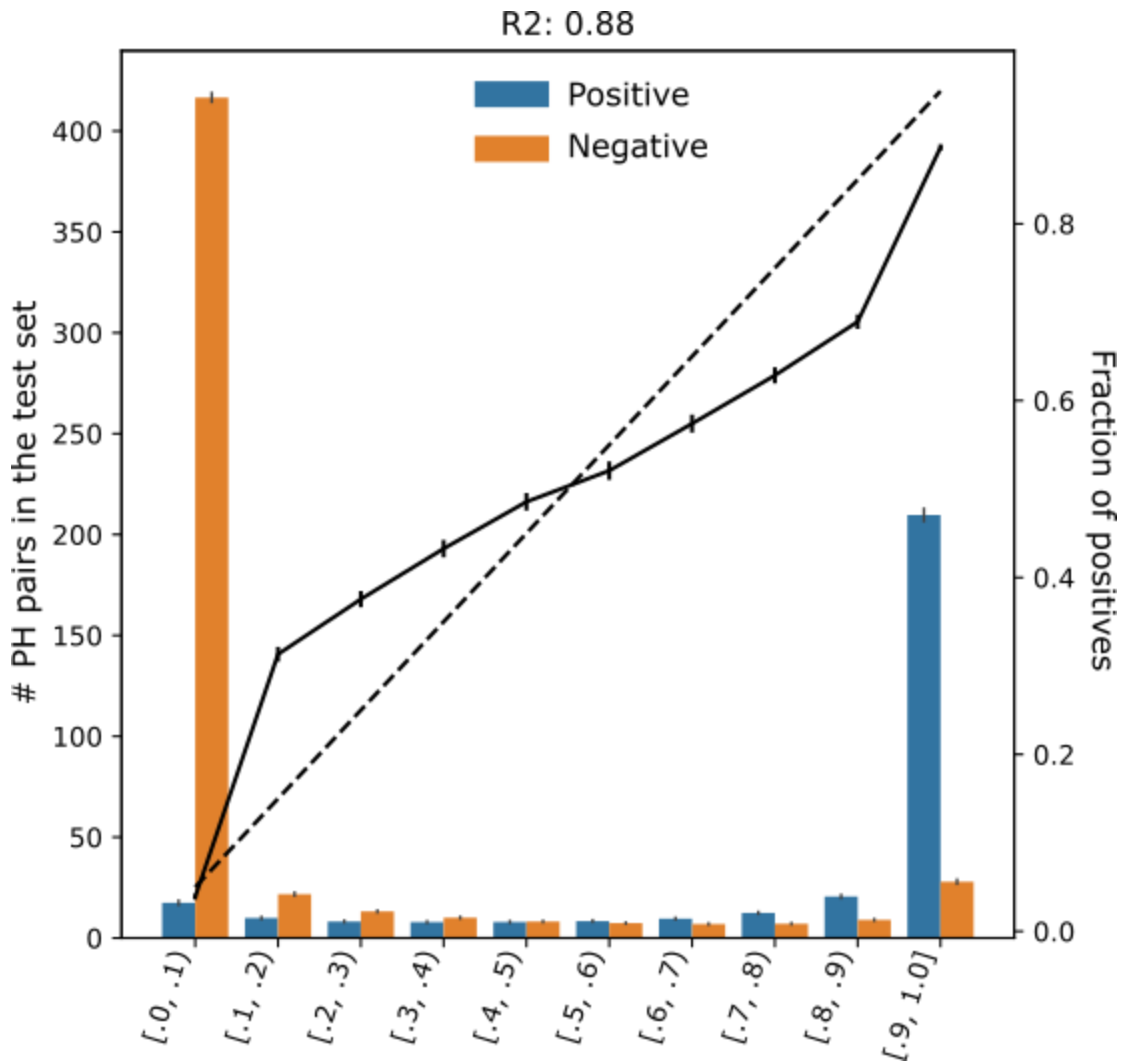


Supplementary Figure 26. Precision, recall, and F1 score of four AL sampling methods over the ten rounds of AL.

The box represents the interquartile range, the middle line represents the median, the whisker line extends from minimum to maximum values, and the diamond represents outliers.

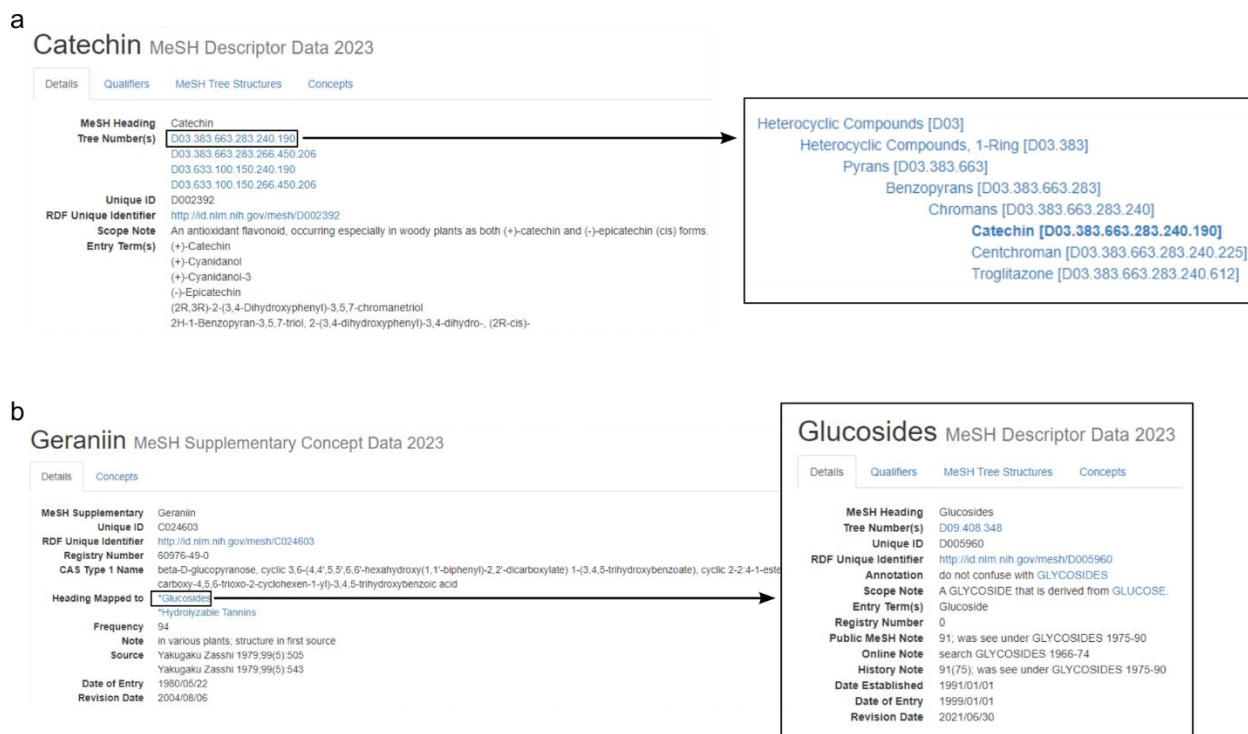


Supplementary Figure 27. Percentage of the positive PH pairs in the training data for each round of 4 different active learning strategies, validation set, and test set. The error bars represent the standard deviation of the percentage of positives ($n = 100$ for 100 random seeds).



Supplementary Figure 28. Calibration plot of the entailment model with ten bins.

The 5-bin R^2 was lower than that of the 10-bin (0.88 vs. 0.94), which was expected due to the small number of samples in the intermediate bins.



Supplementary Figure 29. Sample illustration of querying two types of MeSH data.

a Catechin is a MeSH descriptor data that has a unique MeSH ID with the prefix 'D'. All descriptor data have one unique MeSH ID and can have one or more MeSH tree numbers (4 in the case of catechin). Each tree number encodes the ontological relationships of that MeSH entry in the hierarchical format as seen on the right. **b** Geraniin is MeSH supplementary concept data that has a unique MeSH ID with the prefix 'C'. All supplementary concept data have one unique MeSH ID, but they do not have a MeSH tree number. Instead, supplementary concept data is mapped to one or more descriptor data (e.g., Geraniin is mapped to Glucosides and Hydrolyzable Tannins).

[Fragaria x ananassa](#)

Taxonomy ID: 3747 (for references in articles please use NCBI:txid3747)

current name

Fragaria × *ananassa* (Weston) Duchesne ex Rozier

Genbank common name: **strawberry**

NCBI BLAST name: **eudicots**

Rank: **species**

Genetic code: [Translation table 1 \(Standard\)](#)

Mitochondrial genetic code: [Translation table 1 \(Standard\)](#)

Plastid genetic code: [Translation table 11 \(Bacterial, Archaeal and Plant Plastid\)](#)

Other names:

heterotypic synonym

Fragaria ananassa

heterotypic synonym

Fragaria chiloensis × *Fragaria virginiana*

heterotypic synonym

Fragaria virginiana × *Fragaria chiloensis*



[Lineage \(full\)](#)

[cellular organisms](#); [Eukaryota](#); [Viridiplantae](#); [Streptophyta](#); [Streptophytina](#); [Embryophyta](#); [Tracheophyta](#); [Euphyllophyta](#); [Spermatophyta](#); [Magnoliopsida](#); [Mesangiospermae](#); [eudicotyledons](#); [Gunneridae](#); [Pentapetalae](#); [rosids](#); [fabids](#); [Rosales](#); [Rosaceae](#); [Rosoideae](#); [Potentilleae](#); [Fragariinae](#); [Fragaria](#)

Supplementary Figure 30. Sample illustration of an NCBI taxonomy entry. Strawberry (*Fragaria x ananassa*) with NCBI taxonomy ID 3747 has a taxonomic lineage with a root node 'cellular organism' down to an immediate parent node 'Fragaria'.

COMPOUND SUMMARY

Stearic Acid

PubChem CID	5281
Structure	 <p>2D Crystal</p> <p>Find Similar Structures</p>
Chemical Safety	 <p>Irritant</p> <p>Laboratory Chemical Safety Summary (LCSS) Datasheet</p>
Molecular Formula	$C_{18}H_{36}O_2$ or $CH_3(CH_2)_{16}COOH$
Synonyms	<p>stearic acid Octadecanoic acid 57-11-4 n-Octadecanoic acid Stearophanic acid</p> <p>More...</p>
Molecular Weight	284.5

Cite Download

CONTENTS

- Title and Summary
- 1 Structures
- 2 Names and Identifiers
- 3 Chemical and Physical Properties
- 4 Spectral Information
- 5 Related Records
- 6 Chemical Vendors
- 7 Drug and Medication Information
- 8 Food Additives and Ingredients
- 9 Pharmacology and Biochemistry
- 10 Use and Manufacturing
- 11 Identification
- 12 Safety and Hazards
- 13 Toxicity
- 14 Associated Disorders and Diseases
- 15 Literature
- 16 Patents
- 17 Interactions and

Supplementary Figure 31. Illustration of a PubChem entry. *Stearic acid* is the most common chemical name given its structure. When the most common name did not return evidence during the link prediction validation query, we used the non-ID synonyms shown on the front page. In this example, the synonyms used for searching were *Octadecanoic acid*, *n-Octadecanoic acid*, and *Stearophanic acid*. PubChem heuristically sorts chemical names based on <https://pubchem.ncbi.nlm.nih.gov/docs/compounds#section=Name-Weighting>. There are more synonyms if the user clicks the *More* icon, but we decided not to use the complete list of synonyms, which would excessively increase the search space.

Appendix C. Supplementary tables

Supplementary Table 1. Comparison of our *E. coli* knowledge graph with individual sources used for creating the knowledge graph. We integrated data from 10 different sources to create a comprehensive *E. coli* knowledge graph.

Source	Entity types					Knowledge discovery method	# of triplets
	Gene	Antibiotic	Molecular function	Biological process	Cellular component		
CARD	72	33	-	-	-	Curation of existing knowledge	147
GO	3,672	-	1,782	1,522	152	Curation of existing knowledge	17,739
Liu et al.	3,851	22	-	-	-	MIC profile	55,877
Tamae et al.	3,850	7	-	-	-	MIC profile	26,926
Shaw et al.	3,839	4	-	-	-	Expression profile	15,327
Nichols et al.	3,666	51	-	-	-	Growth profile	186,941
Zhou et al.	47	31	-	-	-	Phenotype microarray	1,457
Soo et al.	4,294	44	-	-	-	Phenotype microarray	188,936
hiTRN	3,686	-	-	-	-	Curation of existing knowledge	101,878
Girgis et al.	3,813	17	-	-	-	Growth profile	63,636
Total							658,864
Ours	4,488	104	1,782	1,522	152	Curation of existing knowledge	651,758

Supplementary Table 2. The number of triplets for each triplet type in the knowledge graph. The domain represents the type of entities that can act as a subject node for a certain predicate, while the range represents the type of entities that can act as an object node for a certain predicate.

Domain	Predicate	Range	# of triplets
gene	activates	gene	2,549
	no activates		48,312
	represses		2,473
	no represses		48,544
	upregulated by antibiotic after 30 minutes	antibiotic	100
	not upregulated by antibiotic after 30 minutes		15,227
	confers resistance to antibiotic		3,642
	confers resistance to antibiotic after 30 minutes		100
	confers resistance to antibiotic after 15 hours		1,613
	confers resistance to antibiotic after 18 hours		1,086
	confers resistance to antibiotic after 36 hours		78
	confers resistance to antibiotic after 3 days		576
	confers resistance to antibiotic after 7 days		59
	confers no resistance to antibiotic after 30 minutes		15,227
	confers no resistance to antibiotic after 15 hours		185,328
	confers no resistance to antibiotic after 18 hours		54,753
	confers no resistance to antibiotic after 36 hours		1,379
	confers no resistance to antibiotic after 3 days		63,027
	confers no resistance to antibiotic after 7 days		188,745
	has		molecular function
	is involved in	biological process	6,481
	is part of	cellular component	4,313
	targeted by	antibiotic	31
	Total		

Supplementary Table 3. The confusion matrix of the computational inconsistency resolver for different levels of inconsistency. Wet-lab validation results are considered to be ground truth.

Level 1		Actual			
		True	False		
Predicted	True	5	30	14.3%	Precision
	False	2	199	99.9%	NPV
		71.4%	86.9%	F1	23.8%
		Recall	Specificity		

Level 2		Actual			
		True	False		
Predicted	True	7	19	26.9%	Precision
	False	0	210	100.0%	NPV
		100.0%	91.7%	F1	42.4%
		Recall	Specificity		

Level 3		Actual			
		True	False		
Predicted	True	7	14	33.3%	Precision
	False	0	215	100.0%	NPV
		100.0%	93.9%	F1	50.0%
		Recall	Specificity		

Supplementary Table 4. The number of triplets for each triplet type in the modified knowledge graph. We used a modified version of the knowledge graph where the temporal information was removed.

Domain	Predicate	Range	# of triplets
gene	activates	gene	2,549
	no activates		48,312
	represses		2,473
	no represses		48,544
	upregulated by antibiotic	antibiotic	100
	not upregulated by antibiotic		15,227
	confers resistance to antibiotic		1,606
	confers no resistance to antibiotic		357,068
	has	molecular function	8,115
	is involved in	biological process	6,481
	is part of	cellular component	4,313
	targeted by	antibiotic	31
Total			494,819

Supplementary Table 5. The confusion matrix for different types of hypotheses generators. The entries show the average and standard deviation of the 5-fold cross-validation.

PRA		Actual			
		True	False		
Predicted	True	50.2 ± 9.2	317.0 ± 77.0	13.9% ± 1.4%	Precision
	False	269.0 ± 9.3	15323.8 ± 82.3	98.3% ± 0.1%	NPV
		15.7% ± 2.9%	98.0% ± 0.5%	F1	14.6% ± 1.2%
		Recall	Specificity		

MLP		Actual			
		True	False		
Predicted	True	100.8 ± 5.7	281.2 ± 56.5	26.9% ± 3.4%	Precision
	False	218.4 ± 5.6	15359.6 ± 48.6	98.6% ± 0.0%	NPV
		31.6% ± 1.8%	98.2% ± 0.4%	F1	28.9% ± 1.9%
		Recall	Specificity		

Stacked		Actual			
		True	False		
Predicted	True	77.6 ± 7.9	117.8 ± 24.2	40.3% ± 4.2%	Precision
	False	241.6 ± 8.2	15523.0 ± 43.2	98.5% ± 0.0%	NPV
		24.3% ± 2.5%	99.2% ± 0.2%	F1	30.1% ± 1.7%
		Recall	Specificity		

TransE		Actual			
		True	False		
Predicted	True	91.4 ± 10.0	436.0 ± 93.9	17.7% ± 2.0%	Precision
	False	227.8 ± 10.2	15204.8 ± 106.4	98.5% ± 0.1%	NPV
		28.6% ± 3.2%	97.2% ± 0.6%	F1	21.7% ± 1.1%
		Recall	Specificity		

TransD		Actual			
		True	False		
Predicted	True	68.6 ± 18.0	182.2 ± 54.7	27.5% ± 2.6%	Precision
	False	250.6 ± 18.3	15458.6 ± 67.2	98.4% ± 0.1%	NPV
		21.5% ± 5.6%	98.8% ± 0.4%	F1	23.7% ± 3.8%
		Recall	Specificity		

TuckER		Actual			
		True	False		

Predicted	True	97.6 ± 3.6	220.2 ± 41.2	31.1% ± 4.5%	Precision
	False	221.6 ± 3.8	15420.6 ± 43.7	98.6% ± 0.0%	NPV
		30.6% ± 1.2%	98.6% ± 0.3%	F1	30.8% ± 2.4%
	Recall	Specificity			

Supplementary Table 6. Overview of the experimental setup used to find the ARGs for each source. Source characteristics of the 7 sources that are related to ARGs are shown here.

Source	<i>E. coli</i> parent strain	Strain characteristics	Method	Media	Temperature	Length of exposure
Liu et al.	BW25113	Single knockout genes (Full Keio collection)	MIC	LB agar	37°C	18 hrs
Tamae et al.	BW25113	Single knockout genes (Full Keio collection)	MIC	LB Agar	37°C	18 hrs
Shaw et al.	MG1655	Wild type	Gene expression (microarray)	LB	37°C	30 min
Nichols et al.	BW25113	Single knockout genes (Keio collection)	Growth profile	LB agar	37°C	14-16 hrs
	BW25113	SPA-tagged derivatives of essential genes	Quantitative growth scores	LB agar	37°C	14-16 hrs
	BW25113	Point-mutants	Quantitative growth scores	LB agar	37°C	14-16 hrs
	BW25113	DAS-tagged essential genes	Quantitative growth scores	LB agar	37°C	14-16 hrs
	BW25113	Truncated genes	Quantitative growth scores	LB agar	37°C	14-16 hrs
	MG1655	Deletion of sRNA and small proteins	Quantitative growth scores	LB agar	37°C	14-16 hrs
Zhou et al.	BW25113	Deletion of two component systems	Phenotype MicroArrays	Various	36°C	24 or 48 hrs
Soo et al.	DH5 α -E	Overexpression of genes with plasmid (ASKA)	Phenotype MicroArrays	IF10	37°C	up to 7 days
Girgis et al.	MG1655 Δ lacZ	Transposon-mutagenized library	Genetic footprinting	M9	37°C	2-4 days

Supplementary Table 7. Hyperparameter investigation of TruthFinder. Optimal hyperparameters are compared under the different configurations of the simulated datasets (number of sources and error rate of sources). We chose each of the two TruthFinder parameters with the one that appears most often as optimal among the configurations explored. We find the performance of the chosen parameters has little difference from the performance optimized for each of the dataset configurations.

Parameters of synthetic dataset		Optimal parameters of TruthFinder		Mean accuracy	Chosen parameters of TruthFinder		Mean Accuracy
Sources	Error rate	ρ	γ		ρ	γ	
3	0.1	2.0	0.8	0.688	1.8	0.8	0.676
3	0.4	1.8	0.2	0.587	1.8	0.8	0.582
9	0.1	1.8	0.8	0.952	1.8	0.8	0.952
9	0.4	1.8	0.8	0.688	1.8	0.8	0.688

Supplementary Table 8. Impact of the size of knowledge sources in the accuracy of inconsistency correction (PooledInvestment). The impact is tested under the different configurations of the simulated datasets (number of sources and error rate of sources).

Parameters of the synthetic dataset			Accuracy
# of sources	Error rate	# of triplets per source	
3	0.1	1,000	87.07
		2,000	88.62
		10,000	90.20
		20,000	90.04
	0.2	1,000	77.26
		2,000	79.68
		10,000	79.27
		20,000	80.58
	0.3	1,000	70.96
		2,000	70.44
		10,000	69.96
		20,000	70.01
	0.4	1,000	61.11
		2,000	62.11
		10,000	60.03
		20,000	59.48
5	0.1	1,000	98.28
		2,000	97.02
		10,000	97.83
		20,000	97.83
	0.2	1,000	89.88
		2,000	91.38
		10,000	91.67
		20,000	91.89
	0.3	1,000	83.00
		2,000	81.86
		10,000	80.93
		20,000	80.89
	0.4	1,000	65.30
		2,000	66.47
		10,000	66.82
		20,000	66.20
7	0.1	1,000	99.43
		2,000	99.42
		10,000	99.33

	0.2	20,000	99.43
		1,000	96.49
		2,000	95.34
		10,000	95.95
		20,000	95.79
	0.3	1,000	86.71
		2,000	87.28
		10,000	86.40
		20,000	86.33
	0.4	1,000	71.46
		2,000	69.05
		10,000	69.96
20,000		70.32	
9	0.1	1,000	100.00
		2,000	99.83
		10,000	99.82
		20,000	99.87
	0.2	1,000	97.22
		2,000	97.69
		10,000	97.65
		20,000	97.71
	0.3	1,000	88.69
		2,000	88.96
		10,000	89.51
		20,000	89.60
	0.4	1,000	73.01
		2,000	73.06
		10,000	72.69
		20,000	73.14

Supplementary Table 9. Path features used by the PRA for the first iteration of hypothesis generation. The PRA used the path features below for training a logistic regression model to infer missing edges in the knowledge graph. Weights are proportionate to the importance of the corresponding path features.

Path Feature	Weight
<i>gene</i> $\xrightarrow{\text{is involved in}}$ <i>biological_process</i> $\xrightarrow{\text{is involved in}^{-1}}$ <i>gene</i> $\xrightarrow{\text{confers resistance to antibiotic}}$ <i>antibiotic</i>	5.25
<i>gene</i> $\xrightarrow{\text{has}}$ <i>molecular_function</i> $\xrightarrow{\text{has}^{-1}}$ <i>gene</i> $\xrightarrow{\text{confers resistance to antibiotic}}$ <i>antibiotic</i>	2.75
<i>gene</i> $\xrightarrow{\text{upregulated by antibiotic}}$ <i>antibiotic</i> $\xrightarrow{\text{confers resistance to antibiotic}^{-1}}$ <i>gene</i> $\xrightarrow{\text{upregulated by antibiotic}}$ <i>antibiotic</i>	2.37
<i>gene</i> $\xrightarrow{\text{upregulated by antibiotic}}$ <i>antibiotic</i>	1.29
<i>gene</i> $\xrightarrow{\text{upregulated by antibiotic}}$ <i>antibiotic</i> $\xrightarrow{\text{confers resistance to antibiotic}^{-1}}$ <i>gene</i> $\xrightarrow{\text{confers resistance to antibiotic}}$ <i>antibiotic</i>	0.74
<i>gene</i> $\xrightarrow{\text{upregulated by antibiotic}}$ <i>antibiotic</i> $\xrightarrow{\text{upregulated by antibiotic}^{-1}}$ <i>gene</i> $\xrightarrow{\text{confers resistance to antibiotic}}$ <i>antibiotic</i>	0.66

Supplementary Table 10. Path features used by the PRA for the second iteration of hypothesis generation. The path features identified by the PRA vary as the knowledge graph evolves.

Path Feature	Weight
$gene \xrightarrow{\text{is involved in}} biological_process \xrightarrow{\text{is involved in}^{-1}} gene \xrightarrow{\text{confers resistance to antibiotic}} antibiotic$	7.36
$gene \xrightarrow{\text{has}} molecular_function \xrightarrow{\text{has}^{-1}} gene \xrightarrow{\text{confers resistance to antibiotic}} antibiotic$	4.31
$gene \xrightarrow{\text{upregulated by antibiotic}} antibiotic$	3.80
$gene \xrightarrow{\text{upregulated by antibiotic}} antibiotic \xrightarrow{\text{confers resistance to antibiotic}^{-1}} gene \xrightarrow{\text{upregulated by antibiotic}} antibiotic$	1.37

Supplementary Table 11. Hypothesis generator results trained on a single source. Three hypothesis generators PRA, MLP, and stacked were trained on a single source to test if the hypothesis generator trained using our knowledge graph predicts better associations than the ones trained using individual sources or the best alternative.

Sources	PRA AUCPR	MLP AUCPR	Stacked AUCPR	Comment
hiTRN	-	-	-	No CRA predicate.
GO	-	-	-	No CRA predicate.
Shaw et al.	-	-	-	No CRA predicate.
Zhou et al.	-	-	-	Not enough training data.
Nichols et al.	-	0.21±0.01	-	Cannot train PRA as these sources only contain a single CRA predicate.
Tamae et al.	-	0.38±0.07	-	
Liu et al.	-	0.46±0.04	-	
Girgis et al.	-	0.12±0.03	-	
Soo et al.	-	0.13±0.11	-	
CARD	-	-	-	Not enough training data.
Our KG	0.11±0.01	0.22±0.01	0.28±0.03	-

Supplementary Table 12. Dissemination of novel ARGs across the human digestive system.

Novel ARGs	# of samples containing ARG / Total # of samples	Percentage
<i>lrp</i>	7,292 / 94,342	7.73%
<i>rbsK</i>	8,062 / 94,342	8.55%
<i>qorB</i>	1,963 / 94,342	2.08%
<i>hdfR</i>	8,292 / 94,342	8.79%
<i>ftsP</i>	628 / 94,342	0.67%

Supplementary Table 13. PCR primers used to amplify the kanamycin cassette.

Target gene	Primer	Sequence (5→3)
<i>lrp</i>	Forward	ACCAGGCATTGCGCGCCGTTAATCCCTCTGGGTTTCGGTCTATCGTGATG ATTCCGGGGATCCGTCGACC
	Reverse	TCAAACACTACAGCGATTTTGCACCTGTTCCGTGTTAGCGTGTCTTAATAACCAG TGTAGGCTGGAGCTGCTTCG
<i>proV</i>	Forward	CATGCCAGAAGCAAATTCAGGGTTGTCTCAGATTCTGAGTATGTTAGGGTATTCCGGGGATCCGTCGACC
	Reverse	CTGTGCGGTATCCACGGATTCGTTTGATCAGCCATTGTTACCCCCTC TGTAGGCTGGAGCTGCTTCG
<i>rbsK</i>	Forward	AAAGAAAAGCAGGGCACGCGCCACCCTAACCGGTGGCGCACTTTGACGTG ATTCCGGGGATCCGTCGACC
	Reverse	CGGGCAACATCTTTCATAGTAGCCAAGCGTTACCCCTGCTGATGTAAAAA TGTAGGCTGGAGCTGCTTCG
<i>ftsP</i>	Forward	GTTATTGTAGAAATCATTTTTTCAGGCACAACCTCTTAGCCTGTTTTACATATTCCGGGGATCCGTCGACC
	Reverse	TGCGCTATTCAGACCCGTACTIONCGGACGCTTTACGACGCTGGATTACCCAGTGTAGGCTGGAGCTGCTTCG
<i>yifA</i>	Forward	GACAGAGTGTAACAACAACATTTAAATCATAACGACAAATAATTTTGTGATTCCGGGGATCCGTCGACC
	Reverse	AAGTTCCTTCTTTTTCTTTTCATCATTTTCATTGTTCCAGCACATCTGTAGGCTGGAGCTGCTTCG

Supplementary Table 14. Confusion matrix of the entailment models (400 models from four active learning strategies) trained on the entire training dataset (the final active learning round) for the test set.

		Ground Truth					
		Positive	Negative			Precision	NPV
Prediction	Positive	260.6 ± 20.3	58.3 ± 15.3	318.9 ± 33.9	0.82 ± 0.03		
	Negative	51.4 ± 20.3	469.7 ± 15.3	521.1 ± 33.9	0.90 ± 0.03		
		312	528				
		0.84 ± 0.07	0.89 ± 0.03	0.87 ± 0.01	0.83 ± 0.03		
		Recall	Specificity	Accuracy	F1		

Supplementary Table 15. Confusion matrix of the PH pairs in the test set separated by which section the premise is from. In addition to the named entity recognition results for each premise, LitSense API also returns where in the literature the premise is extracted from. N/A refers to any other sections aside from the commonly used 8 sections listed below.

Section	Precision	Recall	F1	TP	FP	FN	TN
N/A	0.84	0.93	0.88	468	86	37	499
Intro	0.91	0.94	0.93	602	57	40	206
Discussion	0.83	0.93	0.88	285	57	20	160
Abstract	0.77	0.88	0.82	484	144	66	1,000
Methods	0.89	0.81	0.85	81	10	19	613
Results	0.80	0.88	0.84	254	63	36	194
Conclusion	0.74	0.97	0.84	35	12	1	31
Title	0.75	0.89	0.81	65	22	8	89
Table	0.33	1.00	0.50	1	2	0	31

Supplementary Table 16. Potential food-chemical associations generated by the link-prediction (LP) model. Validation of LP-generated (food, contains, chemical) hypotheses with scientific literature identified 355 of them to be true. However, we did not find any evidence for the following 11 pairs with a probability >80%.

Food (scientific name; NCBI taxonomy)	Chemical (PubChem CID)	Probability
Japanese persimmon (<i>Diospyros kaki</i> ; 35925)	3-Rhamnosyl-Glucosyl Quercetin (156963207)	92.0% ± 2.9%
Chervil (<i>Anthriscus cerefolium</i> ; 40888)	Loxoprofen (3965)	91.9% ± 3.0%
Dudaim melon (<i>Cucumis melo var. dudaim</i> ; 2034236)	Matairesinol (119205)	90.5% ± 6.4%
Bearded tooth (<i>Hericium erinaceus</i> ; 91752)	Lumisterol (6436872)	90.2% ± 6.2%
Chive (<i>Allium schoenoprasum</i> ; 74900)	Salicylic acid (338)	90.1% ± 5.9%
Chickpea (<i>Cicer arietinum</i> ; 3827)	Triglyceride (5460048)	90.1% ± 3.1%
Cumin (<i>Cuminum cyminum</i> ; 52462)	Sodium caffeate (23694762)	90.0% ± 3.2%
Chinese cinnamon (<i>Cinnamomum aromaticum</i> ; 119260)	Phenol (996)	87.0% ± 5.9%
Atlantic cod (<i>Gadus morhua</i> ; 8049)	Beta-carotene (5280489)	84.3% ± 9.2%
Butternut (<i>Juglans cinerea</i> ; 91214)	2-Hydroxy-1,4-naphthoquinone (6755)	83.4% ± 13.9%
Turmeric (<i>Curcuma longa</i> ; 136217)	3-Rhamnosyl-Glucosyl Quercetin (156963207)	81.1% ± 10.4%

Supplementary Table 17. Different sources of data in the FoodAtlas KG and their corresponding triplet types.

Source	Head Type	Relation Type	Tail Type	Quality
<i>FoodAtlas_{annotation}</i>	<i>Cellular organism (food)</i>	<i>contains</i>	<i>chemical</i>	High
<i>FoodAtlas_{annotation}</i>	<i>food - part</i>	<i>contains</i>	<i>chemical</i>	High
<i>FoodAtlas_{annotation}</i>	<i>Cellular organism (food)</i>	<i>hasPart</i>	<i>food - part</i>	High
<i>FoodAtlas_{entailment_prediction}</i>	<i>Cellular organism (food)</i>	<i>contains</i>	<i>chemical</i>	Medium
<i>FoodAtlas_{entailment_prediction}</i>	<i>food - part</i>	<i>contains</i>	<i>chemical</i>	Medium
<i>FoodAtlas_{entailment_prediction}</i>	<i>Cellular organism (food)</i>	<i>hasPart</i>	<i>food - part</i>	Medium
<i>FoodAtlas_{link_prediction}</i>	<i>Cellular organism (food)</i>	<i>Contains</i>	<i>Chemical</i>	Low
<i>FoodAtlas_{MeSH}</i>	<i>chemical</i>	<i>isA</i>	<i>chemical</i>	Medium
<i>FoodAtlas_{NCBI}</i>	<i>Cellular organism</i>	<i>hasChild</i>	<i>Cellular organism</i>	Medium
<i>FoodAtlas_{Frida}</i>	<i>Cellular organism (food)</i>	<i>contains</i>	<i>chemical</i>	Medium & Low
<i>FoodAtlas_{Phenol-Explorer}</i>	<i>Cellular organism (food)</i>	<i>contains</i>	<i>chemical</i>	Medium & Low
<i>FoodAtlas_{FDC}</i>	<i>Cellular organism (food)</i>	<i>contains</i>	<i>chemical</i>	Low

Supplementary Table 18. Food groups excluded in external databases.

Frida	Phenol-Explorer
Biscuits and cookies	Alcoholic beverages
Boiled, smoked, cured or dried meat	Coffee and cocoa
Breast milk and infant formula	Cereal products
Canned fruit products	Cocoa beverage – Chocolate
Canned legumes	Coffee beverage - Arabica Coffee beverages
Canned vegetable products	Coffee beverage - Robusta Coffee beverages
Cold cuts	Coffee beverage - Unknown Coffee beverages
Condiments	Jams - Berry jams
Fermented milk products	Jams - Drupe jams
Firm rennet cheese	Jams - Pome jams
Marmelade, jelly etc.	Other seasonings
Other legume products	Soy and soy products
Other meat and fresh meat products	Soy drinks
Other vegetable products	Spices - Spice blends
Potato chip and snacks	Tea infusions
Processed cheese	
Unfermented milk products	
Yeast and baking powder	

Supplementary Table 19. Comparison of the number of associations in external databases. We considered a reference indexed if the corresponding ISSN or journal title could be found in at least one of AGRICOLA, CABI, WoS, and Scopus. Food was considered indexed if it was associated with any identifier (including scientific name). A chemical was considered indexed if it was associated with any identifier, not including chemical formula or molecular weight. Note that the numbers of the final column are different from the numbers of triplets merged into FoodAtlas because FoodAtlas requires triplets to not only be indexed but also to specifically have NCBI Taxonomy IDs for foods and PubChem CIDs or MeSH IDs for chemicals; Also, triplets without evidence, e.g., FDC, were added as low-quality triplets to FoodAtlas.

Source	# Associations	# Associations w/ ref. (# Ref.)	# Associations w/ indexed ref. (# Indexed ref.)	# Associations w/ indexed ref., food, and chemical
FDC	135,073	0 (0)	0 (0)	0
Frida	111,098	97,111 (377)	4,367 (54)	2,830
Phenol-Explorer	7,486	7,486 (1,308*)	7,486 (1,308*)	5,285
<i>FoodAtlas_{annotation}</i>	3,979	3,979 (1,385)	3,979 (1,385)	3,979
<i>FoodAtlas_{prediction}</i>	225,902	225,902 (146,476)	225,902 (146,476)	225,902

*: All literature sources were curated from WoS according to the Phenol-Explorer website

Supplementary Table 20. Confusion matrix of different knowledge extraction methods on two different datasets: 100 sentences without concentration and 100 sentences with concentration.

Dataset	Method	Precision	Recall	F1	TP	FP	FN
Without concentration	Lit2KG	45.9%	18.2%	26.1%	89	105	400
	GPT3.5	69.6%	59.7%	64.3%	295	129	199
	GPT4	82.8%	86.4%	84.6%	427	89	67
With concentration	Lit2KG	13.6%	4.3%	6.6%	19	121	421
	GPT3.5	55.2%	46.9%	50.7%	205	166	232
	GPT4	72.6%	71.8%	72.2%	316	119	124