Faculty of Informatics
Masaryk University
Czech Republic

A Journey in Biomedical Discovery Informatics: From Ontology Learning to Knowledge Graph Embeddings

Habilitation Thesis

Vít Nováček

2021

Abstract

The thesis begins with a commentary part where I describe the broader context of the presented work. This is followed by two more parts that illustrate the evolution of my research in biomedical discovery informatics from text- and ontology-based solutions to predictive models based on relational machine learning. These latter parts are a collection of previously published works.

Discovery informatics is a loosely defined area of computer science that primarily aims at providing solutions to the information overload problem. More specifically, discovery informatics research tries to come up with new, more efficient ways of acquiring, integrating, organising, augmenting and utilising information and knowledge from data sets that are typically large, heterogeneous, poorly structured, fast-evolving, unreliable—or, in other words, realistic. These characteristics are arguably relevant to virtually any domain of human activity nowadays. However, they are particularly pertinent to life sciences.

The presented work has been motivated chiefly by the following instances of the information overload problem in life sciences: 1. The vast and ever-growing breadth and depth of published articles that can hardly be utilised in a focused and exhaustive manner. 2. The untapped potential of networked biomedical resources for making discoveries.

My solutions to the first specific challenge were based on advances in ontology learning, population and integration. My more recent research motivated by the second challenge has been about enabling new discoveries by applying relational machine learning to link prediction in networked biomedical datasets.

The presented works have been internationally recognised, garnering over 70 citations in the Web of Science database (and roughly three times as many in Google Scholar). Moreover, the research reported in one of the included publications was awarded the 2nd prize in the Elsevier Grand Challenge in Knowledge Enhancement in Life Sciences, where we won $15,000 prize money in a tough competition of over 70 teams from world-renowned institutions like Stanford or Carnegie Mellon University. Another publication reports predictions of previously unknown protein interactions in cancer pathways that were then observed in living human cells—a strong real-world validation of my work. Last but not least, research reported in the last part of the thesis has been taken up for commercial development by Fujitsu Laboratories Limited and led to five patents (two pending, three granted in the USPTO, EPO and/or Japan jurisdictions). That clearly demonstrates also the industrial relevance of my work.

The core of the thesis are nine previously published works (7 high-impact journal articles, 2 A-ranked conference papers). I have been the first author of 4 of them, and senior author of the rest. I have conceptualised and coordinated the research that has led to all the publications, and substantially contributed to each of them (either in terms of implementation of the corresponding prototype, devising validation methodology and pilots, manuscript writing and/or editing, funding acquisition and overall coordination, or combination thereof).

Acknowledgements

I am indebted to the coauthors of the included publications: Loredana Laera, Siegfried Handschuh, Brian Davis, Tudor Groza, Stefan Decker, Gully A.P.C. Burns, Pasquale Minervini, Luca Costabello, Emir Muñoz, Pierre-Yves Vandenbussche, Brian Walsh, Sameh K. Mohamed, Aayah Nounu, Gavin McGauran, David Matallanas, Adrián Vallejo Blanco, Piero Conca, Kamalesh Kanakaraj, Zeeshan Nawaz, Colm J. Ryan, Walter Kolch, Dirk Fey. Many of you will forever be not only my much cherished colleagues, but friends. It has been a blast to coordinate the various teams we have worked in together, and I have learned a lot during our interdisciplinary collaborations.

I also gratefully acknowledge various grants that have supported the work I present here: European Union's 6th IST Framework projects Knowledge Web (EU FP6-507482) and NEPOMUK (EU FP6-027705), Science Foundation Ireland's projects DERI-Líon I, II, Insight 1, 2 and my personal short term travel fellowship (SFI/02/CE1/1131, SFI/08/CE/I1380, SFI/12/RC/2289, SFI/12/RC/2289_2 and SFI/08/CE/I1380–STTF 11 (2), respectively), KI2NA and TOMOE projects funded by Fujitsu Laboratories Ltd., Japan, and the CLARIFY H2020 project funded by European Commission under the grant number 875160.

Other than that, a shout out to the passionate pizzaiolos Ronan and Eugene, and all the amazing Dough Bros staff in Galway—without our team Fridays over your excellent napoletanas, many of the ideas presented here would be much less thoroughly baked. And, last but not least, love to Hanka, Vojta and Eli; you are my islands of sanity in the seas I've had to navigate all the way to the here and now.

Contents

Part I

Context of the Work

Chapter 1

Introduction

Automating discoveries has recently been touted as one of the foremost upcoming challenges for computer science [10]. However, the challenge is far from new. Already decades ago, some research communities were trying hard to come up with computational approaches that could assist people in the process of turning data to information, and to knowledge [3].

According to later works like [5] or [8], such efforts can be conveniently grouped under a common label—discovery informatics. In [8], this field is succinctly defined as a discipline of applied computer science that aims at: i) formal description of the entire scientific process, amenable to machine understanding and processing; ii) design, development, and evaluation of computational artifacts based on such formalisation; iii) application of the resulting artifacts to advance science, either in a fully automated or machine-aided manner.

This thesis tracks the evolution of my discovery informatics research vision over the last 13 years, and can be classified as a coherent collection of previously published works that explore various applied AI approaches to specific problems. All these problems are, however, motivated by one of two high-level information overload challenges in life sciences:

1. The vast and ever-growing breadth and depth of published articles that can hardly be utilised in a focused and exhaustive manner.

2. The untapped potential of networked biomedical resources for making discoveries.

My solutions to the first specific challenge were based on advances in ontology learning, population and integration [26]. My more recent research motivated by the second challenge has been about enabling new discoveries by

applying knowledge graph embeddings [25] to link prediction in networked biomedical datasets.

The rest of the thesis is organised as follows:

- In the remainder of this commentary part, I first introduce the overall concept of the presented research and discuss the most essential related approaches (Section 2). Then I describe my specific contributions (Section 3) and review the impact of the works included in the thesis (Section 4).

- Part II presents three of my published works that aim at making life science literature search more efficient and truly knowledge-based by means of text mining and ontology learning.

- Part III presents six of my published works that pave the way towards using knowledge graph embeddings for making discoveries about drugs, proteins and other biomedical entities of practical interest.

Chapter 2

Overall Concept and Related Work

This chapter consists of two sections where I describe the overall concept of the presented work. This reflects the specific focus of each of the two parts that list the corresponding detailed publications later on. In each section here, I also give a brief overview of the most relevant related works (details are then given in the particular included publications).

2.1 Semantic Literature Search Enabled by Ontology Learning

Bringing scientific publishing largely online in the last decades has made knowledge production and dissemination much more efficient than before. The publication process is faster, since the essential phases like authoring, submission, reviewing, and final typesetting are largely computerised. Moreover, the published content is easily disseminated to global audiences via the Internet. In effect, more and more knowledge is being made available.

However, the big question is whether all this hypothetically *available* knowledge is also truly *accessible*? In our works [19, 17, 16], we claimed the answer to this question is negative, and we showed how this particular instance of the information overload problem could be alleviated in the context of life science publishing.

As of 2009, Medline, a comprehensive source of life sciences and biomedical bibliographic information (available via the PubMed search engine, cf. `https://pubmed.ncbi.nlm.nih.gov/`), hosted over 18 million resources. It

had a growth rate of 0.5 million items per year, which represented around 1,300 new resources per day [23]. Using contemporary publication search engines, one could explore the vast and ever-growing article repositories using relevant keywords. But this was very often not enough. Imagine for instance a junior researcher compiling a survey on various types of leukemia (example taken from [17]). The researcher wants to state and motivate in the survey that *acute granulocytic leukemia* is different from *T-cell leukemia*. Although such a statement might be obvious to a life scientist, one should support it in the survey by a citation of a published paper. Our researcher may be a bit inexperienced in oncology and may not know the proper reference straightaway. Using, e.g., the PubMed search service, it is easy to find articles that contain both leukemia names. Unfortunately, one can find hundreds of such results. It is tedious or even impossible to go through them all to discover one that actually supports that acute granulocytic leukemia is different from T-cell leukemia.

The problem was that asking queries more expressive than (boolean combinations of) mere keywords was virtually impossible. My work presented in [17, 16] addressed this problem by technologies that can operate at an enhanced level, using more expressive concepts and their various relationships. This required collecting, extracting, and interrelating knowledge scattered across large numbers of available life science publications. For that we used the groundwork set forth in one of my earlier works on automated ontology population via text mining [19], and also a distributional semantics framework we introduced in [18].

2.1.1 Overview of Related Work

The first publication [19] included into this thesis deals with biomedical ontology population by new knowledge extracted from textual resources. It defines several practical requirements on an implementation of such type of knowledge integration (namely the ability to process texts and incorporate the contents extracted from them automatically, and resolve potential inconsistencies based on user-defined preferences). While books like [6] or [22] and other, more focused works offered potentially applicable solutions back then, none of them satisfied all the requirements defined in the article. This motivated the development presented there.

The second and third included publication [17, 16] describe two different solutions for knowledge-based search in biomedical literature, where the

queries can express rather complex semantics going beyond key-words and their Boolean combinations. From the end-user's point of view, which is arguably the crucial perspective in this context, these publications were, at that time, most related to works like FindUR [12], Melisa [1], GoPubMed [4] or Textpresso [13]. The award-winning CORAAL tool described in [17] addressed the major limitation and scalability bottleneck of the contemporary state of the art systems—their dependence on manually provided ontologies enabling the semantic search—by extracting the ontologies automatically. The SKIMMR tool [16] then improved CORAAL by using our new distributional semantics framework [18] as the underlying knowledge representation, and by general streamlining of the original technology and front-end motivated by the simple, yet powerful metaphor of machine-aided skim reading.

2.2 Automating Discoveries via Knowledge Graph Embeddings

While letting people search in publications more efficiently can no doubt facilitate progress in life sciences, it cannot directly lead to new discoveries. Therefore, in the more recent stage of my research career, I decided to come up with new approaches that could address this problem.

Complex biological systems can be conveniently modelled as networks of interconnected biological entities [2]. Such networks can then be converted into so called knowledge graphs [7], which are lightweight representations of interlinked knowledge that mitigate many disadvantages of the more traiditional, logics-based ontologies. Owing to their simple design principles, knowledge graphs are robust, and easy to create and maintain, which makes them readily available in practical applications. Yet they are also sufficiently well formalised to support many machine learning and inference tasks, such as relation extraction, link prediction or knowledge base completion [25].

Until as recently as 2017, however, the potential of knowledge graphs in the field of biomedical informatics had been largely unexplored. For instance, our work [14] is arguably the first approach that addresses the problem of discovering adverse drug reactions using knowledge graphs (a substantially extended version [15] of this paper is included here).

My recent works included in Part III of this thesis build on the initial success reported in [14]. More specifically, they show how as diverse problems as discovering adverse drug reactions, polypharmacy side effects, protein drug targets or cancer signalling reactions can be solved simply by

casting them as a link prediction task, and solving that task by means of knowledge graph embeddings [25]. This essentially consists of learning low-rank vector representations of the knowledge graph nodes and edges that preserve the graph's inherent structure. Efficient prediction of new possible links in the original graph using the trained model can in turn lead to new discoveries in the domain the graph represents, as detailed in the particular works included in the thesis.

2.2.1 Overview of Related Work

Various research communities have been trying to apply advanced machine learning techniques to biomedical use cases for well over a decade now. For instance, [28] or [11] came up with state of the art models for predicting adverse drug reactions using sophisticated applications of supervised machine learning. Polypharmacy side effect prediction by relational learning on graph data was investigated in [29]. Competitive machine learning models for prediction of protein drug targets were recently introduced for instance in [21] or [24]. And approaches like [9] or [27] dealt with combining biological background knowledge and general-purpose machine learning and other AI methods to discover signalling protein interactions.

The success of most of these methods, however, was dependent on carefully crafted datasets, lots of extensive (and expensive) manual feature engineering, limited flexibility of the predictive pipelines and/or impractical training times. Motivated by these challenges, the research groups I have been coordinating since 2015 have been coming with an increasingly refined approach based on knowledge graph embeddings. This eventually allowed for targeting many discovery tasks with essentially one method—largely automated conversion of relevant datasets into the knowledge graph form and consequent application of an off-the-shelf relational learning model to achieve state of the art performance. These results, as well as exhaustive lists of related approaches, are described in detail by a series of works included in Part III of the thesis.

Chapter 3

Specific Contributions

This chapter reviews the published works included in the thesis, providing a brief summary of each in a dedicated section (typically based on the abstract of the corresponding publication). I also list my specific personal contributions to each of the works here.

3.1 Semantic Literature Search Enabled by Ontology Learning

3.1.1 Ontology Population and Evolution

The first specific contribution of my research I present in this thesis is based on the following journal article:

- Vít Nováček, Loredana Laera, Siegfried Handschuh, and Brian Davis. Infrastructure for dynamic knowledge integration—automated biomedical ontology extension using textual resources. *Journal of biomedical informatics*, 41(5):816–828, 2008.

The article is a direct follow-up of my master's thesis in ontology learning. It describes the final results of Knowledge Web, an EU Network of Excellence, in which I coordinated a work package on ontology evolution during my research internship and, later on, early PhD studies at the DERI institute of National University of Ireland Galway. More specifically, in the article we present a novel ontology integration technique that explicitly takes the dynamics and data-intensiveness of e-health and biomedicine application domains into account. Changing and growing knowledge, possibly contained in unstructured natural language resources, is handled by application of

cutting-edge Semantic Web technologies. In particular, semi-automatic integration of ontology learning results into a manually developed ontology is employed. This integration relies on automatic negotiation of agreed alignments, inconsistency resolution and natural language generation methods. Their novel combination alleviates the end-user effort in the incorporation of new knowledge to large extent. This allows for efficient application in many practical use cases, as we show in the paper.

I personally contributed to the publication in the following ways:

- I conceptualised the reported research and coordinated the corresponding work.
- I implemented a substantial portion of the presented research prototype.
- I designed, performed and interpreted a validation study for the prototype.
- I wrote the manuscript.

3.1.2 Semantic Literature Search

The second specific contribution of my research I present in this thesis is based on the following journal article:

- Vít Nováček, Tudor Groza, Siegfried Handschuh, and Stefan Decker. Coraal—dive into publications, bathe in the knowledge. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2-3):176–181, 2010.

The article describes our award-winning prototype Coraal. This research was motivated by the shortcomings of prevalent search engines used in online scientific publishing that mostly exploit raw publication data (bags of words) and shallow metadata (authors, key words, citations, etc.). Making use of the knowledge contained implicitly in published texts is still largely not utilised. Following our long-term ambition to take advantage of such knowledge, we have implemented CORAAL (COntent extended by emeRgent and Asserted Annotations of Linked publication data), an enhanced-search prototype and the second-prize winner of the Elsevier Grand Challenge. CORAAL extracts asserted publication metadata together with the knowledge implicitly present in the relevant text, integrates the emergent

content, and displays it using a multiple-perspective search&browse interface. This way we enable semantic querying for individual publications, and convenient exploration of the knowledge contained within them. In other words, recalling the metaphor in the article title, we let the users dive into publications more easily, and allow them to freely bathe in the related unlocked knowledge.

I personally contributed to the publication in the following ways:

- I conceptualised the reported research and coordinated the corresponding work.
- I implemented a substantial portion of the presented research prototype.
- I designed, performed and interpreted a validation study for the prototype.
- I wrote the manuscript.

3.1.3 Distributional Semantics in Semantic Literature Search

The third specific contribution of my research I present in this thesis is based on the following journal article:

- Vít Nováček and Gully APC Burns. Skimmr: Facilitating knowledge discovery in life sciences by machine-aided skim reading. *PeerJ*, 2:e483, 2014.

The article describes a prototype facilitating the process of skim-reading scientific publication via automatically generated, interlinked graphical summaries. This research improved the previous contribution (i.e., the Coraal prototype) by reworking the internal knowledge representation mechanism from scratch, using a distributional semantics framework I developed in the final stage of my PhD research. I also simplified the user interface and user interaction modes based on extensive feedback from various sample users of my research prototypes.

Unlike full reading, "skim-reading" involves the process of looking quickly over information in an attempt to cover more material whilst still being able to retain a superficial view of the underlying content. Within this work, we specifically emulate this natural human activity by providing a dynamic

graph-based view of entities automatically extracted from text. For the extraction, we use shallow parsing, co-occurrence analysis and semantic similarity computation techniques. Our main motivation is to assist biomedical researchers and clinicians in coping with increasingly large amounts of potentially relevant articles that are being published ongoingly in life sciences.

To construct the high-level network overview of articles, we extract weighted binary statements from the text. We consider two types of these statements, co-occurrence and similarity, both organised in the same distributional representation (i.e., in a vector-space model). For the co-occurrence weights, we use point-wise mutual information that indicates the degree of non-random association between two co-occurring entities. For computing the similarity statement weights, we use cosine distance based on the relevant co-occurrence vectors. These statements are used to build fuzzy indices of terms, statements and provenance article identifiers, which support fuzzy querying and subsequent result ranking. These indexing and querying processes are then used to construct a graph-based interface for searching and browsing entity networks extracted from articles, as well as articles relevant to the networks being browsed. Last but not least, we describe a methodology for automated experimental evaluation of the presented approach. The method uses formal comparison of the graphs generated by our tool to relevant gold standards based on manually curated PubMed, TREC challenge and MeSH data.

We provide a web-based prototype (called "SKIMMR") that generates a network of inter-related entities from a set of documents which a user may explore through our interface. When a particular area of the entity network looks interesting to a user, the tool displays the documents that are the most relevant to those entities of interest currently shown in the network. We present this as a methodology for browsing a collection of research articles. To illustrate the practical applicability of SKIMMR, we present examples of its use in the domains of Spinal Muscular Atrophy and Parkinson's Disease. Finally, we report on the results of experimental evaluation using the two domains and one additional dataset based on the TREC challenge. The results show how the presented method for machine-aided skim reading outperforms tools like PubMed regarding focused browsing and informativeness of the browsing context.

I personally contributed to the publication in the following ways:

- I conceptualised the reported research and coordinated the corresponding work.

- I implemented the presented research prototype.
- I designed, performed and interpreted a validation study for the prototype.
- I wrote the manuscript.

3.2 Automating Discoveries via Knowledge Graph Embeddings

3.2.1 Injecting Axioms into Knowledge Graphs

The fourth specific contribution of my research I present in this thesis is based on the following conference paper:

- Pasquale Minervini, Luca Costabello, Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Regularizing knowledge graph embeddings via equivalence and inversion axioms. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 668–683. Springer, 2017.

The paper presents a method for augmenting predictive models that are based on essentially statistical relational machine learning with symbolic knowledge in the form of logics-based axioms. While the application context and validation of the method presented in the paper is not specifically targeted at the domain of life sciences, it nevertheless presents a solid formal groundwork for injecting background knowledge into statistical predictive models trained purely from relational data. Thus it is an important stepping stone towards models that can be made more accurate and/or focused on concrete biological or clinical use cases via selected bits of domain knowledge provided by experts.

As to the method itself, it explores symbolic augmentation of knowledge graph embedding models. Learning embeddings of entities and relations into low-rank continuous vector spaces using neural architectures is an effective method of performing statistical learning on large-scale relational data, such as knowledge graphs. In this paper, we consider the problem of regularising the training of neural knowledge graph embeddings by leveraging external background knowledge. We propose a principled and scalable method for leveraging equivalence and inversion axioms during the learning process, by

imposing a set of model-dependent soft constraints on the predicate embeddings. The method has several advantages: (i) the number of introduced constraints does not depend on the number of entities in the knowledge base; (ii) regularities in the embedding space effectively reflect available background knowledge; (iii) it yields more accurate results in link prediction tasks over non-regularized methods; and (iv) it can be adapted to a variety of models, without affecting their scalability properties. We demonstrate the effectiveness of the proposed method on several large knowledge graphs. Our evaluation shows that it consistently improves the predictive accuracy of several neural knowledge graph embedding models (for instance, the MRR of TransE on WordNet increases by 11%) without compromising their scalability properties.

I personally contributed to the publication in the following ways:

- I helped to conceptualise the reported research and coordinated the corresponding work.
- I performed the formal analysis of specific approaches to injecting the selected axioms into the three embedding models we chose for validating the concept.
- I wrote the corresponding draft sections of the manuscript, commented on the overall structure and contents and edited the final version.

3.2.2 Prediction of Adverse Drug Reactions

The fifth specific contribution of my research I present in this thesis is based on the following journal article:

- Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. *Briefings in bioinformatics*, 20(1):190–202, 2019.

This article is the first major work where we demonstrated the potential of knowledge graphs for solving practically relevant biomedical discovery informatics challenges via off-the-shelf, efficient machine learning methods. More specifically, the article deals with the problem of predicting adverse drug reactions (ADRs). Timely identification of ADRs is highly important in the domains of public health and pharmacology. Early discovery of potential ADRs can limit their effect on patient lives and also make

drug development pipelines more robust and efficient. Reliable in silico prediction of ADRs can be helpful in this context, and thus, it has been intensely studied. Recent works achieved promising results using machine learning. The presented work focuses on machine learning methods that use drug profiles for making predictions and use features from multiple data sources. We argue that despite promising results, existing works have limitations, especially regarding flexibility in experimenting with different data sets and/or predictive models. We suggest to address these limitations by generalisation of the key principles used by the state of the art. Namely, we explore the effects of: (1) using knowledge graphs—machine-readable interlinked representations of biomedical knowledge—as a convenient uniform representation of heterogeneous data; and (2) casting ADR prediction as a multi-label ranking problem. We present a specific way of using knowledge graphs to generate different feature sets and demonstrate favourable performance of selected off-the-shelf multi-label learning models in comparison with existing works. Our experiments suggest better suitability of certain multi-label learning methods for applications where ranking is preferred. The presented approach can be easily extended to other feature sources or machine learning methods, making it flexible for experiments tuned toward specific requirements of end users. Our work also provides a clearly defined and reproducible baseline for any future related experiments.

I personally contributed to the publication in the following ways:

- I helped to conceptualise the reported research and coordinated the corresponding work.
- I advised on the selection of training data sets and the process of their conversion into a knowledge graph form.
- I helped to design the validation study.
- I edited the manuscript.

3.2.3 Integrated Biomedical Knowledge Graph

The sixth specific contribution of my research I present in this thesis is based on the following conference paper:

- Brian Walsh, Sameh K Mohamed, and Vít Nováček. Biokg: A knowledge graph for relational learning on biological data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3173–3180, 2020.

The paper describes a method and a software framework for automated fetching and integration of a number of widely used biomedical data sets covering proteins, genes, drugs, chemicals, pathways and their mutual interactions into a common knowledge graph. We also describe various machine learning benchmark data sets that can be derived from the knowledge graph and thus support training and validation of manifold models addressing challenges like drug side effect or protein interaction prediction.

More specific summary of this contribution is as follows. Knowledge graphs became a popular means for modelling complex biological systems where they model the interactions between biological entities and their effects on the biological system. They also provide support for relational learning models which are known to provide highly scalable and accurate predictions of associations between biological entities. Despite the success of the combination of biological knowledge graph and relation learning models in biological predictive tasks, there is a lack of unified biological knowledge graph resources. This forced all current efforts and studies for applying a relational learning model on biological data to compile and build biological knowledge graphs from open biological databases. This process is often performed inconsistently across such efforts, especially in terms of choosing the original resources, aligning identifiers of the different databases and assessing the quality of included data. To make relational learning on biomedical data more standardised and reproducible, we propose a new biological knowledge graph which provides a compilation of curated relational data from open biological databases in a unified format with common, interlinked identifiers. We also provide a new module for mapping identifiers and labels from different databases which can be used to align our knowledge graph with biological data from other heterogeneous sources. Finally, to illustrate practical relevance of our work, we provide a set of benchmarks based on the presented data that can be used to train and assess the relational learning models in various tasks related to pathway and drug discovery.

I personally contributed to the publication in the following ways:

- I helped to conceptualise the reported research and coordinated the corresponding work.
- I advised on the selection of the original biomedical data sets and the process of their conversion into a knowledge graph form.
- I helped to design the task-specific benchmarks.

- I edited the manuscript.

3.2.4 Discovering Protein Drug Targets

The seventh specific contribution of my research I present in this thesis is based on the following journal article:

- Sameh K Mohamed, Vít Nováček, and Aayah Nounu. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics*, 36(2):603–610, 2020.

The article shows how state of the art performance in a number of standard drug target prediction benchmarks can be achieved by a relatively simple application of a knowledge graph embedding model. More specifically, the article falls under the broad area of computational approaches for predicting drug–target interactions (DTIs) that can provide valuable insights into the drug mechanism of action. DTI predictions can help to quickly identify new promising (on-target) or unintended (off-target) effects of drugs. However, existing models face several challenges. Many can only process a limited number of drugs and/or have poor proteome coverage. The current approaches also often suffer from high false positive prediction rates. In this work, we propose a novel computational approach for predicting drug target proteins. The approach is based on formulating the problem as a link prediction in knowledge graphs (robust, machine-readable representations of networked knowledge). We use biomedical knowledge bases to create a knowledge graph of entities connected to both drugs and their potential targets. We propose a specific knowledge graph embedding model, TriModel, to learn vector representations (i.e. embeddings) for all drugs and targets in the created knowledge graph. These representations are consequently used to infer candidate drug target interactions based on their scores computed by the trained TriModel model. We have experimentally evaluated our method using computer simulations and compared it to five existing models. This has shown that our approach outperforms all previous ones in terms of both area under ROC and precision–recall curves in standard benchmark tests.

I personally contributed to the publication in the following ways:

- I helped to conceptualise the reported research and coordinated the corresponding work.
- I advised on the selection of the original biomedical data sets and the process of their conversion into a knowledge graph form.

- I helped to design the validation study.
- I edited the manuscript.

3.2.5 Prediction of Kinase-Substrate Networks

The eighth specific contribution of my research I present in this thesis is based on the following journal article:

- Vít Nováček, Gavin McGauran, David Matallanas, Adrián Vallejo Blanco, Piero Conca, Emir Muñoz, Luca Costabello, Kamalesh Kanakaraj, Zeeshan Nawaz, Brian Walsh, et al. Accurate prediction of kinase-substrate networks using knowledge graphs. *PLoS computational biology*, 16(12):e1007578, 2020.

The article describes a conceptually new computational approach to prediction of signalling protein interactions (phosphorylations). It addresses the problem from a totally different viewpoint than existing solutions, while outperforming them in two independent benchmarks, and in laboratory validations. Phosphorylation of specific substrates by protein kinases is a key control mechanism for vital cell-fate decisions and other cellular processes. However, discovering specific kinase-substrate relationships is time-consuming and often rather serendipitous. Computational predictions alleviate these challenges, but the current approaches suffer from limitations like restricted kinome coverage and inaccuracy. They also typically utilise only local features without reflecting broader interaction context. To address these limitations, we have developed an alternative predictive model. It uses statistical relational learning on top of phosphorylation networks interpreted as knowledge graphs, a simple yet robust model for representing networked knowledge. Compared to a representative selection of six existing systems, our model has the highest kinome coverage and produces biologically valid high-confidence predictions not possible with the other tools. Specifically, we have experimentally validated predictions of previously unknown phosphorylations by the LATS1, AKT1, PKA and MST2 kinases in human. Thus, our tool is useful for focusing phosphoproteomic experiments, and facilitates the discovery of new phosphorylation reactions. Our model can be accessed publicly via an easy-to-use web interface (LinkPhinder).

I personally contributed to the publication in the following ways:

- I conceptualised the reported research and coordinated the corresponding work.

- I devised and implemented the process of converting site-specific kinase-substrate interaction data into the knowledge graph form and back.
- I advised on implementing the relational machine learning model for predicting the signalling reactions.
- I designed, coordinated and interpreted the computational validation studies for the prototype.
- I wrote the manuscript.

3.2.6 One Method to Rule Them All

The ninth specific contribution of my research I present in this thesis is based on the following journal article:

- Sameh K Mohamed, Aayah Nounu, and Vít Nováček. Biological applications of knowledge graph embedding models. *Briefings in bioinformatics*, 22(2): 1679–1693, 2021.

The article reviews several of our previous works in a comprehensive comparison to other related approaches. This is done to present a survey on solving a broad range of biomedical prediction problems with essentially one suite of techniques based on knowledge graph embedding models. This can be done due to the fact that complex biological systems are traditionally modelled as graphs of interconnected biological entities. These graphs, i.e. biological knowledge graphs, are then processed using graph exploratory approaches to perform different types of analytical and predictive tasks. Despite the high predictive accuracy of these approaches, they have limited scalability due to their dependency on time-consuming path exploratory procedures. In recent years, owing to the rapid advances of computational technologies, new approaches for modelling graphs and mining them with high accuracy and scalability have emerged. These approaches, i.e. knowledge graph embedding (KGE) models, operate by learning low-rank vector representations of graph nodes and edges that preserve the graph's inherent structure. These approaches were used to analyse knowledge graphs from different domains where they showed superior performance and accuracy compared to previous graph exploratory approaches. In this work, we study this class of models in the context of biological knowledge graphs and their different applications. We then show how KGE models can be a natural fit

for representing complex biological knowledge modelled as graphs. We also discuss their predictive and analytical capabilities in different biology applications. In this regard, we present two example case studies that demonstrate the capabilities of KGE models: prediction of drug–target interactions and polypharmacy side effects. Finally, we analyse different practical considerations for KGEs, and we discuss possible opportunities and challenges related to adopting them for modelling biological systems.

I personally contributed to the publication in the following ways:

- I helped to conceptualise the reported research and coordinated the corresponding work.
- I helped to define the scope of the survey and the set of discovery problems to be covered.
- I edited the manuscript.

Chapter 4

Impact

This chapter gives a brief overview of the impact of my work in real world, first in academic and then in industry settings.

4.1 Reception in Academia

The seven articles included in the thesis have been published in internationally-disseminated peer-reviewed journals with an impact factor ranging from 1.897 (Web Semantics Journal, Elsevier) to 11.622 (Briefings in Bioinformatics, Oxford University Press). The other two works appeared in the proceedings of ECML-PKDD and CIKM conferences that are both A-ranked in the CORE conference database.

As of October 2021, the works included in this thesis have been cited by over 70 other publications according to the Web of Science database. The number of citations according to the Google Scholar service is roughly three-times higher. This demonstrates that my research has been acknowledged by the global scientific community and used in many consequent works that are pushing the state of the art further.

In terms of other means of academic recognition, I would like to highlight the following points:

- The work described in [17] has been awarded the 2nd prize in the Elsevier Grand Challenge in Knowledge Enhancement in Life Sciences, where we won $15,000 prize money in a tough competition of over 70 teams from world-renowned institutions like Stanford or Carnegie Mellon University. The competition was judged by world-leading scientists and publishers (such as Eduard H. Hovy, a renowned AI and

NLP researcher, or Emilie Marcus, then the Editor in Chief of Cell, one of the most influential journals in the world across any field of science). The fact that such a prestigious committee considered my work relevant enough to be awarded the prize was a tremendous validation of my research vision and the capability to realise it already at a rather early stage of my career.

- The works described in [17, 16] were used for literature search on a daily basis by clinicians and by representatives of a large US patient organisation (Spinal Muscular Atrophy Foundation). Their positive feedback was another strong signal that I was on the right path in terms of turning my research into societally relevant services.

- The interdisciplinary character of my research vision has allowed me to establish a number of fruitful collaborations with biologists, clinicians and representatives of the pharma industry. Their feedback has been crucial to motivate my work by actual needs of users who can benefit from the research outcomes right away. In many cases, these groups have also provided a strong, realistic validation of my work, which has been priceless.

- Last but not least, the research reported in this thesis was instrumental for building a platform I used to successfully acquire funding for my research groups from various bodies in Europe, USA and Japan. The total amount of funding directly dedicated to groups I have supervised corresponds to ca. €1,779,000, or roughly 43.6 million CZK, as of October 2021. This is yet another confirmation of the international relevance of the presented research.

4.2 Reception in Industry

I have spent a substantial portion of my post-doctoral career (2012-2019) coordinating a research group in a large industrial collaboration fully funded by Fujitsu Laboratories Limited (FLL). FLL has appreciated the relevance and uniqueness of our research by applying for several patents on our behalf (6 in total, 5 of which have been related to the research reported in the publications here, with 3 granted and 2 pending as of October 2021). More importantly, however, the Fujitsu group's research scientists, developers and business units have been taking up our research outcomes. They

have built dedicated internal teams around our prototypes that have been commercialising the resulting applications in Japan (for instance, in 2020, a genomic AI system based on our work was brought to a production stage by a subsidiary company of the Fujitsu group). This clearly demonstrates a tangible impact of my research vision not only in academia, but also in the global corporate world.

More recently, this has been corroborated by my other involvements within the healthcare, pharma and biotech verticals, such as:

- Serving as an AI consultant in a large oncology hospital (Masaryk Memorial Cancer Institute).
- A request by the large biotech company QIAGEN to license experimental and prediction data reported in [20] as a part of their pathway analysis tool.
- An invitation to serve on the Advisory Board of the BioXcel Therapeutics, Inc. pharma company.

I expect to be involved in more such dissemination efforts in the future, which will further help to get my ideas out into the real world.

Bibliography

[1] Jose María Abasolo, M Gómez, and M Melisa. An ontology-based agent for information retrieval in medicine. In *Proceedings of the First International Workshop on the Semantic Web (SemWeb2000)*, pages 73–82, 2000.

[2] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113, 2004.

[3] Brian L Claus and Dennis J Underwood. Discovery informatics: its evolving role in drug discovery. *Drug discovery today*, 7(18):957–966, 2002.

[4] Andreas Doms and Michael Schroeder. Gopubmed: exploring pubmed with the gene ontology. *Nucleic acids research*, 33(suppl_2):W783–W786, 2005.

[5] Yolanda Gil and Haym Hirsh. Discovery informatics: Ai opportunities in scientific discovery. In *2012 AAAI Fall Symposium Series*, 2012.

[6] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media, 2006.

[7] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37, 2021.

[8] Vasant G Honavar. The promise and potential of big data: A case for discovery informatics. *Review of Policy Research*, 31(4):326–330, 2014.

[9] Heiko Horn, Erwin M Schoof, Jinho Kim, Xavier Robin, Martin L Miller, Francesca Diella, Anita Palma, Gianni Cesareni, Lars Juhl Jensen, and Rune Linding. Kinomexplorer: an integrated platform for kinome biology studies. *Nature methods*, 11(6):603–604, 2014.

[10] Hiroaki Kitano. Nobel turing challenge: creating the engine for scientific discovery. *npj Systems Biology and Applications*, 7(1):1–12, 2021.

[11] Mei Liu, Yonghui Wu, Yukun Chen, Jingchun Sun, Zhongming Zhao, Xue-wen Chen, Michael Edwin Matheny, and Hua Xu. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *Journal of the American Medical Informatics Association*, 19(e1):e28–e35, 2012.

[12] Deborah L McGuinness. Ontology-enhanced search for primary care medical literature. In *Proceedings of the Medical Concept Representation and Natural Language Processing Conference*, pages 16–19, 1999.

[13] Hans-Michael Müller, Eimear E Kenny, Paul W Sternberg, and Michael Ashburner. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS biology*, 2(11):e309, 2004.

[14] Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Using drug similarities for discovery of possible adverse reactions. In *AMIA Annual Symposium Proceedings*, volume 2016, page 924. American Medical Informatics Association, 2016.

[15] Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. *Briefings in bioinformatics*, 20(1):190–202, 2019.

[16] Vít Nováček and Gully APC Burns. Skimmr: Facilitating knowledge discovery in life sciences by machine-aided skim reading. *PeerJ*, 2:e483, 2014.

[17] Vít Nováček, Tudor Groza, Siegfried Handschuh, and Stefan Decker. Coraal—dive into publications, bathe in the knowledge. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2-3):176–181, 2010.

[18] Vít Nováček, Siegfried Handschuh, and Stefan Decker. Getting the meaning right: A complementary distributional layer for the web semantics. In *International Semantic Web Conference*, pages 504–519. Springer, 2011.

[19] Vít Nováček, Loredana Laera, Siegfried Handschuh, and Brian Davis. Infrastructure for dynamic knowledge integration—automated biomedical ontology extension using textual resources. *Journal of biomedical informatics*, 41(5):816–828, 2008.

[20] Vít Nováček, Gavin McGauran, David Matallanas, Adrián Vallejo Blanco, Piero Conca, Emir Muñoz, Luca Costabello, Kamalesh Kanakaraj, Zeeshan Nawaz, Brian Walsh, et al. Accurate prediction of kinase-substrate networks using knowledge graphs. *PLoS computational biology*, 16(12):e1007578, 2020.

[21] Rawan S Olayan, Haitham Ashoor, and Vladimir B Bajic. Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics*, 34(7):1164–1173, 2018.

[22] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

[23] Junichi Tsujii. Refine and pathtext, which combines text mining with pathways. *Keynote at Semantic Enrichment of the Scientific Literature*, 2009.

[24] Fangping Wan, Lixiang Hong, An Xiao, Tao Jiang, and Jianyang Zeng. Neodti: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. *Bioinformatics*, 35(1):104–111, 2019.

[25] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[26] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4):1–36, 2012.

[27] Yu Xue, Jian Ren, Xinjiao Gao, Changjiang Jin, Longping Wen, and Xuebiao Yao. Gps 2.0, a tool to predict kinase-specific phosphorylation sites in hierarchy. *Molecular & cellular proteomics*, 7(9):1598–1608, 2008.

[28] Yoshihiro Yamanishi, Edouard Pauwels, and Masaaki Kotera. Drug side-effect prediction based on the integration of chemical and biological spaces. *Journal of chemical information and modeling*, 52(12):3284–3292, 2012.

[29] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

Part II

From Ontology Learning...

Chapter 5

Ontology Population and Evolution

Journal of Biomedical Informatics 41 (2008) 816–828

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Infrastructure for dynamic knowledge integration—Automated biomedical ontology extension using textual resources

Vít Nováček [a,*], Loredana Laera [b], Siegfried Handschuh [a], Brian Davis [a]

[a] Digital Enterprise Research Institute, National University of Ireland, Galway, IDA Business Park, Lower Dangan, Galway, Co. Galway, Ireland
[b] Department of Computer Science, University of Liverpool, UK

ARTICLE INFO

Article history:
Received 1 September 2007
Available online 24 July 2008

Keywords:
Dynamic ontology integration
Ontology evolution
Ontology alignment and negotiation
Ontology learning
Biomedical ontologies
Knowledge acquisition
Lifecycle

ABSTRACT

We present a novel ontology integration technique that explicitly takes the dynamics and data-intensiveness of e-health and biomedicine application domains into account. Changing and growing knowledge, possibly contained in unstructured natural language resources, is handled by application of cutting-edge Semantic Web technologies. In particular, semi-automatic integration of ontology learning results into a manually developed ontology is employed. This integration bases on automatic negotiation of agreed alignments, inconsistency resolution and natural language generation methods. Their novel combination alleviates the end-user effort in the incorporation of new knowledge to large extent. This allows for efficient application in many practical use cases, as we show in the paper.

1. Introduction

Ontologies (formal knowledge bases) on the Semantic Web are often very likely subject to change given the dynamic nature of domain knowledge. Knowledge changes and evolves over time as experience accumulates—it is revised and augmented in the light of deeper understanding; new facts are becoming known while some of the older ones need to be revised and/or retracted at the same time. This holds especially for scientific domains, however, even virtually any industrial domain is dynamic—changes typically occur in product portfolios, personnel structure or industrial processes, which can all be reflected by an ontology in a knowledge management policy.

The domains of e-health and biomedicine are both scientific (biomedical research) and industrial (clinical practice, pharmaceutics). The need for ontologies in biomedicine knowledge and data management has already been recognised by the community. Ontologies can serve as structured repositories giving a shared meaning to data and thus making it possible to process and query them in a more efficient and expressive manner. The shared meaning provided by ontologies also results in facilitation of integration between different medical data formats once they are bound to an ontology. Moreover, the state of the art ontology-based techniques (like alignment or reasoning as described in [39]) can help to integrate the data even if they adhere to different ontologies.

In the biomedical domain, ontology construction is usually a result of a collaboration involving ontology engineers and domain experts, where the knowledge is being extracted and modelled manually. However, it is not always feasible to process all the relevant data and extract the knowledge manually from domain resources, since we might not have a sufficiently large committee of ontology engineers and/or dedicated experts at hand in order to process new data anytime it arrives. This implies a need for automation of knowledge extraction and maintenance processes in dynamic and data-intensive medical environments. If the knowledge is available in textual resources, ontology learning (see [33]) can help in this task. Therefore, a lifecycle of an ontology development process apt for universal application in the medicine domain should also support appropriate mechanisms for the incorporation of dynamically extracted knowledge. In this paper, we introduce such a lifecycle scenario and a novel solution to the dynamic knowledge integration task.

Our efforts have several particular *motivations*. While there has been a great deal of work on ontology learning for ontology construction, e.g. in [10], as well as on manual or collaborative ontology development in [41], relatively little attention has been paid to the user-friendly integration of both approaches within an ontology lifecycle scenario. By user-friendly we mean especially accessible to users who are not experts in ontology engineering (i.e. biomedicine researchers or practitioners). In this paper, we

* Corresponding author. Fax: +353 91 495541.
E-mail addresses: vit.novacek@deri.org (V. Nováček), lori@csc.liv.ac.uk (L. Laera), siegfried.handschuh@deri.org (S. Handschuh), brian.davis@deri.org (B. Davis).

1532-0464/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2008.06.003

introduce our framework for practical handling of dynamic and large data-sets in an ontology lifecycle, focusing particularly on dynamic integration of learned knowledge into manually maintained ontologies. However, the introduced integration mechanism is not restricted only to learned ontologies—arbitrary "external" ontology can be integrated into the primary ontology in question by the very same process.

The dynamic nature of knowledge is one of the most challenging problems not only in biomedicine, but in the whole current Semantic Web research. Here we provide a solution for dealing with these dynamics on a large scale, based on the properly developed connection of ontology learning and dynamic manual development. We do not concentrate on formal specification of respective ontology integration operators, we focus rather on implementation of them,
following certain practical requirements:

(1) the ability to process new knowledge (resources) automatically whenever it appears and when it is inappropriate for human users to incorporate it
(2) the ability to automatically compare the new knowledge with a "master" ontology that is manually maintained and select the new knowledge accordingly
(3) the ability to resolve possible major inconsistencies between the new and current knowledge, possibly favouring the assertions from presumably more complex and precise master ontology against the learned ones
(4) the ability to automatically sort the new knowledge according to user-defined preferences and present it to them in a very simple and accessible way, thus further alleviating human effort in the task of knowledge integration

On one hand, using the automatic methods, we are able to deal with large amounts of changing data. On the other hand, the final incorporation of new knowledge is to be decided by the expert human users, repairing possible errors and inappropriate findings of the automatic techniques. The key to success and applicability is to let machines do most of the tedious and time-consuming work and provide people with concise and simple suggestions on ontology integration.

The main *contribution* of the presented work is two-fold:

- proposal and implementation of a generic algorithm for dynamic integration of knowledge automatically extracted from various unstructured resources (e.g., natural language articles or web pages) into manually maintained formal ontologies (described in Sections 4 and 5)
- presentation of an example application of the implemented algorithm in a task of biomedical ontology extension by integrating knowledge automatically learned from textual domain resources, showing usability of the approach in the context of the presented use cases (Section 6)

The rest of the paper is organized as follows: Section 2 gives basic overview of the essential notions and background of the paper, together with respective relevant references. Section 3 discusses the related work. Section 4 gives an overview of our ontology lifecycle scenario and framework, whereas Section 5 presents the integration of manually designed and automatically learned ontologies in more detail. In Section 6, we describe an example practical application of our integration technique, using real world input data (from the biomedicine research domain). Preliminary evaluation and its discussion is also provided. Section 7 outlines relevant real-world settings, challenges and contributions our framework can bring in these contexts. A related user feedback analysis is provided in Section 7, too. Section 8 summarises the paper and future directions of the presented research.

2. Key notions

In the following list we give a brief description of the essential notions that are relevant for the presented content and describe how they relate to the field of bioinformatics:

- *Semantic Web*—the Semantic Web initiative is generally about giving a formal shared meaning to the data present on the normal world wide web in order to make them fully accessible and "comprehensible" by machines, not only by humans (see [5]). However, the technologies that have been developed within Semantic Web research are applicable to many other fields. In the case of bioinformatics, biomedical data management and decision support, we can exploit for instance Semantic Web methods of intelligent and efficient knowledge representation, reasoning, data integration or knowledge management.
- *ontology*—according to a popular definition in [24], ontology is a representation of shared conceptualisation. As such, ontologies are used for formal representation of knowledge in particular domains, i.e. various subfields of biomedicine.
- *ontology integration*—the process of merging, consolidating and respective analysis and modification of two or more ontologies into one (integrated) ontology (see [38]). The process can be either manual or (semi)automatical.
- *ontology learning*—acquisition of an ontology from unstructured or semi-structured natural language text, typically resources relevant for a particular domain (e.g. web pages, articles or other types of documents). Natural Language Processing and Machine Learning methods are mostly used as a base for ontology learning algorithms (see [33]).
- *ontology alignment* —ontology alignment establishes mappings between concepts and other entities (e.g. relations or instances) in two or more ontologies. Either manually designed mappings (created on the fly or contained in appropriate alignment repositories), or automatically generated ones can be used to align the ontologies (see [16,18]).
- *ontology evolution*—development and maintenance of ontologies in dynamic environments (see [40,36,25]), where the knowledge needs to be updated on regular basis and changes in the domain conceptualisation occur often (e.g. science or business domains, where frequent introduction of new concepts or revision of the old ones is essential).
- *ontology lifecycle*–a methodology or scenario, that describes how the particular phases of the ontology development, maintenance and possibly also exploitation are mutually connected and dependent (see [23,34]).

3. Related work

Within the Semantic Web research, several approaches and methodologies have been defined and implemented in the context of ontology lifecycle and integration. Recent overviews of the state-of-the-art in ontologies and related methodologies can be found in [39] and [23]. However, none of them offers a direct solution to the requirements specified in Section 1.

The *Methontology* methodology by [19] was developed in the *Esperonto* EU project. It defines the process of designing ontologies and extends it towards evolving ontologies. It is provided with an ontology lifecycle based on evolving prototypes (see [20]) and defines stages from specification and knowledge acquisition to configuration management. The particular stages and their

requirements are characterised, but rather in a general manner. The automatic ontology acquisition methods are considered in *Methontology*, however, their concrete incorporation into the whole lifecycle is not covered. The ODESeW and WebODE (see [9]) projects base on Methontology and provide an infrastructure and tools for semantic application development/management, which is in the process of being extended for networked and evolving ontologies. However, they focus rather on the application development part of the problem than on the ontology evolution and dynamic ontology integration parts.

The methods and tools referenced above lack concrete mechanisms that would efficiently deal with the dynamics of realistic domains (so characteristic for instance for e-health and biomedicine). Moreover, the need for automatic methods of ontology acquisition in data-intensive environments is acknowledged, but the role and application of the automatic techniques is usually not clearly studied and implemented. Our approach described in [34] offers a complex picture of how to deal with the dynamics in the general lifecycle scenario. The work we present here implements the fundamental semi-automatic dynamic integration component of the scenario.

There are more specific approaches similar to the one presented by our lifecycle framework. [14] incorporates automatic ontology extraction from a medical database and its consequent population by linguistic processing of corpus data. However, the mechanism is rather task-specific—the ontology is represented in RDF(S) format (see [6]) that is less expressive than the OWL language (see [4]), which we use. The extraction is oriented primarily at taxonomies and does not take the dynamics directly into account. Therefore the approach can hardly be applied in universal settings, which is one of our aims.

Protégé (see [22]) and related PROMPT (see [37]) tools are designed for manual ontology development and semi-automatic ontology merging, respectively. PROMPT provides heuristic methods for identification of similarities between ontologies. The similarities are offered to the users for further processing. However, the direct connection to ontology learning, which we find important for dynamic and data-intensive domains like e-health and biomedicine, is missing.

There are several works addressing directly the topic of ontology integration. Alasoud et al. [1] and Calvanese et al. [7] describe two approaches inspired mainly by database techniques of data mediation and query rewriting in order to provide integrated (global) view on several (local) ontologies. Heflin and Hendler [28] present web ontology integration method using SHOE, a web-based knowledge representation language, and semi-automatically generated alignments. Deen and Ponnamperuma [12] implement a dynamic and automatic ontology integration technique in multi-agent environments, based on relatively simple graph ontology model inclusions and other operations. Again, none of the approaches tackles the requirements we specify in Section 1. Even though the methods propose solutions to the integration problem in general, there is no direct way how to integrate knowledge from unstructured resources, minimising human intervention. Furthermore, there is no emphasis on accessibility of the ontology integration to the laymen users. Our approach is distinguished by the fact that it pays special attention to these features, which we find essential for the application in e-health and/or bioinformatics.

4. DINO—a dynamic ontology lifecycle scenario

Our integration platform is a part of a broader lifecycle scenario (see [34]). We refer to both lifecycle and integration platform by the DINO abbreviation, evoking multiple features of our solution: it reflects three key elements of the lifecycle scenario—*Dynamics*, *INtegration* and *Ontology*; however, the first two parts can also be *Data* and *INtensive*; finally, DINO can be read as *Dynamic INtegration* of *Ontologies*, too. All these features express the primary aim of our efforts—to make the knowledge (integration) efficiently and reasonably manageable in data-intensive and dynamic domains.

Fig. 1 depicts the scheme of the proposed dynamic and application-oriented ontology lifecycle that deals with the problems mentioned as a part of our motivations. Our ontology lifecycle builds on four basic phases of an ontology lifecycle: *creation* (comprises both manual and automatic ontology development and update approaches), *versioning*, *evaluation* and *negotiation* (comprises ontology alignment and merging as well as negotiation among different possible alignments). The four main phases are indicated by the boxes annotated by respective names. Ontologies or their snapshots in time are represented by circles, with arrows expressing the information flow and transitions between them. The boxes labelled A_i present actors (institutions, companies, research teams etc.) involved in ontology development, where A_1 is zoomed-in in order to show the lifecycle's components in detail.

The general dynamics of the lifecycle goes as follows: (1), the community experts and/or ontology engineers develop a relatively precise and complex domain ontology (the *Community* part of the *Creation* component); (2), the experts use means for continuous ontology *evaluation* and *versioning* to maintain high quality and manage changes during the development process, respectively; (3), if the amount of data suitable for knowledge extraction (e.g. domain resources in natural language) is too large to be managed by the community, *ontology learning* takes its place; (4), the ontology learning results are *evaluated* by human experts and eventually integrated (using the *negotiation* component) into the more precise reference community ontology, if the respective extensions have been found appropriate.

The integration in the scenario is based on alignment and merging covered by the *negotiation* component. Its proposal, implementation principles and application in selected e-health use case form the key contribution of this paper (see Sections 5 and 6 for details). The *negotiation* component takes its place also when interchanging or sharing the knowledge with other independent actors in the field. All the phases support ontologies in the standard OWL format. In the following we will concentrate on the integration mechanism. More information on other parts of the lifecycle can be found in [34].

5. Dynamic integration of automatically learned knowledge

The key novelty of the presented lifecycle scenario is its support for incorporation of changing knowledge in data-intensive domains, especially when unstructured data (i.e. natural language) is involved. This is achieved by implementation of a specific integration mechanism introduced in this section. The scheme of the integration process is depicted in Fig. 2.

The integration scheme details the combination of several generic lifecycle components—mainly the (automatic) *creation* and *negotiation*—in the process of incorporation of learned ontologies into a collaboratively developed one. The latter ontology serves as a master, presumably precise model in the process of learned knowledge integration. The master ontology—O_M circle in Fig. 2—is supposed to be developed within a dedicated external application such as Protégé[1]. The DINO integration platform itself is implemented as a respective API library and GUI interface. Simple research prototypes of these applications and user documentation can be downloaded at http://smile.deri.ie/tools/dino.

[1] See http://protege.stanford.edu/.

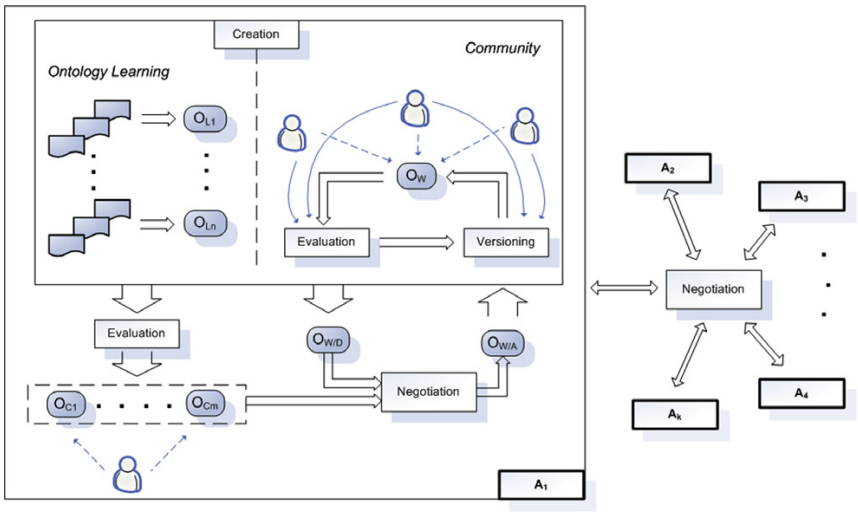


Fig. 1. Dynamic ontology lifecycle scheme.

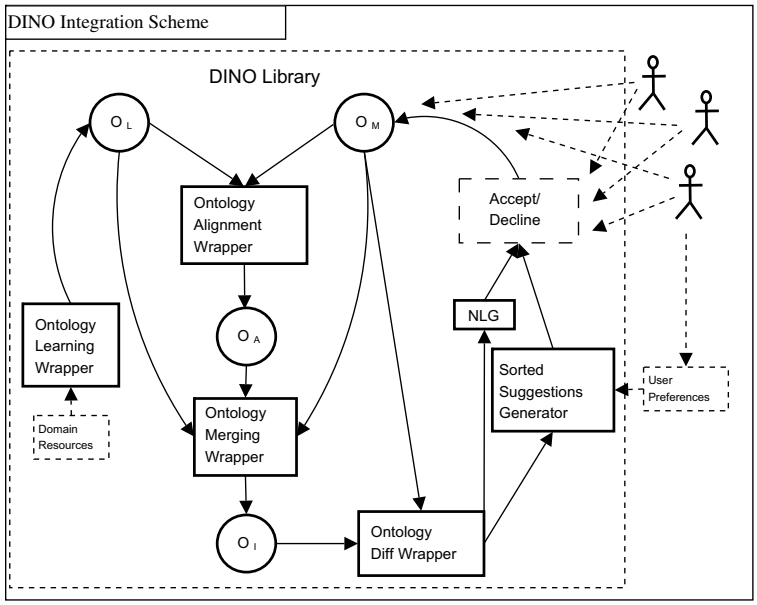


Fig. 2. Dynamic ontology integration scheme.

O_M in Fig. 2 presents a reference for integration with the O_L ontology resulting from the learning process. Ontology Alignment Wrapper produces an alignment ontology O_A that encodes mappings between O_M and O_L. All these ontologies are passed to the Ontology Merging Wrapper that resolves possible inconsistencies and produces integrated ontology O_I. Ontology Diff Wrapper compares O_I with the former master ontology O_M and passes the respective additional statements (not present in O_M) to the NLG and Sorted Suggestions Generator component. NLG (Natural Language Generator) produces a comprehensive natural language representation of all the addition statements. The Sorted Suggestions Generator component outputs the final product of the integration process—particular natural language suggestions on the master ontology extension, sorted according to the user preferences. The suggestions agreed by human users form a base of a next version of the O_M ontology created after the integration. Note that during all phases of integration, we use the former O_M base namespace for all the other ontologies involved. The integration phases outlined in Fig. 2 are described in detail in the sections below.

5.1. Ontology learning wrapper

In this phase, machine learning and NLP methods are used for the processing of relevant resources and extracting knowledge from them (ontology learning). The ontology learning is realised using the Text2Onto framework (see [8]) that is able to extract an ontology from an arbitrary set of textual documents. Due to space restrictions, we cannot properly comment on the methods used for ontology extraction and post-processing in Text2Onto, however, they are described in detail in [32,8]. Note that this component does not tackle selection of the documents the ontology is to be learned from—this task needs to be performed manually by the system users.

In the current implementation, only a restricted subset of possible OWL (DL) constructs is being extracted: `rdfs:subClassOf` axioms, class instances, named class assertions, `owl:disjointWith` axioms and `owl:ObjectProperty` assertions with `rdfs:domain` and `rdfs:range` properties specified. `owl:equivalentClass` relations can be inferred from mutual `rdfs:subClassOf` axioms between particular classes. `owl:equivalentClass` and `owl:sameAs` constructs can also be extracted using the Text2Onto concept/instance similarity determination algorithms, however, their performance, precision and coverage was not found to be sufficient enough, therefore they are not included in the current version of the DINO framework.

We have not performed a rigorous evaluation of the ontology learning step as such, however, the informal precision rate of ontology extraction was about 70% for the sample application described in Section 6 (given by ratio of meaningful axioms to all extracted axioms). Note that even an arbitrary external ontology can be integrated instead of the learned one, however, the integration results are not necessarily complete in the case of more complex ontologies (e.g., containing complex restrictions and anonymous classes). This is due to the fact that the current implementation is tailored specifically to the rather simple learned ontologies.

5.2. Ontology alignment wrapper

When the learned ontology O_L has been created, it has to be reconciled with master ontology O_M since they cover the same domain, but might be structured differently. The reconciliation of these ontologies depends on the ability to reach an agreement on the semantics of the terms used. The agreement takes the form of an alignment between the ontologies, that is, a set of correspondences (or mappings) between the concepts, properties, and relationships in the ontologies. However, the ontologies are developed in different contexts and under different conditions and thus they might represent different perspectives over similar knowledge, so the process by which to come to an agreement will necessarily only come through a negotiation process. The negotiation process is performed using argumentation-based negotiation that uses preferences over the types of correspondences in order to choose the mappings that will be used to finally merge the ontologies (see Section 5.3). The preferences depend on the context and situation. A major feature of this context is the ontology, and the structural features thereof, such as the depth of the subclass hierarchy and branching factor, ratio of properties to concepts, etc. The analysis of the components of the ontology is aligned with the approach to ontology evaluation, demonstrated in [13], and can be formalized in terms of feature metrics. Thus the preferences can be determined on the characteristics of the ontology. For example, we can select a preference for terminological mapping if the ontology is lacking in structure, or prefer extensional mapping if the ontology is rich in instances.

Thus, the alignment/negotiation wrapper interfaces two tools—one for the ontology alignment discovery and one for negotiation of agreed alignment. We call these tools *AKit* and *NKit*, respectively, within this section. For the former, we use the ontology alignment API (see [18]) developed by INRIA Rhone-Alpes[2]. For the negotiation we use the framework described in [30]. Both tools are used by the wrapper in order to produce O_A—an ontology consisting of axioms[3] merging classes, individuals and properties in the O_L and O_M ontologies. It is used in consequent factual merging and refinement in the ontology reasoning and management wrapper (see Section 5.3 for details).

Algorithm 1. Meta-algorithm of the alignment and negotiation

Require: O_L, O_M—ontologies in OWL format
Require: *AKit*, *NKit*—ontology alignment and alignment negotiation tools, respectively
Require: *ALMSET*—a set of the alignment methods to be used
Require: *PREFSET*—a set of alignment formal preferences corresponding to the O_L, O_M ontologies (to be used in N-kit)
1: $S_A \leftarrow \emptyset$
2: **for** $method \in ALMSET$ **do**
3: $S_A \leftarrow S_A \cup AKit.getAlignment(O_L, O_M, method)$
4: **end for**
5: $A_{agreed} \leftarrow NKit.negotiateAlignment(S_A, PREFSET)$
6: $O_A \leftarrow AKit.produceBridgeAxioms(A_{agreed})$
7:**return** O_A

The wrapper itself works according to the meta-code in Algorithm 1. The ontology alignment API offers several possibilities of actual alignment methods, which range from trivial lexical equality detection through more sophisticated string and edit-distance based algorithms to an iterative structural alignment by the OLA algorithm (see [17]). The ontology alignment API has recently been extended by a method for the calculation of a similarity metric between ontology entities, an adaptation of the SRMetric used in [44]. We also consider a set of justifications, that explain why the mappings have been generated. This information forms the basis for the negotiation framework that dynamically generates arguments, supplies the reasons for the mapping choices and negotiates an agreed alignment for both ontologies O_L and O_M.

5.3. Ontology merging wrapper

This wrapper is used for merging of the O_L and O_M ontologies according to the statements in O_A (each of the ontologies technically represented as a respective Jena ontology model). Moreover, the wrapper resolves possible inconsistencies caused by the merging—favouring the assertions in the O_M ontology, which are supposed to be more relevant. The resulting ontology O_I is passed to the ontology diff wrapper to be compared with the former O_M master ontology. The respective addition model forms a basis for the natural language suggestions that are produced as a final product of the integration (see Sections 5.4 and 5.5 for details).

Algorithm 2. Meta-algorithm of the merging and inconsistency resolution

Require: O_L, O_M, O_A—ontologies in OWL format
Require: *merge*()—a function that merges the axioms from input ontologies, possibly implementing reasoning routines according to the ontology model used
Require: C—set of implemented consistency restrictions; each element $r \in C$ can execute two functions $r.detect()$ and $r.resolve()$ that detect (and return) and resolve an inconsistency in the input ontology, respectively
1: $O_I \leftarrow merge(O_M, O_L, O_A)$
2: $inconsistencies \leftarrow \emptyset$
3: **for** $r \in C$ **do**
4: $inconsistencies \leftarrow inconsistencies \cup r.detect(O_I)$
5: $O_I \leftarrow r.resolve(O_I)$
6: **end for**
7: **return** $O_I, inconsistencies$

[2] See http://alignapi.gforge.inria.fr/ for up-to-date information on the API.
[3] Using constructs like *owl:equivalentClass*, *owl:sameAs*, *owl:equivalentProperty*, *rdfs:subClassOf* or *rdfs:subPropertyOf*.

Algorithm 2 describes the meta-code of the process arranged by the ontology merging and reasoning wrapper. We currently employ no reasoning in the *merge*() function. However, sub-class subsumption (as implemented by the Jena framework) is used when detecting and resolving inconsistencies. The inconsistencies are constituted by user-defined restrictions. These restrictions are implemented as extensions of a generic inconsistency detector and resolver in the ontology merging wrapper. Thus we can implement either logical (in terms of Description Logics, see [2]) inconsistencies, or custom-defined inconsistencies (i.e. cyclic definitions) according to requirements of particular practical applications.

The automatic inconsistency resolution itself is somewhat tricky. However, we can apply a sort of "greedy" heuristic, considering the assertions in the master O_M ontology to be more valid. Therefore we can discard axioms from O_L or O_A that are inconsistent with axioms in O_M—we call such axioms *candidate* in the text below. If there are more such axioms, we discard them one by one randomly until the inconsistency is resolved[4]. If all the conflicting axioms originated in O_M, we just report them without resolution.

We currently implement and resolve the following inconsistencies:

- *Sub-class* hierarchy *cycles*: these are resolved by cutting the cycle, i.e. removing a candidate *rdfs:subClassOf* statement;
- *Disjointness-subsumption* conflicts: if classes are said to be disjoint and a sub-class relationship holds between them at the same time, a candidate conflicting assertion is removed;
- *Disjointness-superclass* conflicts: if a class is said to be a sub-class of classes that are disjoint, a candidate conflicting assertion is removed;
- *Disjointness-instantiation* conflicts (specialisation of the above): if an individual is said to be an instance of classes that are disjoint, a candidate conflicting assertion is removed.

The first one is non-logical inconsistency, whereas the remaining free are examples of logical inconsistencies. More on the types and nature of logical (DL) inconsistencies can be found for instance in [21]. Since most logical inconsistencies are introduced by negative constructs like `owl:disjointWith`, `owl:complementOf` or `owl:differentFrom`, we can easily adapt the above techniques related to disjointness in order to support additional inconsistency types.

A transparent and flexible support of arbitrary non-logical consistency constraints is a part of our future work. We plan to implement this feature on the top of user-defined rules (expressing facts like "if X is a male mammal, then it does not have an ovary"). DINO will not include the learned statements that are in a conflict with the respective rule-based constraints into the merged ontology. Certain more subtle issues related to the ontology design (such as possibly unwelcome multiple inheritance) cannot, however, be generally handled even by the rule-based inconsistency resolution, therefore the more sophisticated refinement of the integrated ontology is deliberately left for the user.

Note that each element of the set of inconsistencies returned by Algorithm 2 (besides the integrated ontology itself) is associated with respective simple natural language description. The descriptions are presented for further examinations by human users in the DINO user interface.

[4] This is the currently implemented way, however, we plan to improve the selection of candidate axioms according to confidence ranking produced by the Text2Onto tool—similarly to the technique described in [26]. This is scheduled for the next version of the DINO integration library.

5.4. Ontology diff wrapper

Possible extension of a master ontology O_M by elements contained in the merged and refined ontology O_I naturally corresponds to the differences between them. In particular, the possible extensions are equal to the additions O_I brings into O_M. The additions can be computed in several ways. Ontology diff wrapper in DINO offers a way how to uniformly interface the particular methods of addition computation. No matter which underlying method is employed, a respective Jena ontology model containing the respective additions is returned. Currently, the following methods are implemented within the wrapper:

(1) SemVersion-based diff computation—additions at the RDF (triple) level computed using the SemVersion library (see [43])
(2) addition model computation by set operations on the underlying Jena RDF models
(3) addition model computation by direct iterative querying of the former master ontology model, integrated model and alignment model for reference purposes (see Algorithm 3 for details on implementation)

For the practical experiments with ontologies, we have used the third method—mainly due to the fact that it computes the additions directly at the ontology level and not at the lower triple level (which means subsequent processing load when getting back to the ontology model again).

Algorithm 3. Meta-algorithm of the addition model computation (by direct model querying)

Require: O_M, O_I, O_A—former master, integrated and alignment ontologies, respectively
Require: *copyResource*()—a function that returns a copy of an ontology resource (e.g. class or property) including all relevant features that are bound to it (e.g. subclasses, superclasses, instances for a class or domain and range for a property)
1: $O_{added} \leftarrow \emptyset$
2: **for** $c \in O_I.getNamedOntologyClasses()$ **do**
3: **if not** $O_M.contains(c)$ **or** $O_A.contains(c)$ **then**
4: $O_{added} \leftarrow copyResource(c)$
5: **end if**
6: **end for**
7: **for** $p \in O_I.getOntologyProperties()$ **do**
8: **if not** $O_M.contains(p)$ **or** $O_A.contains(p)$ **then**
9: $O_{added} \leftarrow copyResource(p)$
10: **end if**
11: **end for**
12: **return** O_{added}

Note that the algorithm does not compute all differences between arbitrary ontologies in general. However, this is no drawback for the current implementation of DINO integration. We deal with learned ontology extending the master one. The extensions originating in automatically learned knowledge do not cover the whole range of possible OWL constructs, thus we do not need to tackle e.g. anonymous classes and restrictions in the addition model computation. Therefore the employed custom addition computation can be safely applied without any loss of information. The computed addition ontology model is passed to the suggestion sorter then (see Section 5.5 for details).

Table 1
Scheme of suggestion generation

Axiom pattern	NL suggestion scheme	Example
Class c_1 is related by relation r to class c_2	The class $c_1.label()\ f(r)$ the class $c_2.label()$.	The class "difference_c" is disjoint with the class "inclusion_c".
Individual i is a member of class c	The class $c.label()$ has the $i.label()$ instance.	The class "the_cytoskeleton_organiser_c" has the "centrosome_i" instance.
Property p_1 with features features x is related to property p_2 by relation r	There is a $p_1.label()$ $g(x)$ property. It is $f(r)$ $p_2.label()$.	There is a "contain_r" object property. Its range is the "organ_c" class.
Property p_1 with features x has domain/range class c	There is a $p_1.label()$ $g(x)$ property. Its domain/range is the $c.label()$ class.	There is a "contain_r" object property. It has the "has_part_r" superproperty.

5.5. Sorted suggestions generator

The addition ontology passed to this component forms a base for the eventual extension suggestions for the domain experts. In order to reduce the effort in the final reviewing of the master ontology extensions, we create respective simple natural language suggestions that are associated with corresponding facts in the addition ontology model. The natural language suggestions are then presented to users—when a suggestion is accepted by the users, the associated fact is included into the master ontology model. Table 1 shows a scheme of the natural language (NL) suggestion generation. The r variable represents possible relations between classes or properties (e.g. `rdfs:subClassOf`, `rdfs:subPropertyOf` or `owl:disjointWith`), mapped by the function $f()$ to a respective natural language representation (e.g. *is a sub-class of*, *is a sub-property of* or *is disjoint with*). The x variable represents possible features of a property (e.g. `owl:ObjectProperty` or `owl:FunctionalProperty`, mapped by the function $g()$ to a respective natural language representation (e.g. *object* or *functional*). In general, the number of suggestions originating from the addition ontology model can be quite large, so an ordering that takes a relevance measure of possible suggestions into account is needed. Thus we can for example eliminate suggestions with low relevance level when presenting the final set to the users (without overwhelming them with a large number of possibly irrelevant suggestions). As a possible solution to this task, we have proposed and implemented a method based on string subsumption and a specific distance measure (see [31]). These two measures are used within relevance computation by comparing the lexical labels occurring in a suggestion with respect to two sets S_p, S_n of words, provided by users. The S_p and S_n sets contain preferred and unwanted words respectively, concerning the lexical level of optimal extensions. The suggestions T are sorted according to the respective $rel(T, S_p) - rel(T, S_n)$ values, where $rel(T, S)$ is a function measuring the relevance of the suggestion triple T with respect to the words in the set S. The higher the value, the more relevant the suggestion triple is. We develop the relevance function in detail in Algorithm 4.

Algorithm 4. The relevance function

Require: S_t—a set of (possibly multiword) lexical terms occurring in the suggestion
Require: S—set of words
Require: $\rho \in (0, 1)$ influences the absolute value of relevance measure
Require: t—integer constant; maximal allowed distance
Require: $levDist(s_1, s_2)$—Lev. distance implementation
for $elem \in S_t$ **do**
  $R_{elem} \leftarrow 0$
end for
for $elem \in S_t$ **do**
  if $elem$ is a substring of or equals to any word in S or vice versa **then**
    $R_{elem} \leftarrow 1$
  else
    $d \leftarrow \infty$
    for $v \in S$
      if $levDist(elem, v) < d$ **then**
        $d \leftarrow levDist(elem, v)$
      end if
    end for
    if $d \leqslant t$ **then**
      $R_{elem} \leftarrow (1 - \frac{d}{t+1})$
    else if $elem$ is a multiword term **then**
      $L \leftarrow$ set of single terms in the $elem$ label expression
      $EXP \leftarrow 0$
      for $u \in L$ **do**
        if u is a substring of or equals to any word in S or vice versa **then**
          $EXP \leftarrow EXP + 1$
        else
          $d \leftarrow \infty$
          for $v \in S$ **do**
            if $levDist(u, v) < d$ **then**
              $d \leftarrow levDist(u, v)$
            end if
          end for
          if $d \leqslant t$ **then**
            $EXP \leftarrow EXP + (1 - \frac{d}{t+1})$
          end if
        end if
      end for
      if $EXP = 0$ **then**
        $R_{elem} \leftarrow 0$
      else
        $R_{elem} \leftarrow \rho^{\frac{1}{EXP}}$
      end if
    end if
  end if
end for
return $\frac{\sum_{elem \in S_t} R_{elem}}{|S_t|}$

The function naturally measures the "closeness" of the labels occurring in the suggestion to the set of terms in S. The value of 1 is achieved when the label is a direct substring of or equal to any word in S or vice versa. When the Levenshtein distance between the label and a word in S is lower than or equal to the defined threshold t, the relevance decreases from 1 by a value proportional to the fraction of the distance and t. If this is not the case (i.e. the label's distance is greater than t for each word in S), a similar principle is applied for possible word-parts of the label and the relevance is further proportionally decreased (the minimal possible value being 0).

Note that the complexity of the sorting itself mostly contributes to the overall complexity of the relevance-based sorting of suggestions. As can be found out from Algorithm 4, the complexity is in $O(cmnl^2 + m \log m)$ (c—maximal number of terms occurring in a suggestion, thus a constant; m—number of suggestions; n—number of words in the preference sets; l—maximal length of a word in suggestion terms, basically a constant), which gives $O(m(n + \log m))$. As the size of the sets of user preferences can be practically treated as constant, we obtain the $O(m \log m)$ complexity class with respect to the number of suggestions, which is feasible.

5.6. Natural language generation (NLG) component

The DINO framework is supposed to be used primarily by users who are not experts in ontology engineering. Therefore the suggestions are produced in a form of very simple natural language statements, as seen in the previous section. Moreover, we automatically create a natural language representation of the whole addition model, interfacing the framework described in [42]. This is meant to further support laymen users by readable representation of the whole addition model in order to give them an overall impression of the changes.

The single suggestions are still bound to the underlying statement in the addition ontology model. Therefore a user can very easily add the appropriate OWL axioms into the new version of the O_M master ontology without actually dealing with the intricate OWL syntax itself. Concrete examples of both suggestions and continuous natural language representation of the addition model are given in Section 6.

6. Example application and results of DINO integration

We applied the integration technique described in Section 5 in the context of data typical for biomedical research. However, the way of exploiting the DINO integration technique reported in this section is rather general, since it aims at cost-efficient extension or population of a master ontology by knowledge learned from empirical data. Thus, a similar deployment of the integration can actually help to tackle needs of many other possible use cases.

Real world data for the master ontology and ontology learning sources were used. More specifically, we employed resources from CO-ODE biomedicine ontology fragment repository[5] and data from relevant Wikipedia topics, respectively.

Rigorous evaluation of the whole process of integration is a complex task involving lot of open problems as its sub-problems (for instance, there is no standard ontology evaluation process applicable in general—see [27,13]). Moreover, there is an emphasis on the human-readable and laymen oriented form of the integration process results. This dimension forms a primary axis of the evaluation, however, its realisation involves logistically demanding participation of a broader (biomedicine) expert community.

Accomplishing the above tasks properly is a part of our future work. Nonetheless, there are several aspects that can be assessed and reported even without devising an optimal ontology evaluation method (which may be impossible anyway) and/or getting involved large representative sample of domain experts:

- features of the learned ontology (e.g. size or complexity)
- mappings established by alignment
- basic assessment of the quality and correctness of suggestions and their sorting according to defined preferences

These factors of integration are analysed and discussed within an experimental application described in Section 6.1.

The negotiation component has recently been evaluated separately as a stand-alone module, using the Ontology Alignment Evaluation Initiative test suite[6] and experiments on the impact that the argumentation approach has over a set of mappings. A comparison wrt. current alignment tools is presented in [29]. The preliminary results of these experiments are promising and suggest that the argumentation approach can be beneficial and an effective solution to the problem of dynamically aligning heterogeneous ontologies. This justifies also the application of the implemented technique in the ontology integration task.

6.1. Experimental integration of biomedical research knowledge—extension of (blood) cells ontology fragment

In order to show the basic features of our novel integration technique in practice, we tested the implementation using knowledge resources from biomedicine domain[7]. In particular, we combined fragments of GO cellular component description and

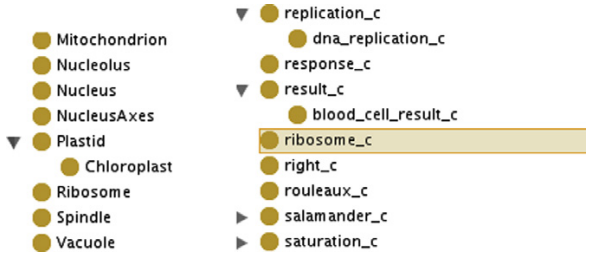


Fig. 3. Sample from master and learned ontology.

eukaryotic cell description[8] to form the master ontology. In the example scenario, we wanted to extend this master ontology using content of Wikipedia entries on `Cells_(biology)` and `Red_blood_cell`. These resources were passed to the ontology learning DINO component and respective ontology was learned. Both master and learned ontology samples are displayed in Fig. 3 (on the left-hand and right-hand side, respectively). Note that these master and learned ontologies correspond to the O_M, O_L ontologies displayed in Fig. 2, Section 5. The names in learned ontology have specific suffixes (i.e. "_c"). This is due to naming conventions of the ontology learning algorithm we use. We keep the suffixes in suggestions, since they help to easily discriminate what comes from empirical data and what from the master ontology. However, we filter them out when generating the text representing the whole extension model (see below for examples).

Table 2 compares metric properties of the master and learned ontologies, as computed by the Protégé tool. The particular metrics are expanded as follows: M_1—number of named classes (all/primitive/defined); M_2—number of parents per class (mean/median/maximum); M_3—number of siblings per class (mean/median/maximum); M_4—number of anonymous classes (restrictions); M_5—number of properties (all/object/datatype); M_6—Description Logics expressivity.

The learned ontology has higher ratio of primitive classes, moreover, it contains no restriction on class definitions. There are some simple object properties with both domains and ranges defined. Its DL expressivity allows concept intersection, full universal and existential quantification, atomic and complex negation and datatypes. The expressivity of the master ontology does not involve datatypes, however, it contains numeric restrictions. Summing up, the master ontology contains several complicated constructs not present in the learned ontology, however, the ontology learned only from two simple and relatively small resources is much larger. When computing the negotiated alignment (the O_A ontology as given in Fig. 2, Section 5) between master and learned ontology, 207 mappings were produced and among them, 16 were accepted. A sample from the alignment ontology is displayed in Fig. 4.

Merging of the learned and master ontologies according to the computed alignments results in several inconsistencies—the report generated by DINO is displayed in Fig. 5. Two of these three inconsistencies are resolved correctly (according to human intuition) by the algorithm, forming an integrated ontology O_I, as displayed in Fig. 2, Section 5.

After resolving the inconsistencies (three inconsistencies per an integrated resource were resolved in average within our experiment) and generating the addition model, natural language

[5] See http://www.co-ode.org/ontologies.

[6] See http://oaei.ontologymatching.org/.

[7] Should the reader be interested, all relevant resources used and/or created during the described experiment are available at http://smile.deri.ie/resources/2007/08/31/dino_exp_data.zip

[8] Samples downloaded from the CO-ODE repository, see http://www.co-ode.org/ontologies/bio-tutorial/sources/GO_CELLULAR_COMPONENT_EXTRACT.owl and http://www.co-ode.org/ontologies/eukariotic/2005/06/01/eukariotic.owl, respectively.

Table 2
Metrics of master and learned ontologies

Metric/ontology	M_1	M_2	M_3	M_4	M_5	M_6
Learned	391/379/12	3/1/5	7/1/16	0	13/13/0	$\mathcal{ALC}(D)$
Master	40/36/4	2/1/2	5/1/15	16 (restr.)	1/1/0	$\mathcal{ALCN}$

```
<owl:Class rdf:about="#Chromosome">
  <owl:equivalentClass rdf:resource="#chromosome_c"/>
</owl:Class>

<owl:Class rdf:about="#Chloroplast">
  <owl:equivalentClass rdf:resource="#chloroplast_c"/>
</owl:Class>

<owl:Class rdf:about="#Ribosome">
  <owl:equivalentClass rdf:resource="#ribosome_c"/>
</owl:Class>

<owl:Class rdf:about="#Ribosome">
  <owl:equivalentClass rdf:resource="#the_ribosome_c"/>
</owl:Class>

<owl:Class rdf:about="#Nucleus">
  <owl:equivalentClass rdf:resource="#nucleus_c"/>
</owl:Class>

<owl:Class rdf:about="#Mitochondrion">
  <owl:equivalentClass rdf:resource="#mitochondrium_c"/>
</owl:Class>
```

Fig. 4. Sample alignment.

suggestions (associated with respective OWL axioms) are produced. Sample suggestions associated with respective relevance measures are displayed in Fig. 6. A portion of the continuous text generated by the NLG component that is corresponding to the addition model is displayed in Fig. 7. Similar "pretty" texts are to be presented to users in the extended DINO interface (the current interface offers only raw text, however, necessary parsing, filtering and highlighting of the ontology terms is under construction). It provides users with additional source of lookup when deciding which suggestions to accept into the next version of the master ontology.

The suggestions are the ultimate output of the integration algorithm. Their main purpose is to facilitate laymen effort in incorporation of new knowledge from unstructured resources into an ontology. Therefore we performed basic evaluation of several parameters that influence actual applicability of the suggestions. We ran the integration algorithm on the same data with four different suggestion-preference sets, simulating four generic trends in the preference definition:

- specification of rather small number of preferred terms, no unwanted terms
- specification of rather small number of preferred and unwanted terms
- specification of larger number of preferred terms, no unwanted terms
- specification of larger number of preferred and unwanted terms

Table 3 gives an overview of the four iterations, the particular preferred and unwanted terms and distribution of suggestions into relevance classes. The terms were set by a human user arbitrarily, reflecting general interest in clinical aspects of the experimental domain knowledge. The terms in preference sets reflect possible topics to be covered by the automatic extension of the current ontology. S_+, S_0 and S_- are classes of suggestions with relevance greater, equal and lower than zero, respectively ($S = S_+ \cup S_0 \cup S_-$).

For each of the relevance classes induced by one iteration, we randomly selected 20 suggestions and computed two values on this sample:

- $P_x, x \in \{+, 0, -\}$—ratio of suggestions correctly placed by the sorting algorithm into an order defined by a human user for the same set (according to the interest defined by the particular preferences)
- $A_x, x \in \{+, 0, -\}$—ratio of suggestions that are considered appropriate by a human user according to his or her knowledge of the domain (among all the suggestions in the sample)

The results are summed up in Table 4. More details on interpretation of all the experimental findings are given in consequent Section 6.2.

6.2. Discussion of the experiment results

The DINO integration library allows users to submit the resources containing knowledge they would like to reflect in their current ontology. The only thing that is needed is to specify preferences on the knowledge to be included using the sets of preferred and unwanted terms. After this, sorted suggestions on possible ontology extensions (after resolution or reporting of possible inconsistencies) can be produced and processed in minutes, whereas the purely manual development and integration of respective ontology would take hours even for relatively simple natural language resources. Moreover, it would require a certain

```
Inconsistency:
The following classes are disjoint and in mutual sub-class relationship at the same time:
"organelle_c" and "nucleus_c"

Inconsistency:
The following classes are disjoint and in mutual sub-class relationship at the same time:
"cell_c" and "blood_cell_c"

Inconsistency:
The following classes are disjoint and in mutual sub-class relationship at the same time:
"cell_wall_c" and "membrane_c"
```

Fig. 5. Report on inconsistencies.

```
...

-----
Relevance: 0.75
Suggestion   : The class "cell_nucleus_c" is disjoint with the class "compartment_c".
-----
Relevance: 0.083333336
Suggestion   : The class "Nucleus" is equivalent to the class "nucleus_c"
-----
Relevance: 0.0
Suggestion   : The class "organelle_c" has the "mitochondrium_c" subclass.
-----
Relevance: 0.0
Suggestion   : The class "Mitochondrion" is equivalent to the class "mitochondrium_c".
-----
Relevance: -0.8333333
Suggestion   : The class "chromosome_c" has the "Organelle" superclass.
-----
Relevance: -0.9166666
Suggestion   : The class "Chromosome" is equivalent to the class "chromosome_c".
-----

...
```

Fig. 6. Sample suggestions.

```
...
There are "Cells", "Nucleuss", "bacteriums", and "genetic diseases".
There are "red blood cells", "absorptions", "additional functions", "advantages", and "archaeons".
There are "autoimmunediseases", "aplasiums", "appendages", "areas", and "atoms".
There are "bacterias", "bacteriums", "beacons", "bilayers", and "blockages".
There are "cannots", "capacitys", "capsules", "cells", and "changes".
There are "chloroplasts", "chromosomals", "ciliums", "coagulations", and "comparisons".
...
```

Fig. 7. Sample from the generated continuous text.

Table 3
Iterations—the preference sets and sizes of the resulting suggestion classes

Iteration	Preferred	Unwanted	$\lvert S_+ \rvert$	$\lvert S_0 \rvert$	$\lvert S_- \rvert$	$\lvert S \rvert$
I_1	cell; autoimmune disease; transport; drug; gene; DNA	$\emptyset$	310	429	0	739
I_2	cell; autoimmune disease; transport; drug; gene; DNA	bacteria; prokaryotic; organelle; wall; chromosome; creation	250	344	145	739
I_3	cell; autoimmune disease; transport; drug; gene; DNA eukaryotic; organ; function; part; protein; disease; treatment; cell part immunosuppression; production	$\emptyset$	485	254	0	739
I_4	cell; autoimmune disease; transport; drug; gene; DNA eukaryotic; organ; function; part; protein; disease; treatment; cell part immunosuppression; production	bilayer; bacteria; prokaryotic; additional function; organelle; macromollecule; archaeon; vessel; wall; volume; body; cell nucleus; chromosome; erythrocyte; creation	314	292	133	739

Table 4
Evaluation of random suggestion samples per class

Iteration	P_+	A_+	P_0	A_0	P_-	A_-
I_1	0.45	0.75	0.90	0.60	—	—
I_2	0.45	0.75	1.00	0.80	0.60	0.70
I_3	0.70	0.80	0.95	0.75	—	—
I_4	0.55	0.75	0.70	0.85	0.50	0.85

experience with knowledge engineering, which is uncommon among biomedicine domain experts.

In Section 6.1 we described the application of our integration technique to an extension of biomedical research ontology fragment. The analysed results show that the suggestions produced are mostly correct (even though rather simple and sometimes obvious) with respect to the domain in question, ranging from 50% to 85% among the algorithm iterations. The relevance-based sorting according to preferences is more appropriate in case of irrelevant (zero relevance) suggestions, ranging from 70% to 100% of correctly placed suggestions. Its precision in case of suggestions with positive and negative relevance is lower, ranging from 45% to 70%. More terms in the preference sets cause better sorting performance (the ratio of appropriate suggestions being independent on this fact). Thus, the best discrimination in terms of presenting the most relevant suggestions first is achieved for larger preference sets. However, even the discrimination for smaller sets is fair enough (as seen in Table 3 in the previous section).

The automatically produced natural language suggestions can be very easily browsed and assessed by users who are not familiar with ontology engineering at all. Since the respective axioms are associated to the suggestions, their inclusion into another version of the master ontology is pretty straightforward once a suggestion is followed by a user. The DINO integration technique still needs to be evaluated with a broader domain expert audience involved, however, even the preliminary results presented here are very promising in the scope of the requirements specified in Section 1.

7. Notes on realistic DINO deployment

The EU IST 6th Framework project RIDE has identified and analysed several biomedical use case areas in [15] relevant concerning deployment of the Semantic Web technologies

(i.e., ontologies and related querying, knowledge and data management tools). The scope of [15] is rather broad, however, we can track few specific areas with significant needs that can be covered by the DINO ontology lifecycle and integration framework (Section 7.1). Section 7.2 discusses preliminary feedback of our potential users and consequently suggests most appropriate modes of the DINO prototype exploitation.

7.1. Selected use case areas

7.1.1. Longitudinal electronic health record

The main topic here is development of standards and platforms supporting creation and management of long-term electronic health records of particular patients. These records should be able to integrate various sources of data coming from different medical institutions a patient may have been treated in during his whole life. Quite obviously, one needs to integrate different data sources, present very often in unstructured natural language form. Ontologies extracted from the respective patient data resources can very naturally support their integration into longitudinal electronic health records by means of DINO.

7.1.2. Epidemiological registries

Epidemiology analyses diseases, their reasons, statistical origins and their relation to a selected population sample's socio-economic characteristic. Epidemiological registries should be able to reasonably store and manage data related to population samples and their medical attributes in order to support efficient processing of the respective knowledge by the experts. In this use case area, one has to integrate knowledge from electronic health records in order to create population-wise repositories. Once the ontology-enabled electronic health records are created (DINO can help here as mentioned above), one can integrate them within another version of an "epidemiology" ontology (again, by means of the DINO framework). The resulting model can be employed in order to perform symbolic analysis (using ontology-based symbolic querying and logical inference) of the registry data, complementing the statistical numeric analysis methods.

7.1.3. Public health surveillance

Public health surveillance presents ongoing collection, analysis, interpretation and dissemination of health-related data in order to facilitate a public health action reducing mortality and/or improving health. The use case area puts an emphasis on efficient dynamic processing of new data that are mostly in the free natural language text form, which can be directly facilitated by the DINO integration of respective learned ontologies. Ontologies created from and extended by urgent dynamic data can efficiently support expert decisions in risk management tasks. Continuous integration of less urgent data from various sources (either texts or ontologies) can support studies on public health issues in the long term perspective then.

7.1.4. Management of clinical trials

Clinical trials are studies of the effects of newly developed drugs on real patient samples. They are essential part of approval of new drugs for normal clinical use and present an important bridge between medical research and practice. Efficient electronic data representation and querying is crucial here. However, even if the data are electronically represented, problems with their heterogeneity and integration occur as there are typically several different institutions involved in a single trial. The presented integration method can help in coping with the data heterogeneity here, especially when some of the data is present in the natural language form.

7.2. Preliminary user feedback and lessons learned

We presented a DINO demo and/or sample knowledge integration results to biomedical domain and ontology engineering experts[9]. We also discussed a sketch of the DINO application in the above practical use cases with them. Their preliminary feedback can be summarised into the following three points: (1) the framework was considered as a helpful complement to the traditional manual ontology development environments (such as Protégé); (2) the results were found promising concerning the scalable ontology extension by the knowledge in unstructured domain resources, however, certain refinement by ontology engineers was generally considered as a must in order to maintain high quality of the respective master biomedical ontologies; (3) the natural language presentation of the sorted extension suggestions was found to be very useful for the domain experts with no ontology engineering background. The last finding has been further supported by the recent evaluation of the natural language generation framework we use in DINO (see [11] for details).

Following the discussion with the domain and ontology engineering experts, we can distinguish between two practical and reliable DINO application modes with different requirements on the expert user involvement:

- *Instance-only integration:* ontology learned from the textual resources is semi-automatically integrated into a master ontology, taking only instance-related assertions into account, i.e., the upper ontology is populated with new instances of the present concepts and with relations among the instances. Such an application does not require any extensive expert involvement of ontology engineers, since the instance-related suggestions produced by DINO are relatively reasonable according to our discussions with domain experts. Severe modelling errors can only be introduced very rarely, therefore only the expert knowledge of the domain is generally enough to decide which DINO suggestions to follow in the master ontology extension.
- *Full-fledged integration:* an unrestricted processing of the DINO suggestions, i.e., taking also the class-related assertions into account, requires more careful expert involvement in order to guarantee high quality of the integration results. Ontology experts are generally still needed when resolving possible modelling bugs (such as multiple class inheritance or redundant disjointness relations) that might be insufficiently tackled by the domain experts when processing the natural language DINO suggestions. State of the art methodologies such as ontology "re-engineering" as introduced in [3]can help when applying DINO this way.

8. Summary and future work

We have presented the basic principles of DINO—a novel lifecycle scenario and framework for ontology integration and maintenance in dynamic and data-intensive domains like medicine. As a core contribution of the paper, we have described the mechanism of integration of automatically learned and manually maintained medical knowledge. The presented method covers all the requirements specified in Section 1. The proposed combination of automatic and manual knowledge acquisition principles, integration and inconsistency resolution ensures more scalable production and extension of ontol-

[9] These were namely researchers from the REMEDI institute, see http://www.nui-galway.ie/remedi/, Prof. Werner Ceusters, M.D. (director of the Ontology Research Group of the New York State Center of Excellence in Bioinformatics and Life Sciences) and ontology engineers from the Knowledge Engineering Group at the University of Economics in Prague.

ogies in dynamic domains. We presented and analysed results of a preliminary practical application of the DINO integration technique in Section 6. Section 7 outlined possible applications in realistic use case areas that have been recently identified in the biomedicine and e-health fields. The section also summarised preliminary feedback of our potential users. Based on the feedback analysis, two practical DINO application modes were suggested. Note that we have also delivered prototype implementations of a DINO API library and a respective GUI interface (research prototypes of the respective software can be downloaded at http://smile.deri.ie/tools/dino).

Since the primary funding project has finished, we are in the process (as of 2008) of securing another funding that could support further improvements of DINO. These improvements consist mainly of an extended support for inconsistency resolution, integration with state of the art ontology editors (primarily Protégé) and extension of the DINO user interface (e.g., providing explicit support for the two application modes given in Section 7.2).

Moreover, we have recently started to work on another project with motivations similar to DINO, however, with much more ambitious goals. [35] presents a preliminary proposal and results of a novel empirical knowledge representation and reasoning framework. One of the principal applications of the researched framework is a complex empirical inference-based integration of arbitrary emergent knowledge (e.g., learned ontologies) with precise manually designed knowledge bases. We plan to combine the ontology integration powered by the reasoning described in [35] with the results achieved within the DINO implementation in order to allow for more efficient, scalable, user-friendly and robust dynamic maintenance of (partially emergent) ontologies. Last but not least, we are going to continuously evaluate the resulting framework among broader biomedicine expert communities and improve it in line with demands of interested industry partners (possibly, but not only within the presented real-world application domains).

Acknowledgments

The article is a significantly extended version of the paper: Vít Nováček, Loredana Laera, Siegfried Handschuh. *Dynamic Integration of Medical Ontologies in Large Scale*. In: Proceedings of the WWW2007/HCLSDI workshop. ACM Press, 2007. (see http://www2007.org/workshops/paper_141.pdf). The presented work has been kindly supported by the EU IST 6th framework's Network of Excellence 'Knowledge Web' (FP6-507482), by the 'Líon' project funded by Science Foundation Ireland under Grant No. SFI/02/CE1/I131 and partially by Academy of Sciences of the Czech Republic, 'Information Society' national research program, the Grant number AV 1ET100300419. Moreover, we greatly appreciated the anonymous reviewers' remarks that resulted in significant improvements of the submitted text.

References

[1] Alasoud A, Haarslev V, Shiri N. A hybrid approach for ontology integration. In: Proceedings of the 31st VLDB conference. Very large data base endowment, 2005.
[2] Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF. The decription logic handbook: theory, implementation, and applications. Cambridge, USA: Cambridge University Press; 2003.
[3] Bechhofer S, Gangemi A, Guarino N, van Harmelen F, Horrocks I, Klein M, Masolo C, Oberle D, Staab S, Stuckenschmidt H, Volz R. Tackling the ontology acquisition bottleneck: an experiment in ontology re-engineering, 2003. Retrieved at: http://citeseer.ist.psu.edu/bechhofer03tackling.html, Apr 3 2008.
[4] Bechhofer S, van Harmelen F, Hendler J, Horrocks I, McGuinness DL, Patel-Schneider PF, Stein LA. OWL Web Ontology Language Reference, 2004. Available from (February 2006): http://www.w3.org/TR/owl-ref/.
[5] Berners-Lee T, Hendler J, Lassila O. The semantic web. Scientific American, 5, 2001.
[6] Brickley D, Guha RV. RDF Vocabulary Description Language 1.0: RDF Schema, 2004. Available from (February 2006): http://www.w3.org/TR/rdf-schema/.
[7] Calvanese D, Giacomo GD, Lenzerini M. A framework for ontology integration. In: Proceedings of the first semantic web working symposium. Springer-Verlag, 2001.
[8] Cimiano P, Völker J. Text2Onto—a framework for ontology learning and data-driven change discovery. In: Proceedings of the NLDB 2005 conference. Springer-Verlag; 2005. p. 227–38.
[9] Corcho O, Lopez-Cima A, Gomez-Perez A. The ODESeW 2.0 semantic web application framework. In: Proceedings of WWW 2006. New York: ACM Press; 2006. p. 1049–50.
[10] Brewster FCC, Wilks Y. User-centred onlology learning for knowledge management. In: Proceedings seventh international workshop on applications of natural language to information systems, Stockholm. 2002.
[11] Davis B, Iqbal AA, Funk A, Tablan V, Bontcheva K, Cunningham H, Handschuh S. Roundtrip ontology authoring. In: Proceedings of ISWC 2008. Springer-Verlag; 2008.
[12] Deen SM, Ponnamperuma K. Dynamic ontology integration in a multi-agent environment. In: Proceedings of AINA '06. IEEE computer society. 2006.
[13] Dellschaft K, Staab S. On how to perform a gold standard based evaluation of ontology learning. In: Proceedings of the international semantic web conference. Athens, GA, USA. 2006.
[14] Dieng-Kuntz R, Minier D, Ruzicka M, Corby F, Corby O, Alamarguy L. Building and using a medical ontology for knowledge management and cooperative work in a health care network. Comput Biol Med 2006;36:871–92.
[15] Eichelberg M. Requirements analysis for the ride roadmap. Deliverable D2.1.1, RIDE. 2006.
[16] Euzenat J, Bach TL, Barrasa J, Bouquet P, Bo JD, Dieng R et al. D2.2.3: state of the art on ontology alignment. Technical report, Knowledge Web. 2004.
[17] Euzenat J, Loup D, Touzani M, Valtchev P. Ontology alignment with ola. In: Proceedings of the third international workshop on evaluation of ontology based tools (EON), Hiroshima, Japan, 2004. CEUR-WS.
[18] Euzenat J. An API for ontology alignment. In: ISWC 2004: third international semantic web conference. Proceedings. Springer-Verlag; 2004. p. 698–12.
[19] Fernandez-Lopez M, Gomez-Perez A, Juristo N. Methontology: from ontological art towards ontological engineering. In: Proceedings of the AAAI97 spring symposium series on ontological engineering. Stanford, USA, March 1997. p. 33–40.
[20] Fernandez-Lopez M, Gomez-Perez A, Rojas MD. Ontologies' crossed life cycles. In: Proceedings of international conference in knowledge engineering and management. Springer-Verlag; 2000. p. 65–79.
[21] Flouris G, Huang Z, Pan JZ, Plexousakis D, Wache H. Inconsistencies, negations and changes in ontologies. In: Proceedings of AAAI 2006. AAAI Press; 2006.
[22] Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubezy M, Eriksson H, et al. The evolution of Protégé: an environment for knowledge-based systems development. International Journal of Human—Computer Studies 2003;58(1):89–123.
[23] Gomez-Perez A, Fernandez-Lopez M, Corcho O. Ontological engineering. Advanced information and knowledge processing. Springer-Verlag; 2004.
[24] Gruber TR. Towards principles for the design of ontologies used for knowledge sharing. In: Guarino N, Poli R, editors. Formal ontology in conceptual analysis and knowledge representation. Deventer, The Netherlands: Kluwer Academic Publishers; 1993.
[25] Haase P, Sure Y. State-of-the-art on ontology evolution. Deliverable 3.1.1.b, SEKT. 2004.
[26] Haase P, Völker J. Ontology learning and reasoning—dealing with uncertainty and inconsistency. In: Proceedings of the URSW2005 workshop. NOV 2005; p. 45–55.
[27] Hartmann J, Spyns P, Giboin A, Maynard D, Cuel R, Suarez-Figueroa MC, et al. Methods for ontology evaluation (D1.2.3). Deliverable 123, Knowledge Web. 2005.
[28] Heflin J, Hendler J. Dynamic ontologies on the web. In: Proceedings of AAAI 2000. AAAI Press; 2000.
[29] Laera L, Blacoe I, Tamma V, Payne T, Euzenat J, Bench-Capon T. Argumentation over ontology correspondences in MAS. In: Proceedings of the sixth international joint conference on autonomous agents and multi-agent systems (AAMAS 2007). New York, NY, USA: ACM Press; 2007.
[30] Laera L, Tamma V, Euzenat J, Bench-Capon T, Payne TR. Reaching agreement over ontology alignments. In: Proceedings of fifth international semantic web conference (ISWC 2006). Springer-Verlag; 2006.
[31] Levenshtein VI. Binary codes capable of correcting deletions, insertions and reversals. Cybern Control Theory 1966;10:707–10.
[32] Maedche A, Staab S. Learning ontologies for the semantic web. In: Semantic web workshop 2001. 2001.
[33] Maedche A, Staab S. Ontology learning. In: Staab S, Studer R, editors. Handbook on ontologies. Springer-Verlag; 2004. p. 173–90. chapter 9.
[34] Nováček V, Handschuh S, Laera L, Maynard D, Völkel M, Groza T, et al. Report and prototype of dynamics in the ontology lifecycle (D2.3.8v1). Deliverable 238v1, Knowledge Web. 2006.
[35] Nováček V. Complex inference for emergent knowledge. Technical Report DERI-TR-2008-04-18, DERI, NUIG, 2008. Available from: http://smile.deri.ie/resources/2008/vit/pubs/aerTR0408.pdf.
[36] Noy NF, Klein M. Ontology evolution: not the same as schema evolution. Knowledge Inf Syst 2004:428–40.
[37] Noy N, Musen M. The prompt suite: interactive tools for ontology merging and mapping. 2002.
[38] Pinto HS, Martins JP. A methodology for ontology integration. In: Proceedings of K-CAP'01. 2001.

[39] Staab S, Studer R, editors. Handbook on ontologies. International handbooks on information systems. Springer-Verlag; 2004.
[40] Stojanovic L. Methods and tools for ontology evolution. PhD thesis, University of Karlsruhe, 2004.
[41] Sure Y, Erdmann M, Angele J, Staab S, Studer R, Wenke D. OntoEdit: collaborative ontology development for the Semantic Web. In: First international Semantic Web conference (ISWC2002), Sardinia, Springer; 2002.
[42] Tablan V, Polajnar T, Cunningham H, Bontcheva K. User-friendly ontology authoring using a controlled language. In: Proceedings of LREC 2006—fifth international conference on language resources and evaluation. ELRA/ELDA Paris, 2006.
[43] Völkel M, Groza T. SemVersion: RDF-based ontology versioning system. In: Proceedings of the IADIS international conference WWW/Internet 2006 (ICWI 2006), 2006.
[44] Tamma BLSV, Blacoe I, Wooldridge M. Introducing autonomic behaviour in semantic web agents. In: Proceedings of the fourth international Semantic Web conference (ISWC 2005), Galway, Ireland, November, 2005.

Chapter 6

Semantic Literature Search

Web Semantics: Science, Services and Agents on the World Wide Web 8 (2010) 176–181

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Invited Paper

CORAAL—Dive into publications, bathe in the knowledge

Vít Nováček*, Tudor Groza, Siegfried Handschuh, Stefan Decker

Digital Enterprise Research Institute, National University of Ireland Galway, IDA Business Park, Dangan, Galway, Ireland

ARTICLE INFO

Article history:
Received 26 May 2009
Received in revised form 9 October 2009
Accepted 26 March 2010
Available online 24 April 2010

Keywords:
Knowledge acquisition
Knowledge integration
Life sciences
Knowledge-based publication search

ABSTRACT

Search engines used in contemporary online scientific publishing mostly exploit raw publication data (bags of words) and shallow metadata (authors, key words, citations, etc.). Exploitation of the knowledge contained implicitly in published texts is still largely not utilized. Following our long-term ambition to take advantage of such knowledge, we have implemented CORAAL (*COntent extended by emeRgent and Asserted Annotations of Linked publication data*), an enhanced-search prototype and the second-prize winner of the Elsevier Grand Challenge. CORAAL extracts asserted publication metadata together with the knowledge implicitly present in the relevant text, integrates the emergent content, and displays it using a multiple-perspective search&browse interface. This way we enable semantic querying for individual publications, and convenient exploration of the knowledge contained within them. In other words, recalling the metaphor in the article title, we let the users dive into publications more easily, and allow them to freely bathe in the related unlocked knowledge.

1. Introduction

Online scientific publishing makes knowledge production and dissemination much more efficient than before. The publication process is faster, since the essential phases like authoring, submission, reviewing, and final typesetting are largely computerised. Moreover, the published content is easily disseminated to global audiences via the Internet. In effect, more and more knowledge is being made available.

However, is this growing body of knowledge also easily accessible? We believe the answer is negative, since the rapid growth of the number of available resources is making it harder to identify any particular desired piece of knowledge using current solutions. For instance, Medline, a comprehensive source of life sciences and biomedical bibliographic information (cf. http://medline.cos.com/) currently hosts over 18 million resources. It has a growth rate of 0.5 million items per year, which represents around 1300 new resources per day [9]. Using the current publication search engines,[1] one can explore the vast and ever-growing article repositories using relevant keywords. But this is very often not enough. Imagine for instance a junior researcher compiling a survey on various types of leukemia. The researcher wants to state and motivate in the survey that *acute granulocytic leukemia* is different from *T-cell leukemia*. Although such a statement might be obvious for a life scientist, one should support it in the survey by a citation of a published paper. Our researcher may be a bit inexperienced in oncology and may not know the proper reference straightaway. Using, e.g., the PubMed search service, it is easy to find articles that contain both leukemia names. Unfortunately, there are more than 500 such results. It is tedious or even impossible to go through them all to discover one that actually supports that *acute granulocytic leukemia* is different from *T-cell leukemia*.

Given the wealth of knowledge in life science publications and the limitations of current search engines, it is often necessary to manually scan a lot of possibly irrelevant content. To overcome the problem that anything more expressive than (Boolean combinations of) mere keywords is virtually impossible today, it is necessary to develop technologies that can operate at an enhanced level, using more-expressive concepts and their various relationships. This requires collecting, extracting, and interrelating the knowledge scattered across the large numbers of available life science publications. Unfortunately, manual creation of the necessary information is not really possible at large scale, and automated extraction produces noisy and sparse results [2].

We believe that a few essential elements will enable more knowledge-based search in scientific publications: (i) extraction of publication annotations asserted by people (e.g., author names, titles, references or text structure). (ii) Extraction of knowledge implicitly present in publications (e.g., statements encoding typed relations between particular concepts, or structured representations of arguments made by authors in the text).

* Corresponding author. Tel.: +353 91 495738.
E-mail addresses: vit.novacek@deri.org (V. Nováček), tudor.groza@deri.org (T. Groza), siegfried.handschuh@deri.org (S. Handschuh), stefan.decker@deri.org (S. Decker).

[1] For example, ScienceDirect, Elsevier's front-end to their journals (cf. http://www.sciencedirect.com/), or PubMed, a search service covering bibliographic entries from Medline and many additional life science journals together with links to article full texts (cf. http://www.ncbi.nlm.nih.gov/pubmed/).

1570-8268/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2010.03.008

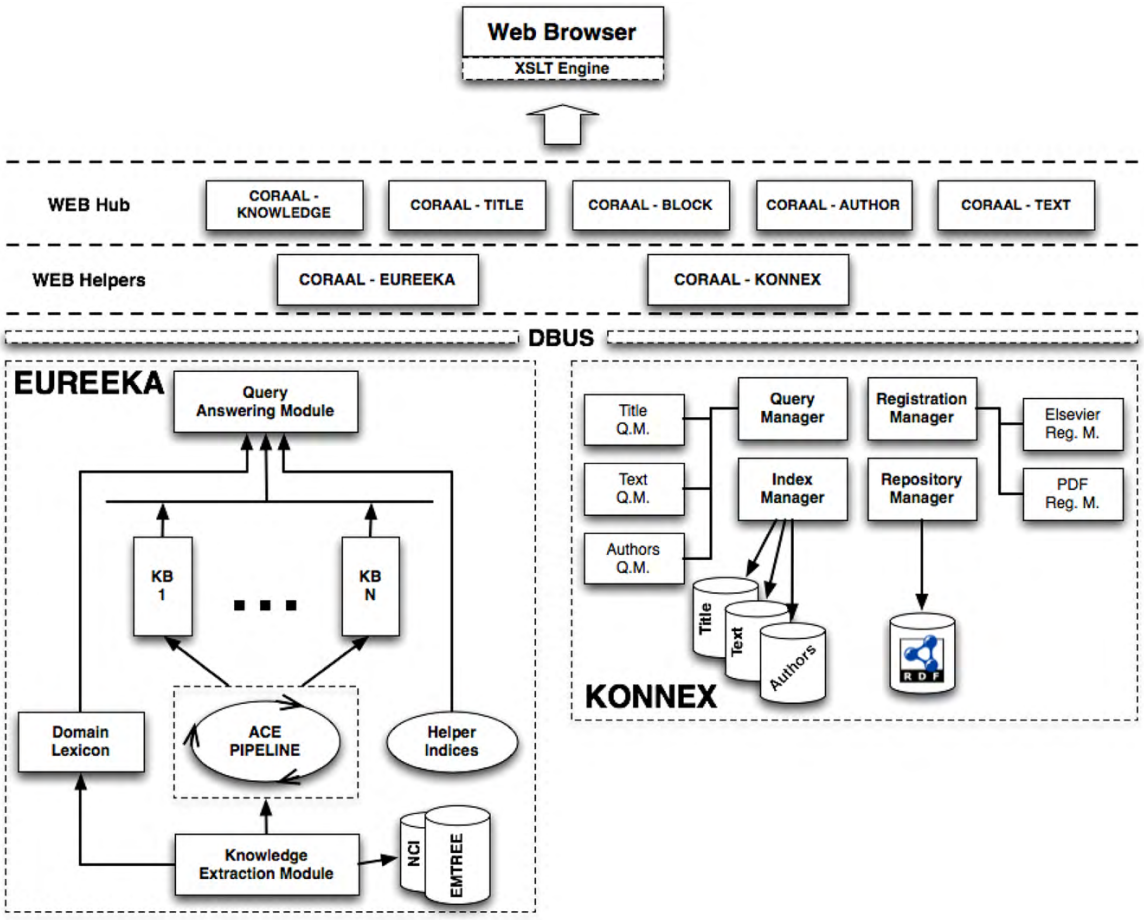


Fig. 1. CORAAL architecture.

(iii) Comprehensive integration, augmentation, and refinement of the extracted content, possibly using extant machine-readable resources (e.g., life science thesauri or vocabularies). (iv) Interlinking of the processed content (e.g., connecting relevant arguments across publications, or preserving provenance of statements about relations between particular concepts). (v) Intuitive access to and display of the content extracted from publications, so that everybody can easily search for the extracted knowledge and track its provenance. (vi) Methods for collaborative curation of the resulting content, so that global expert communities can contribute to further refinement of the extracted publication knowledge.

CORAAL constitutes a particular solution to some of the principal problems inherent in knowledge-based publication search. We provide an overview of the system and its implementation in Section 2. Then we describe its application and evaluation within the Elsevier Grand Challenge in Section 3. Summary of related systems is given in Section 4. Section 5 concludes the paper with a discussion of a future outlook.

2. CORAAL essentials

In the following we introduce the basic features of the CORAAL system. Section 2.1 provides an overview of CORAAL, its architecture, and relevant implementation details. In Section 2.2 we describe the pipeline in which we extract, process, and display the knowledge and metadata extracted from publications.

2.1. Overview of the solution

In order to provide comprehensive search capabilities in CORAAL, we augment standard (full-text) publication search with novel services that enable knowledge-based search. By knowledge-based search we mean the ability to query for and browse statements that capture relations between concepts in the retrieved source articles.

CORAAL is built on top of two substantial research products of our group at DERI—the KONNEX [4] and EUREEKA [8] frameworks. The former is used for storing and querying of full-text publications and associated metadata. The latter supports exploitation of the knowledge implicitly contained in the texts by means of knowledge-based search.

CORAAL itself essentially extracts asserted publication metadata together with the knowledge implicitly present in the respective text, integrates the emergent content with existing domain knowledge, and displays it via a multiple-perspective search&browse interface. This allows fine-grained publication search to be combined with convenient and effortless large-scale exploitation of the knowledge associated with and implicit within the texts.

2.1.1. Architecture

The architecture of CORAAL is depicted in Fig. 1. The EUREEKA library caters for knowledge extraction from text and other knowledge resources (e.g., ontologies or machine readable thesauri) via

the knowledge extraction module. After being processed by the ACE knowledge refinement and augmentation pipeline (details provided in Section 2.2), new facts are added into a knowledge base. There can be multiple knowledge bases if users wish to keep content from different domains separate. The knowledge bases are exposed to consumers via a semantic query answering module. Indices are used to help optimize the retrieval and sorting of statements based upon relevance scores.

Another crucial part of the CORAAL back-end is KONNEX, which processes the publication text and metadata in order to complement the knowledge-based search supported by EUREEKA by rather traditional full-text services. KONNEX integrates the texts and metadata extracted from the input publication corpus in a triple store, representing all the information as RDF graphs (cf. http://www.w3.org/TR/rdf-primer/). Operations related to data incorporation and necessary full-text indices (for the publication text and particular metadata types) are handled by dedicated manager modules.

2.1.2. Relevant implementation details

Since CORAAL contains several conceptually separate modules, we utilise an inter-process communication layer implemented using the D-BUS framework (cf. http://dbus.freedesktop.org/). A set of proxy helper services rests on top of the core-level EUREEKA and KONNEX APIs. These manage the user requests and forward the data returned by the core APIs to the so-called web hub layer, which is organized as a decoupled set of stateless web services, each of which handles particular types of search.

The web services produce machine-readable RDF expressions that represent answers to user queries. The RDF has XSL style sheets attached, allowing both its rendering for human readability and machine consumption to be provided simultaneously. The human-readable form is also enhanced by the Exhibit faceted browsing web front-end (cf. http://www.simile-widgets.org/exhibit/). A hidden advantage of this mechanism is the shifting of the processing of the data for visualization purposes from the server to the client side, as the XSL transformation is performed by the client's browser. Such a solution results in CORAAL being a pure Semantic Web application, as the data flow between the core infrastructure and the other modules is strictly based on RDF graphs.

2.2. Knowledge acquisition pipeline

The publications' metadata and full text were stored and indexed within KONNEX for link processing [4]. After parsing the input articles (either directly from PDF, or from annotated text-based files as provided, e.g., by Elsevier), the metadata and structural annotations were processed by KONNEX. First we eliminated possible duplicate metadata annotations using a string-based similarity heuristic. Each article was then represented as a comprehensive RDF graph consisting of its shallow metadata, such as title, authors, linear structure with pointers to the actual content (sections, paragraphs, etc.), and references. The references were anchored in citation contexts (i.e., paragraphs they occur in), and represented as individual graphs allowing for incremental enrichment over time. The article's full-text information was managed using multiple Lucene indices (cf. http://lucene.apache.org/), while the graphs were integrated and linked within the KONNEX RDF repository.

While KONNEX catered for the raw publication text and metadata, exploitation of the more structured publication knowledge was tackled by our novel EUREEKA framework for emergent (e.g., automatically extracted) knowledge processing [8]. The framework builds on a simple subject-predicate-object triple model. We extend the subject-predicate-object triples by adding positive or negative certainty measures and organised them in so-called conceptual matrices, concisely representing every positive and negative relation between an entity and other entities. Metrics can be easily defined on the conceptual matrices. The metrics then serve as a natural basis for context-dependent concept similarity scoring that provides the basic light-weight empirical semantics in EUREEKA. On top of the similarity-based semantics, we implemented two simple yet practical inference services: (i) retrieval of knowledge similar to an input concept, and/or its extension by means of similar stored content; (ii) rule-based materialisation of relations implied by the explicit knowledge base content, and/or complex querying (similarity as a basis for finding query variable instances for approximate evaluation of rules). The inference algorithms have anytime behaviour, meaning that it is possible to programmatically adjust their completeness/efficiency trade-off (i.e., one can either have complete, but possibly largely irrelevant set of solutions in a long time, or incomplete, but rather relevant set in a relatively short time). Technical details of the solution are out of scope of this system overview article, but one can find them in [8].

We applied our EUREEKA prototype to: (i) automate extraction of machine-readable knowledge from particular life science article texts; (ii) integrate, refine, and extend the extracted knowledge within one large emergent knowledge base; (iii) expose the processed knowledge via a query-answering and faceted browsing interface, tracking the article provenance of statements.

For the initial knowledge extraction, we used a heuristics based on natural language processing (NLP)—stemming essentially from [5,10]—to process chunk-parsed texts into subject-predicate-object-score quads.[2] The scores were derived from aggregated absolute and document-level frequencies of subject/object and predicate terms. The extracted quads encoded three major types of ontological relations between concepts: (I) taxonomical—*type*—relationships; (II) concept difference (i.e., negative *type* relationships); (III) "facet" relations derived from verb frames in the input texts (e.g., *has part*, *involves*, or *occurs in*). Over 27,000 types of facet relations were extracted. We imposed a taxonomy on them, considering the head verb of the phrase as a more generic relation (e.g., *involves expression of* was assumed to be a type of *involves*). Also, several artificial relation types were introduced to restrict the semantics of some of the most frequent relations: a (positive) *type* was considered transitive and anti-symmetric, and *same as* was set transitive and symmetric. Similarly, *part of* was assumed transitive and being inverse of *has part*. Note that the *has part* relation has rather general semantics within the extracted knowledge, i.e., its meaning is not strictly physically mereological, it can refer also to, e.g., conceptual parts or possession of entities.

The quads were processed as follows in the ACE pipeline (details of the particular steps are described in [8]):

(I) *A*ddition—The extracted quads were incrementally added into an growing knowledge base *K*, using a fuzzy aggregation of the relevant conceptual matrices. To take into account the basic domain semantics (i.e., synonymy relations and core taxonomy of *K*), we used the EMTREE (http://www.embase.com/emtree/) and NCI (http://nciterms.nci.nih.gov) thesauri.
(II) Closure—After the addition of new facts into *K*, we computed its materialisation according to imported RDFS entailment rules (cf. http://www.w3.org/TR/rdf-schema/).
(III) *E*xtension—the extracted concepts were analogically extended using similar stored knowledge.

[2] Implemented for English only in the current version. However, the EUREEKA framework itself is language-agnostic—it requires only input entities and their relations to be represented in an RDF-compatible format. Porting CORAAL to another language is quite straightforward, given a suitable relation extraction pipeline.

We display the content of the knowledge base via a query-answering module. Answers to queries are sorted according to their relevance scores and similarity to the query [8]. Answers are provided by an intersection of publication provenance sets corresponding to the respective statements' subject and object terms. The module currently supports queries in the following form: $t \mid s : (NOT\ )?p : o(\ AND\ s : (NOT\ )?p : o)^*$, where *NOT* and *AND* stands for negation and conjunction, respectively (the ? and * wildcards mean zero or one and zero or more occurrences of the preceding symbols, respectively, | stands for OR). s, o, p may be either a variable—anything starting with the ? character or even the ? character alone—or a lexical expression. t may be lexical expressions only.

3. Elsevier grand challenge deployment

This section describes the deployment of CORAAL for the Elsevier grand challenge. Section 3.1 describes the data we processed, while Section 3.2 illustrates the query answering capabilities of CORAAL using the example outlined in the article's introduction. Finally, Sections 3.3 and 3.4 report on the continuous tests with real users and on the evaluation of the quality of the exposed knowledge.

3.1. Data

Input: As of March 2009, we had processed 11,761 Elsevier journal articles from the provided XML repositories that were related to cancer research and treatment. Access to the articles was provided within the Elsevier Grand Challenge competition (cf. http://www.elseviergrandchallenge.com). The domain was selected to conform to the expertise of our sample users and testers from Masaryk Oncology Institute in Brno, Czech Republic. We processed cancer-related articles from a selection of Elsevier journals focusing on oncology, genetics, pharmacology, biochemistry, general biology, cell research and clinical medicine. From the article repository, we extracted the knowledge and publication metadata for further processing by CORAAL. Besides the publications themselves, we employed extant machine-readable vocabularies for the refinement and extension of the extracted knowledge (currently, we use the NCI and EMTREE thesauri).

Output: CORAAL exposes two datasets as an output of the publication processing: First, we populated a triple store with publication metadata (citations, their contexts, structural annotations, titles, authors and affiliations) and built auxiliary indices for each metadata type to facilitate full-text querying of the stored content. The resulting store contained 7,608,532 RDF subject-predicate-object statements describing the input articles. This included 247,392 publication titles and 374,553 authors (extracted from both processed articles and their literature reference lists).

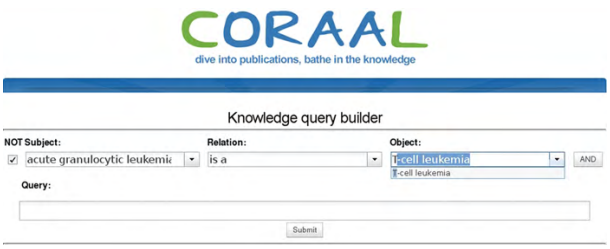


Fig. 2. Knowledge-based query construction.

Apart from the triple store, we employed a custom EUREEKA knowledge base [8], containing facts of variable levels of certainty extracted and inferred from the article texts and the imported life science thesauri. Over 215,000 concepts were extracted from the articles. Together with the data from the initial thesauri, the domain lexicon contained 622,611 terms, referring to 347,613 unique concepts. The size of the emergent knowledge base was 4,715,992 weighed statements (ca. 99 and 334 extracted and inferred statements per publication on average, respectively). The contextual knowledge related to the statements, namely provenance information, amounted to more than 10,000,000 additional statements (when expressed in RDF triples). Query evaluation on the produced content typically took fractions of seconds.

3.2. Asking queries, browsing answers

CORAAL can answer classical full-text or knowledge-based queries using a simple yet powerful query language (details are given in http://smile.deri.ie/projects/egc/quickstart). Answers in CORAAL are presented as a list of query-conforming *s* tatements (for the knowledge-based search) or *resources* (publication titles, paragraphs or author names for the full-text search). The statement results can be filtered based on their particular elements (e.g., subjects, properties, and objects), associated contextual information and whether the statement is positive or negative. The resource

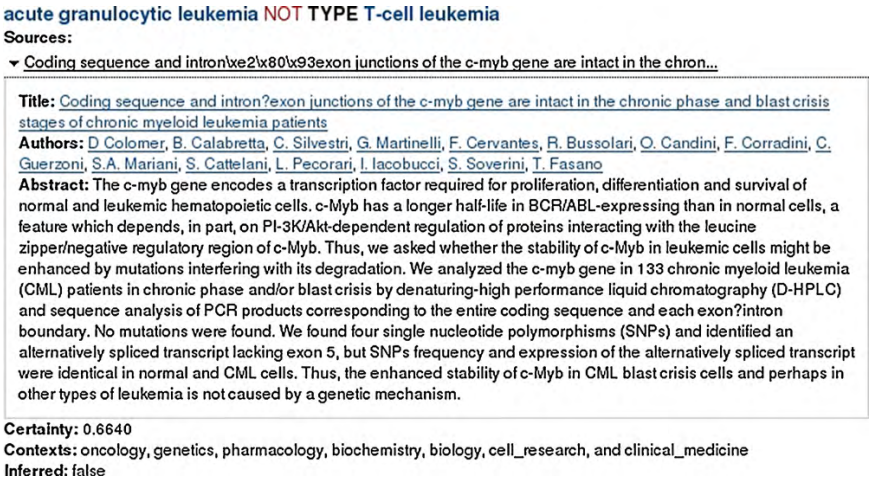


Fig. 3. Query answer detail.

results can be filtered according to the concepts associated with them (both extracted and inferred) and additional metadata (e.g., authors or citations present in the context of the resulting paragraphs). Using filtering (i.e., faceted browsing), one can quickly focus on items of interest within the whole result set.

Recalling the example from Section 1, the query for sources supporting that *acute granulocytic leukemia* is different from *T-cell leukemia* can be conveniently constructed in CORAAL as depicted in Fig. 2 (guided query building using a form-based interface).

The query builder includes a context-sensitive auto-completion capability; if one rests the cursor on, e.g., a subject, only relations (properties) actually associated with that subject in the knowledge base are displayed.

Fig. 3 shows the highest ranked answer to the query constructed in Fig. 2, proving that the two types of leukemia are not the same. The source article of the statement (displayed as an inline summary in Fig. 3) is the desired reference supporting the claim. The particular types of contextual information associated with statements (as can be observed in Fig. 3) are: (I) *source* provenance—articles relevant to the statement, which can be expanded into an inline summary (as shown in Fig. 3) or explored in detail after clicking on the respective publication title; (II) *context* provenance—domain of life sciences that the statement relates to (determined according to the main topic of the journal that contained the articles the statement was extracted from); (III) *certainty* —a number describing how certain the system is that the statement holds and is relevant to the query (values between 0 and 1; derived from the absolute value of the respective statement degree and from the actual similarity of the statement to the query); (IV) *inferred*—a Boolean value determining whether the statement was inferred or not (the latter indicating it was directly extracted). More information can be seen with CORAAL at http://coraal.deri.ie:8080/coraal.

3.3. Continuous tests with users

During the development of the CORAAL prototype, we continually collaborated with several biomedical experts, who formed a committee of sample users and evaluators. Before the final stages of the Elsevier Grand Challenge, we prepared five tasks to be worked out with CORAAL and a baseline application (ScienceDirect or PubMed). Our hypothesis was that users should perform better with CORAAL than with the baseline, since the tasks were focused on knowledge rather than on a plain text-based search.[3]

Users indicated that the tasks used for evaluating CORAAL were relevant to their day to day work by giving it a score of 3.9 out of 6 (the scale was from 1 to 6, with 1 indicating no relevance, and 6 indicating high relevance). The success rate of task accomplishment was 60.7% with CORAAL and 10.7% with the baseline application. This confirms our above-mentioned hypothesis that users will be able to accomplish the given tasks better with CORAAL due to its enhanced querying and search capabilities.

Besides evaluating the users' performance in sample knowledge-based search tasks, we interviewed them regarding the overall usability of the CORAAL interface. The most critical issue was related to the query language—half of the users were not always able to construct appropriate queries. However, CORAAL also offers a form-based query builder that assists the user as illustrated in Section 3.2. Using this feature, users performed up to six times faster and 40% more efficiently than with purely manually constructed queries.

The expert users also had problems with too general, obvious, or irrelevant results. These concerns were expressed when the users were presented with a raw list of answer statements within the evaluation. After being discussed with the users within the evaluation interviews, the problems were addressed by the following features in the user interface: (i) relevance-based sorting of concepts and statements [8]—the most relevant statements were displayed at the top of the results list; (ii) intuitive faceted browsing functionality—support for fast and easy reduction of the displayed results to a subset which reference particular entities (i.e., statements having only certain objects or authors writing about certain topics). The solutions were considered as mostly sufficient regarding the users' concerns (an average $4.\bar{6}$ score on the $1-6$ scale going from least to most sufficient).

3.4. Knowledge quality evaluation

To evaluate the quality of the knowledge served by CORAAL,[4] we generated 200 random queries composed of anything from single terms to a conjunction of multiple possibly negated statements. To ensure non-empty answer sets, the queries were generated from the actual content of the knowledge base. Also, we took into account only the content extracted from the input articles and not from the NCI or EMTREE seed thesauri. We let our domain expert committee vote on the relevance of queries to their day-to-day work and used the ten most relevant ones to evaluate the answers provided by CORAAL.

We used the traditional notions of precision, recall, and F-measure for the evaluation of the quality of the answers. A gold standard set of statements relevant to the queries used in the evaluation was created by the user committee, who employed their own knowledge combined with the full-text search of the publications incorporated in CORAAL. For a baseline comparison, we imported the knowledge extracted by CORAAL from the input articles and thesauri into a state-of-the art RDF store. The store had inference and querying support, however, it lacked proper means for emergent knowledge processing (namely regarding the negation, uncertainty, inconsistence resolution and approximate query processing features). The set of queries used for CORAAL evaluation was executed using the baseline RDF store and the results were compared. Due to the novel support of the emergent knowledge, CORAAL quite substantially outperformed the baseline, achieving F-measures from two- to eight-times better for the various evaluated features.

The absolute CORAAL results may still be considered rather poor when compared to the gold standard generated by the users (i.e., F-measures for some queries around 0.2). However, one must recognise that the answers to the gold standard questions took almost two working days for an expert committee to generate. In about the same time, the CORAAL knowledge base was produced purely automatically for much larger amounts of data (involving statements about hundreds of thousands of concepts instead of a few query entities). The queries take seconds to evaluate and one can find many relevant answers very quickly due to the relevance-based sorting of the results (the first 10 statements contained more than 67% of relevant answers on average, while the 200th to 400th results contained only about 5% correct statements). The evaluation committee unequivocally considered the ability of CORAAL

[3] For instance, the users were asked to find all authors who support the fact that the `acute granulocytic leukemia` and `T-cell leukemia` concepts are disjoint, or to find which process is used as a `complementary method`, while being different from the `polymerase chain reaction`, and identify publications that support their findings.

[4] Note that this section provides only an outline of the actual evaluation, summarising the most important points. A full description of the knowledge quality evaluation and the numeric results achieved is provided in [8].

to perform purely automatically as an acceptable trade-off for the detected noise in the results.

4. Related work

Approaches tackling problems related to those addressed by the core technologies powering CORAAL are analysed in [8,4]. Here, we offer an overview of systems targeting similar problems to those tackled by our framework.

State-of-the-art applications like ScienceDirect or PubMed Central require almost no effort in order to expose arbitrary life science publications for search (therefore we used them as a baseline in the user-centric experiment). However, the benefit they provide is rather limited when compared to cutting-edge approaches aimed at utilising also the publication knowledge within the query construction and/or result visualisation. Such innovative solutions may require much more a priori effort in order to work properly, though.

FindUR [6], Melisa [1] and GoPubMed [3] are ontology-based interfaces to a traditional publication full-text search. GoPubMed allows for effective restriction and intelligent visualisation of the query results. FindUR and Melisa support focusing the queries on particular topics based on an ontology (FindUR uses a Description Logic ontology built from scratch, while Melisa employs a custom ontology based on MeSH, cf. http://www.nlm.nih.gov/mesh/). GoPubMed dynamically extracts parts of the Gene Ontology (cf. http://www.geneontology.org/) relevant to the query, which are then used for restriction and a sophisticated visualisation of the classical PubMed search results. Nevertheless, none of the tools mentioned so far offers querying for or browsing of arbitrary publication knowledge. Terms and relations not present in the systems' rather static ontologies simply cannot be reflected in the search. On the other hand, CORAAL works on any domain and extracts arbitrary knowledge from publications automatically, although the offered benefits may not be that high due to a possibly higher level of noise.

Textpresso [7] is quite similar to CORAAL concerning searching for relations between concepts in particular chunks of text. However, the underlying ontologies and their instance sets have to be provided manually, whereas CORAAL can operate with or without any available ontology. Moreover, CORAAL includes far more full-text publications and concepts.

The biggest challenge of systems with goals similar to CORAAL is a reliable automation of truly expressive content extraction. In contrast to CORAAL, none of the related systems addresses this problem appropriately, which makes them scale poorly, or makes them difficult to port to new domains. This is why we were not able to use the related systems for a baseline comparison in our domain-specific application scenario—we simply could not adapt them so that they would be able to perform reasonably, both due to technical difficulties and lack of necessary resources.

5. Conclusions and future work

With CORAAL, we have addressed most of the elements of a truly knowledge-based scientific publication search as specified in Section 1. We are able to extract and integrate emergent knowledge and metadata from a large number of publications, as well as augment and refine the extracted content. CORAAL also allows for intuitive searching and browsing of the processed knowledge. Although the primary focus of CORAAL is the knowledge-based search, the underlying technologies are straightforwardly applicable to many other tasks. These are for instance automated tagging of articles by the associated general concepts, population of existing domain-specific vocabularies, or utilisation of CORAAL as a general-purpose knowledge back-end exposing arbitrary services (e.g., knowledge-based retrieval of similar articles or profile-based article recommendation).

However, we still have to tackle several challenges in order to fully realize the current potential of CORAAL. First, we want to utilise the wisdom of the crowds by supporting intuitive and unobtrusive community-based curation of the emergent knowledge, namely by validation or invalidation of existing statements, introduction of new statements and submission of new rules refining the domain semantics. Then we intend to make the step from CORAAL to a CORAAL reef, a distributed peer-to-peer model covering multiple CORAAL installations autonomously communicating with each other (e.g., asking for answers when no answer is available locally or exchanging appropriate rules to improve the local semantics). After incorporating the capabilities of the prospective CORAAL reefs into the ecosystem of the current online publishing, we will be able to unlock, connect, augment and retrieve the knowledge with unprecedented scale and efficiency.

Acknowledgments

This work has been supported by the 'Líon', 'Líon II' projects funded by SFI under Grants No. SFI/02/CE1/I131, SFI/08/CE/I1380, respectively. Big thanks goes to our evaluators: Doug Foxvog, Peter Grell, MD, Miloš Holánek, MD, Matthias Samwald, Holger Stenzhorn and Jiří Vyskočil, MD. We also appreciated the challenge judges' feedback that helped to streamline the final prototype a lot. We acknowledge the support provided by Noelle Gracy, Anita de Waard and other Elsevier people regarding the challenge organisation. Finally, we thank to the anonymous reviewers and to Ed Hovy and Susie Stephens, the special issue editors, who all helped us to improve the final article version a lot.

References

[1] J.M. Abasolo, M. Gómez, M. Melisa, An ontology-based agent for information retrieval in medicine, in: Proceedings of the SemWeb2000, 2000.
[2] S. Bechhofer, et al., Tackling the ontology acquisition bottleneck: an experiment in ontology re-engineering, at http://tinyurl.com/96w7ms, Apr'08. (2003).
[3] H. Dietze, et al., Gopubmed: exploring pubmed with ontological background knowledge, in: Ontologies and Text Mining for Life Sciences, IBFI, 2008.
[4] T. Groza, S. Handschuh, K. Moeller, S. Decker, KonneXSALT: first steps towards a semantic claim federation infrastructure, in: Proceedings of the ESWC 2008, Springer-Verlag, 2008.
[5] A. Maedche, S. Staab, Discovering conceptual relations from text, in: Proceedings of the ECAI 2000, IOS Press, 2000.
[6] D.L. McGuinness, Ontology-enhanced search for primary care medical literature, in: Proceedings of the Medical Concept Representation and NLP Conference, 1999.
[7] H.M. Müller, E.E. Kenny, P.W. Sternberg, Textpresso: an ontology-based information retrieval and extraction system for biological literature, PLoS Biology 2 (11) (2004) e309.
[8] V. Nováček, S. Decker, Towards lightweight and robust large scale emergent knowledge processing, in: Proceedings of the ISWC'09, 2009.
[9] J. Tsujii, Refine and pathtext, which combines text mining with pathways, in: Keynote at SESL 2009, 2009.
[10] J. Voelker, D. Vrandecic, Y. Sure, A. Hotho, Learning disjointness, in: Proceedings of the ESWC'07, Springer, 2007.

Chapter 7

Distributional Semantics in Semantic Literature Search

PeerJ

SKIMMR: facilitating knowledge discovery in life sciences by machine-aided skim reading

Vít Nováček[1] and Gully A.P.C. Burns[2]

[1] Insight Centre (formerly DERI), National University of Ireland Galway, Galway, Ireland
[2] Information Sciences Institute, University of Southern California, Marina del Rey, CA, USA

ABSTRACT

Background. Unlike full reading, 'skim-reading' involves the process of looking quickly over information in an attempt to cover more material whilst still being able to retain a superficial view of the underlying content. Within this work, we specifically emulate this natural human activity by providing a dynamic graph-based view of entities automatically extracted from text. For the extraction, we use shallow parsing, co-occurrence analysis and semantic similarity computation techniques. Our main motivation is to assist biomedical researchers and clinicians in coping with increasingly large amounts of potentially relevant articles that are being published ongoingly in life sciences.

Methods. To construct the high-level network overview of articles, we extract weighted binary statements from the text. We consider two types of these statements, co-occurrence and similarity, both organised in the same distributional representation (i.e., in a vector-space model). For the co-occurrence weights, we use point-wise mutual information that indicates the degree of non-random association between two co-occurring entities. For computing the similarity statement weights, we use cosine distance based on the relevant co-occurrence vectors. These statements are used to build fuzzy indices of terms, statements and provenance article identifiers, which support fuzzy querying and subsequent result ranking. These indexing and querying processes are then used to construct a graph-based interface for searching and browsing entity networks extracted from articles, as well as articles relevant to the networks being browsed. Last but not least, we describe a methodology for automated experimental evaluation of the presented approach. The method uses formal comparison of the graphs generated by our tool to relevant gold standards based on manually curated PubMed, TREC challenge and MeSH data.

Results. We provide a web-based prototype (called 'SKIMMR') that generates a network of inter-related entities from a set of documents which a user may explore through our interface. When a particular area of the entity network looks interesting to a user, the tool displays the documents that are the most relevant to those entities of interest currently shown in the network. We present this as a methodology for browsing a collection of research articles. To illustrate the practical applicability of SKIMMR, we present examples of its use in the domains of Spinal Muscular Atrophy and Parkinson's Disease. Finally, we report on the results of experimental evaluation using the two domains and one additional dataset based on the TREC challenge. The results show how the presented method for machine-aided skim reading

Submitted 6 December 2013
Accepted 23 June 2014
Published 22 July 2014

Corresponding author
Vít Nováček, vit.novacek@deri.org

Academic editor
Harry Hochheiser

Additional Information and Declarations can be found on page 35

DOI **10.7717/peerj.483**

How to cite this article Nováček and Burns (2014), SKIMMR: facilitating knowledge discovery in life sciences by machine-aided skim reading. **PeerJ 2:e483; DOI 10.7717/peerj.483**

outperforms tools like PubMed regarding focused browsing and informativeness of the browsing context.

Subjects Bioinformatics, Neuroscience, Human–Computer Interaction, Computational Science
Keywords Machine reading, Skim reading, Publication search, Text mining, Information visualisation

INTRODUCTION

In recent years, knowledge workers in life sciences are increasingly overwhelmed by an ever-growing quantity of information. PubMed[1] contained more than 23 million abstracts as of November 2013, with a new entry being added every minute. The current textual content available online as PubMed abstracts amount to over 2 billion words (based on estimates derived from a random sample of about 7,000 records). Information retrieval technology helps researchers pinpoint individual papers of interest within the overall mass of documents, but how can scientists use that to acquire a sense of the overall organization of the field? How can users discover new knowledge within the literature when they might not know what they are looking for ahead of time?

[1] The central US repository of published papers in the life sciences since the 1950s, see http://www.ncbi.nlm.nih.gov/pubmed.

Strategic reading aided by computerised solutions may soon become essential for scientists (*Renear & Palmer, 2009*). Our goal is to provide a system that can assist readers to explore large numbers of documents efficiently. We present 'machine-aided skim-reading' as a way to extend the traditional paradigm of searching and browsing a text collection (in this case, PubMed abstracts) through the use of a search tool. Instead of issuing a series of queries to reveal lists of ranked documents that may contain elements of interest, we let the user search and browse a *network of entities and relations* that are explicitly or implicitly present in the texts. This provides a simplified and high-level overview of the domain covered by the text, and allows users to identify and focus on items of interest without having to read any text directly. Upon discovering an entity of interest, the user may transition from our 'skimming' approach to read the relevant texts as needed.

This article is organised as follows. 'Methods' describes methods used in SKIMMR for: (1) extraction of biomedical entities from data; (2) computation of the co-occurrence and similarity relationships between the entities; (3) indexing and querying of the resulting knowledge base; (4) evaluating the knowledge base using automated simulations. Each of the methods is explained using examples. 'Results' presents the SKIMMR prototype and explains typical usage of the tool in examples based on user interactions. We also describe evaluation experiments performed with three different instances of the tool. In 'Discussion' we discuss the results, give an overview of related work and outline our future directions. There is also 'Formulae Definitions' that provides details on some of the more complex formulae introduced in the main text.

The main contributions of the presented work are: (1) machine-aided skim-reading as a new approach to semi-automated knowledge discovery; (2) fuzzy indexing and querying method for efficient on-demand construction and presentation of the high-level

graph-based article summaries; (3) detailed examples that explain the applied methods in a step-by-step fashion even to people with little or no computer science background; (4) an open-source prototype implementing the described method, readily available for processing custom data, and also in the form of two pre-computed instances deployed on Spinal Muscular Atrophy and Parkinson's Disease data; (5) an evaluation methodology based on simulations and formally defined measures of semantic coherence, information content and complexity that can be used not only for evaluating SKIMMR (as we did in the article), but also for assessment of other tools and data sets utilising graph structures.

METHODS

This section describes how the knowledge base supporting the process of machine-aided skim reading is generated from the input data (i.e., biomedical articles and data). Firstly we describe extraction of entities and basic co-occurrence relationships between them ('Extracting basic co-occurrence statements from texts'). 'Computing a knowledge base from the extracted statements' is about how we compute more general, corpus-wide relationships from the basic extracted co-occurrence statements. 'Indexing and querying the knowledge base' explains how the processed content can be indexed and queried in order to generate the graph-based summaries with links to the original documents. Finally, 'Evaluation methodology' introduces a method for a simulation-based evaluation of the generated content in the context of machine-aided skim reading. For the research reported in this article, we received an exemption from IRB review by the USC UPIRB, under approval number UP-12-00414.

Extracting basic co-occurrence statements from texts

We process the abstracts by a biomedical text-mining tool[2] in order to extract named entities (e.g., drugs, genes, diseases or cells) from the text. For each abstract with a PubMed ID *PMID*, we produce a set of $(e_x, e_y, cooc((e_x, e_y), PubMed_{PMID}), PubMed_{PMID})$ tuples, where e_x, e_y range over all pairs of named entities in the abstract with the *PMID* identifier, and $cooc((e_x, e_y), PubMed_{PMID})$ is a co-occurrence score of the two entities computed using the formula (1) detailed in 'Co-occurrences'. The computation of the score is illustrated in the following example.

[2] A part of the LingPipe suite, see http://alias-i.com/lingpipe/ for details.

Example 1 *Imagine we want to investigate the co-occurrence of the* `parkinsonism` *and* `DRD` *(dopamine-responsive dystopia) concepts in a data set of PubMed abstracts concerned with clinical aspects of Parkinson's disease.*[3] *There are two articles in the data set where the corresponding terms co-occur:*

[3] Which we have processed in one of the pre-computed instances of SKIMMR, see 'Parkinson's disease' for details.

- *Jeon BS, et al.* Dopamine transporter density measured by 123Ibeta-CIT single-photon emission computed tomography is normal in dopa-responsive dystonia *(PubMed ID: 9629849).*
- *Snow BJ, et al.* Positron emission tomographic studies of dopa-responsive dystonia and early-onset idiopathic parkinsonism *(PubMed ID: 8239569).*

The relevant portions of the first abstract (PubMed ID: 9629849) are summarised in the following table (split into sentences numbered from the beginning of the text):

…	…
12	Therefore, we performed 123Ibeta-CIT single-photon emission computed tomography (123Ibeta-CIT SPECT) in clinically diagnosed DRD, PD, and JPD, and examined whether DAT imaging can differentiate DRD from PD and JPD.
…	…
14	Five females (4 from two families, and 1 sporadic) were diagnosed as DRD based on early-onset foot dystonia and progressive parkinsonism beginning at ages 7–12.
…	…
17	123Ibeta-CIT striatal binding was normal in DRD, whereas it was markedly decreased in PD and JPD.
…	…
22	A normal striatal DAT in a parkinsonian patient is evidence for a nondegenerative cause of parkinsonism and differentiates DRD from JPD.
23	Finding a new mutation in one family and failure to demonstrate mutations in the putative gene in other cases supports the usefulness of DAT imaging in diagnosing DRD.

Based on the sentence numbers in the excerpt, we can compute the co-occurrence score of the (parkinsonism, DRD) *tuple as:*

$$cooc((\texttt{parkinsonism}, \texttt{DRD}), PubMed_{9629849}) = \left(1 + \frac{1}{4} + \frac{1}{3} + \frac{1}{3}\right) + \left(1 + \frac{1}{2}\right) = 3.41\bar{6}.$$

Similar to the above, the portions relevant to the (parkinsonism, DRD) *co-occurrences according to the second abstract (PubMed ID: 8239569) are as follows:*

1	There are two major syndromes presenting in the early decades of life with dystonia and parkinsonism: dopa-responsive dystonia (DRD) and early-onset idiopathic parkinsonism (EOIP).
2	DRD presents predominantly in childhood with prominent dystonia and lesser degrees of parkinsonism.
…	…
5	Some have suggested, however, that DRD is a form of EOIP.
…	…

The co-occurrence score is then:

$$cooc((\texttt{parkinsonism}, \texttt{DRD}), PubMed_{8239569}) = \left(1 + \frac{1}{2} + 1 + \frac{1}{2}\right) + \frac{1}{4} = 3.25.$$

Therefore the basic co-occurrence tuples produced from the two articles are:

$(\texttt{parkinsonism}, \texttt{DRD}, 3.41\bar{6}, PubMed_{9629849})$,
$(\texttt{parkinsonism}, \texttt{DRD}, 3.25, PubMed_{8239569})$.

Computing a knowledge base from the extracted statements

From the basic co-occurrence statements, we compute a knowledge base, which is a comprehensive network of interlinked entities. This network supports the process of navigating a skeletal structure of the knowledge represented by the corpus of the input PubMed articles (i.e., the actual skim reading). The knowledge base consists of two types of statements: (1) corpus-wide co-occurrence and (2) similarity. The way to compute the particular types of statements in the knowledge base is described in the following two sections.

Corpus-wide co-occurrence

The basic co-occurrence tuples extracted from the PubMed abstracts only express the co-occurrence scores at the level of particular documents. We need to aggregate these scores to examine co-occurrence across the whole corpus. For that, we use point-wise mutual information (*Manning, Raghavan & Schütze, 2008*), which determines how much two co-occurring terms are associated or disassociated, comparing their joint and individual distributions over a data set. We multiply the point-wise mutual information value by the absolute frequency of the co-occurrence in the corpus to prioritise more frequent phenomena. Finally, we filter and normalise values so that the results contain only scores in the $[0, 1]$ range. The scores are computed using the formulae (2)–(5) in 'Co-occurrences'.

The aggregated co-occurrence statements that are added to the knowledge base are in the form of $(x, cooc, y, v(fpmi(x, y), P))$ triples, where x, y range through all terms in the basic co-occurrence statements, the scores are computed across all the documents where x, y co-occur, and the *cooc* expression indicates co-occurrence as the actual type of the relation between x, y. Note that the co-occurrence relation is symmetric, meaning that if $(x, cooc, y, w_1)$ and $(y, cooc, x, w_2)$ are in the knowledge base, w_1 must be equal to w_2.

Example 2 *Assuming our corpus consists only of the two articles from Example 1, the point-wise mutual information score of the* (parkinsonism, DRD) *tuple can be computed using the following data:*

- p(parkinsonism, DRD)*–joint distribution of the* (parkinsonism, DRD) *tuple within all the tuples extracted from the PubMed abstracts with IDs 9629849 and 8239569, which equals* $3.41\bar{6} + 3.25 = 6.\bar{6}$ *(sum across all the* (parkinsonism, DRD) *basic co-occurrence tuples);*
- p(parkinsonism), p(DRD)*–individual distributions of the* parkinsonism, DRD *arguments within all extracted tuples, which equal 28.987 and 220.354, respectively (sums of the weights in all basic co-occurrence statements that contain* parkinsonism *or* DRD *as one of the arguments, respectively);*
- F(parkinsonism, DRD), $|T|$*–the absolute frequency of the* parkinsonism, DRD *co-occurrence and the number of all basic co-occurrence statements extracted from the abstracts, which equals to 2 and 1,414, respectively;*
- P*–the percentile for the normalisation, equal to 95, which results in the normalisation constant 2.061 (a non-normalised score such that only 5% of the scores are higher than that).*

The whole formula is then:

$$npmi(\texttt{parkinsonism, DRD}) = v(fpmi(\texttt{parkinsonism, DRD}), P) =$$
$$= v(F(\texttt{parkinsonism, DRD}) \cdot log_2 \frac{p(\texttt{parkinsonism,DRD})}{p(\texttt{parkinsonism})p(\texttt{DRD})}, 95) \doteq$$
$$\doteq \frac{2 \cdot log_2 \frac{6.\bar{6}}{28.987 \cdot 220.354}}{2.061} \doteq 0.545.$$

Thus the aggregated co-occurrence statement that is included in the knowledge base is

$(\texttt{parkinsonism}, \texttt{cooc}, \texttt{DRD}, 0.545)$.

Similarity

After having computed the aggregated and filtered co-occurrence statements, we add one more type of relationship–similarity. Many other authors have suggested ways for computing semantic similarity (see *d'Amato, 2007* for a comprehensive overview). We base our approach on cosine similarity, which has become one of the most commonly used approaches in information retrieval applications (*Singhal, 2001*; *Manning, Raghavan & Schütze, 2008*). The similarity and related notions are described in detail in 'Similarities', formulae (6) and (7).

Similarity indicates a higher-level type of relationship between entities that may not be covered by mere co-occurrence (entities not occurring in the same article may still be similar). This adds another perspective to the network of connections between entities extracted from literature, therefore it is useful to make similarity statements also a part of the SKIMMR knowledge base. To do so, we compute the similarity values between all combinations of entities x, y and include the statements $(x, sim, y, sim(x, y))$ into the knowledge base whenever the similarity value is above a pre-defined threshold (0.25 is used in the current implementation).[4]

[4] Similar to the co-occurrence statements described before, the *sim* expression refers to the type of the relation between x, y, i.e., similarity.

A worked example of how to compute similarity between two entities in the sample knowledge base is given below.

Example 3 *Let us use* '`parkinsonisms`', '`mrpi values`' *as sample entities* a, b. *In a full version of Parkinson's disease knowledge base (that contains the data used in the previous examples, but also hundreds of thousands of other statements), there are* 19 *shared entities among the ones related to* a, b *(for purposes of brevity, each item is linked to a short identifier to be used later on):* (1) `msa – p` $\sim t_0$, (2) `clinically unclassifiable parkinsonism` $\sim t_1$, (3) `cup` $\sim t_2$, (4) `vertical ocular slowness` $\sim t_3$, (5) `baseline clinical evaluation` $\sim t_4$, (6) `mr` $\sim t_5$, (7) `parkinsonian disorders` $\sim t_6$, (8) `psp phenotypes` $\sim t_7$, (9) `duration` $\sim t_8$, (10) `patients` $\sim t_9$, (11) `clinical diagnostic criteria` $\sim t_{10}$, (12) `abnormal mrpi values` $\sim t_{11}$, (13) `pd` $\sim t_{12}$, (14) `magnetic resonance parkinsonism index` $\sim t_{13}$, (15) `parkinson disease` $\sim t_{14}$, (16) `mri` $\sim t_{15}$, (17) `parkinson's disease` $\sim t_{16}$, (18) `psp` $\sim t_{17}$, (19) `normal mrpi values` $\sim t_{18}$.

The co-occurrence complements a, b of the `parkinsonisms`, `mrpi values` *entities (i.e., associated co-occurrence context vectors) are summarised in the following table:*

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_{10}	t_{11}	t_{13}	t_{14}	t_{15}	t_{17}	t_{18}
a	0.14	0.39	1.0	0.08	0.26	0.06	0.18	0.4	0.07	0.27	0.09	0.7	0.03	0.14	0.33	0.25
b	0.26	0.57	1.0	0.3	0.82	0.2	0.33	0.26	0.39	0.43	0.36	0.41	0.06	0.34	1.0	1.0

Note that the elements t_9, t_{12}, t_{16} are omitted since their weight in at least one of the complements is <0.01 and thus does not contribute significantly to the result. The sizes of the co-occurrence complement vectors are 3.048, 2.491 for `parkinsonisms`, `mrpi values`, *respectively, while their dot product is 2.773. Therefore their similarity is equal to* $\frac{2.773}{3.048 \cdot 2.491} \doteq 0.365$ *and the new statement to be added to the knowledge base is*

(`parkinsonisms`, `sim`, `mrpi values`, 0.365).

Indexing and querying the knowledge base

The main purpose of SKIMMR is to allow users to efficiently search and navigate in the SKIMMR knowledge bases, and retrieve articles related to the content discovered in the high-level entity networks. To support that, we maintain several indices of the knowledge base contents. The way how the indices are built and used in querying SKIMMR is described in the following two sections.

Knowledge base indices

In order to expose the SKIMMR knowledge bases, we maintain three main indices: (1) a *term* index–a mapping from entity terms to other terms that are associated with them by a relationship (like co-occurrence or similarity); (2) a *statement* index–a mapping that determines which statements the particular terms occur in; (3) a *source* index–a mapping from statements to their sources, *i.e.*, the texts from which the statements have been computed. In addition to the main indices, we use a full-text index that maps spelling alternatives and synonyms to the terms in the term index.

The main indices are implemented as matrices that reflect the weights in the SKIMMR knowledge base:

	T_1	T_2	…	T_n
T_1	$t_{1,1}$	$t_{1,2}$	…	$t_{1,n}$
T_2	$t_{2,1}$	$t_{2,2}$	…	$t_{2,n}$
⋮	⋮	⋮	⋱	⋮
T_n	$t_{n,1}$	$t_{n,2}$	…	$t_{n,n}$

	S_1	S_2	…	S_m
T_1	$s_{1,1}$	$s_{1,2}$	…	$s_{1,m}$
T_2	$s_{2,1}$	$s_{2,2}$	…	$s_{2,m}$
⋮	⋮	⋮	⋱	⋮
T_n	$s_{n,1}$	$s_{n,2}$	…	$s_{n,m}$

	P_1	P_2	…	P_q
S_1	$p_{1,1}$	$p_{1,2}$	…	$p_{1,q}$
S_2	$p_{2,1}$	$p_{2,2}$	…	$p_{2,q}$
⋮	⋮	⋮	⋱	⋮
S_m	$p_{m,1}$	$p_{m,2}$	…	$p_{m,q}$

where:

- $T_1, \ldots, T_n$ are identifiers of all entity terms in the knowledge base and $t_{i,j} \in [0, 1]$ is the maximum weight among the statements of all types existing between entities T_i, T_j in the knowledge base (0 if there is no such statement);

- $S_1, \ldots, S_m$ are identifiers of all statements present in the knowledge base and $s_{i,j} \in \{0, 1\}$ determines whether an entity T_i occurs in a statement S_j or not;
- $P_1, \ldots, P_q$ are identifiers of all input textual resources, and $p_{i,j} \in [0, 1]$ is the weight of the statement S_i if P_j was used in order to compute it, or zero otherwise.

Example 4 *To illustrate the notion of the knowledge base indices, let us consider a simple knowledge base with only two statements from Examples 1 and 3:* $S_1 \sim$ (`parkinsonism`, `cooc`, `DRD`, `0.545`), $S_2 \sim$ (`parkinsonisms`, `sim`, `mrpi values`, `0.365`). *Furthermore, let us assume that: (i) the statement* S_1 *has been computed from the articles with PubMed identifiers 9629849, 8239569 (being referred to by the* P_1, P_2 *provenance identifiers respectively); (ii) the statement* S_2 *has been computed from articles with PubMed identifiers 9629849, 21832222, 22076870 (being referred to by the* P_1, P_3, P_4 *provenance identifiers, respectively*[5]*). This corresponds to the following indices:*

[5] In reality, the number of source article used for computing these statements in Parkinson's disease knowledge base is much larger, but here we take into account only a few of them to simplify the example.

term index	`parkinsonism`	`DRD`	`parkinsonisms`	`mrpi values`
`parkinsonism`	0.0	0.545	0.0	0.0
`DRD`	0.545	0.0	0.0	0.0
`parkinsonisms`	0.0	0.0	0.0	0.365
`mrpi values`	0.0	0.0	0.365	0.0

statement index	S_1	S_2
`parkinsonism`	1.0	0.0
`DRD`	1.0	0.0
`parkinsonisms`	0.0	1.0
`mrpi values`	0.0	1.0

provenance index	P_1	P_2	P_3	P_4
S_1	0.545	0.545	0.0	0.0
S_2	0.0	0.0	0.365	0.365

Querying

The indices are used to efficiently query for the content of SKIMMR knowledge bases. We currently support atomic queries with one variable, and possibly nested combinations of atomic queries and propositional operators of conjunction (AND), disjunction (OR) and negation (NOT). An atomic query is defined as $? \leftrightarrow T$, where ? refers to the query variable and T is a full-text query term.[6] The intended purpose of the atomic query is to retrieve all entities related by any relation to the expressions corresponding to the term T. For instance, the $? \leftrightarrow$ `parkinsonism` query is supposed to retrieve all entities co-occurring-with or similar-to `parkinsonism`.

[6] One can expand the coverage of their queries using the advanced full-text search features like wildcards or boolean operators for the term look-up. Detailed syntax of the full-text query language we use is provided at http://pythonhosted.org/Whoosh/querylang.html.

Combinations consisting of multiple atomic queries linked by logical operators are evaluated using the following algorithm:

1. Parse the query and generate a corresponding 'query tree' (where each leaf is an atomic query and each node is a logical operator; the levels and branches of this tree reflect the nested structure of the query).

2. Evaluate the atomic queries in the nodes by a look-up in the term index, fetching the term index rows that correspond to the query term in the atomic query.
3. The result of each term look-up is a fuzzy set (*Hájek, 1998*) of terms related to the atomic query term, with membership degrees given by listed weights. One can then naturally combine atomic results by applying fuzzy set operations corresponding to the logical operators in the parsed query tree nodes (where conjunction, disjunction and negation correspond to fuzzy intersection, union and complement, respectively).
4. The result is a fuzzy set of terms $R_T = \{(T_1, w_1^T), (T_2, w_2^T), \ldots, (T_n, w_n^T)\}$, with their membership degrees reflecting their relevance as results of the query.

The term result set R_T can then be used to generate sets of relevant statements (R_S) and provenances (R_P) using look-ups in the corresponding indices as follows: (a) $R_S = \{(S_1, w_1^S), (S_2, w_2^S), \ldots, (S_m, w_m^S)\}$, where $w_i^S = \nu_s \sum_{j=1}^{n} w_j^T c_{j,i}$, (b) $R_P = \{(P_1, w_1^P), (P_2, w_2^P), \ldots, (P_q, w_q^P)\}$, where $w_i^P = \nu_p \sum_{j=1}^{m} w_j^S w_{j,i}$. ν_s, ν_p are normalisation constants for weights. The weight for a statement S_i in the result set R_S is computed as a normalised a dot product (*i.e.,* sum of the element-wise products) of the vectors given by: (a) the membership degrees in the term result set R_T, and (b) the column in the statement index that corresponds to S_i. Similarly, the weight for a provenance P_i in the result set R_P is a normalised dot product of the vectors given by the S_T membership degrees and the column in the provenance index corresponding to P_i.

The fuzzy membership degrees in the term, statement and provenance result sets can be used for ranking and visualisation, prioritising the most important results when presenting them to the user. The following example outlines how a specific query is evaluated.

Example 5 *Let us assume we want to query the full SKIMMR knowledge base about Parkinson's Disease for the following:*

? ↔ `parkinsonism` AND (? ↔ `mrpi` OR ? ↔ `magnetic resonance parkinsonism index`)

This aims to find all statements (and corresponding documents) that are related to `parkinsonism` *and either* `magnetic resonance parkinsonism index` *or its* `mrpi` *abbreviation. First of all, the full-text index is queried, retrieving two different terms conforming to the first atomic part of the query due to its stemming features:* `parkinsonism` *and* `parkinsonisms`*. The other two atomic parts of the initial query are resolved as is. After the look-up in the term index, four fuzzy sets are retrieved: 1.* $T_{\texttt{parkinsonism}}$ *(3,714 results), 2.* $T_{\texttt{parkinsonisms}}$ *(151 results), 3.* $T_{\texttt{mrpi}}$ *(39 results). 4.* $T_{\texttt{magnetic resonance parkinsonism index}}$ *(29 results). The set of terms conforming to the query is then computed as*

$$(T_{\texttt{parkinsonism}} \cup T_{\texttt{parkinsonisms}}) \cap (T_{\texttt{mrpi}} \cup T_{\texttt{magnetic resonance parkinsonism index}}).$$

When using maximum and minimum as t-conorm and t-norm for computing the fuzzy union and intersection (Hájek, 1998), respectively, the resulting set has 29 elements with non-zero membership degrees. The top five of them are

(1) `cup`*, (2)* `mrpi`*, (3)* `magnetic resonance parkinsonism index`*, (4)* `clinically unclassifiable parkinsonism`*, (5)* `clinical evolution`

with membership degrees 1.0, 1.0, 0.704, 0.39, 0.34, *respectively. According to the statement index, there are* 138 *statements corresponding to the top five term results of the initial query, composed of* 136 *co-occurrences and 2 similarities. The top five co-occurrence statements and the two similarity statements are:*

Type	$Entity_1$	$Entity_2$	Membership degree
cooc	`mrpi`	`cup`	1.0
cooc	`mrpi`	`magnetic resonance parkinsonism index`	0.852
cooc	`cup`	`magnetic resonance parkinsonism index`	0.852
cooc	`mrpi`	`clinically unclassifiable parkinsonism`	0.695
cooc	`cup`	`clinically unclassifiable parkinsonism`	0.695
sim	`psp patients`	`magnetic resonance parkinsonism index`	0.167
sim	`parkinsonism`	`clinical evolution`	0.069

where the membership degrees are computed from the combination of the term weights as described before the example, using an arithmetic mean for the aggregation. Finally, a look-up in the source index for publications corresponding to the top seven result statements retrieves 8 relevant PubMed identifiers (PMID). The top five of them correspond to the following list of articles:

PMID	Title	Authors	Weight
21832222	The diagnosis of neurodegenerative disorders based on clinical and pathological findings using an MRI approach	Watanabe H et al.	1.0
21287599	MRI measurements predict PSP in unclassifiable parkinsonisms: a cohort study	Morelli M et al.	0.132
22277395	Accuracy of magnetic resonance parkinsonism index for differentiation of progressive supranuclear palsy from probable or possible Parkinson disease	Morelli M et al.	0.005
15207208	Utility of dopamine transporter imaging (123-I Ioflupane SPECT) in the assessment of movement disorders	Garcia Vicente AM et al.	0.003
8397761	Alzheimer's disease and idiopathic Parkinson's disease coexistence	Rajput AH et al.	0.002

where the weights have been computed by summing up the statement set membership degrees multiplied by the source index weights and then normalising the values by their maximum.

Evaluation methodology

In addition to proposing specific methods for creating knowledge bases that support skim reading, we have also come up with a specific methodology for evaluating the generated knowledge bases. An ideal method for evaluating the proposed approach, implemented as a SKIMMR tool, would be to record and analyse user feedback and behaviour via SKIMMR instances used by large numbers of human experts. We do have such means for evaluating SKIMMR implemented in the user interface.[7] However, we have not yet managed to collect sufficiently large sample of user data due to the early stage of the prototype deployment. Therefore we implemented an indirect methodology for automated quantitative evaluation of SKIMMR instances using publicly available manually

[7] See the SMA SKIMMR instance at http://www.skimmr.org:8008/data/html/trial.tmp for details.

curated data. The methodology is primarily based on simulation of various types of human behaviour when browsing the entity networks generated by SKIMMR. We formally define certain properties of the simulations and measure their values in order to determine the utility of the entity networks for the purposes of skim reading. Details are given in the following sections.

Overview of the evaluation methods

The proposed methods intend to simulate human behaviour when using the data generated by SKIMMR. We apply the same simulations also to baseline data that can serve for the same or similar purpose as SKIMMR (i.e., discovery of new knowledge by navigating entity networks). Each simulation is associated with specific measures of performance, which can be used to compare the utility of SKIMMR with respect to the baseline.

The primary evaluation method is based on random walks (*Lovász, 1993*) in an undirected entity graph corresponding to the SKIMMR knowledge base. For the baseline, we use a network of MeSH terms assigned by human curators to the PubMed abstracts that have been used to create the SKIMMR knowledge base.[8] This represents a very similar type of content, i.e., entities associated with PubMed articles. It is also based on expert manual annotations and thus supposed to be a reliable gold standard (or at least a decent approximation thereof due to some level of transformation necessary to generate the entity network from the annotations).

[8] MeSH (Medical Subject Headings) is a comprehensive, manually curated and regularly updated controlled vocabulary and taxonomy of biomedical terms. It is frequently used as a standard for annotation of biomedical resources, such as PubMed abstracts. See http://www.ncbi.nlm.nih.gov/mesh for details.

Example 6 *Returning to the knowledge base statement from Example 2 in 'Corpus-wide co-occurrence':* (`parkinsonism`, `cooc`, `DRD`, `0.545`). *In the SKIMMR entity graph, this corresponds to two nodes* (`parkinsonism`, `DRD`) *and one edge between them with weight 0.545. We do not distinguish between the types of the edges (i.e., co-occurrence or similarity), since it is not of significant importance for the SKIMMR users according to our experience so far (they are more interested in navigating the connections between nodes regardless of the connection type).*

A baseline entity graph is generated from the PubMed annotations with MeSH terms. For all entities X, Y associated with an abstract A, we construct an edge connecting the nodes X and Y in the entity graph. The weight is implicitly assumed to be 1.0 for all such edges. To explain this using concrete data, let us consider the two PubMed IDs from Example 1, 9629849 and 8239569. Selected terms from the corresponding MeSH annotations are {`Parkinson` `Disease/radionuclide` `imaging`, `Male`, `Child`}, {`Parkinson` `Disease/radionuclide` `imaging`, `Dystonia/drug` `therapy`}, *respectively. The graph induced by these annotations is depicted in Fig. 1.*

The secondary evaluation method uses an index of related articles derived from the entities in the SKIMMR knowledge bases. For the baseline, we use either an index of related articles produced by a specific service of PubMed (*Lin & Wilbur, 2007*), or the evaluation data from the document categorisation task of the TREC'04 genomics track (*Cohen & Hersh, 2006*) where applicable. We use the TREC data since they were used also for evaluation of the actual algorithm used by PubMed to compute related articles.

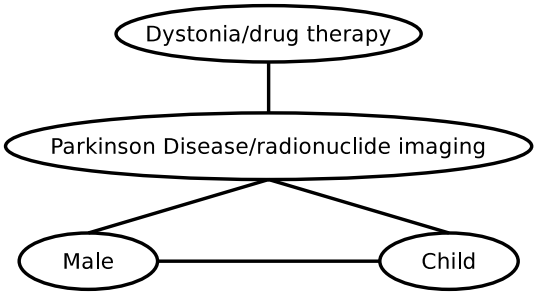


Figure 1 **Example of an entity graph derived from PubMed.**

To generate the index of related articles from the SKIMMR data, we first use the knowledge base indices (see 'Extracting basic co-occurrence statements from texts') to generate a mapping $E_P : E \rightarrow 2^P$ from entities from a set E to a set of corresponding provenance identifiers (subsets of a set P). In the next step, we traverse the entity graph G_E derived from the statements in the SKIMMR knowledge base and build an index of related articles according to the following algorithm:

1. Initialise a map M_P between all possible (P_i, P_j) provenance identifier pairs and the weight of an edge between them so that all values are zero.
2. For all pairs of entities E_1, E_n (i.e., nodes in G_E), do:
   - If there is a path $\mathcal{P}$ of edges $\{(E_1, E_2), (E_2, E_3), \ldots, (E_{n-1}, E_n)\}$ in G_E:
     - compute an aggregate weight of the path as $w_{\mathcal{P}} = w_{E_1,E_2} \cdot w_{E_2,E_3} \cdot \ldots \cdot w_{E_{n-1},E_n}$ (as a multiplication of all weights along the path $\mathcal{P}$);
     - set the values $M_P(P_i, P_j)$ of the map M_P to $max(M_P(P_i, P_j), w_{\mathcal{P}})$ for every P_i, P_j such that $P_i \in E_P(E_1), P_j \in E_P(E_n)$ (i.e., publications corresponding to the source and target entities of the path).
3. Interpret the M_P map as an adjacency matrix and construct a corresponding weighted undirected graph G_P.
4. For every node P in G_P, iteratively construct the index of related articles by associating the key P with a list L of all neighbours of P in G_P sorted by the weights of the corresponding edges.

Note that in practice, we restrict the maximum length of the paths to three and also remove edges in G_P with weight below 0.1. This is to prevent a combinatorial explosion of the provenance graph when the entity graph is very densely connected.

The baseline index of related publications according to the PubMed service is simply a mapping of one PubMed ID to an ordered list of the related PubMed IDs. The index based on the TREC data is generated from the article categories in the data set. For a PubMed ID X, the list of related IDs are all IDs belonging to the same category as X, ordered so that the definitely relevant articles occur before the possibly relevant ones.[9]

[9] The articles in the TREC data set are annotated by membership in a number of specific categories. The membership is gradual, with three possible values–definitely relevant, possibly relevant and not relevant.

Motivation of the evaluation methods

The random walks are meant to simulate user's behaviour when browsing the SKIMMR data, starting with an arbitrary entry point, traversing a number of edges linking the entities and ending up in a target point. Totally random walk corresponds to when a user browses randomly and tries to learn something interesting along the way. Other types of user behaviour can be simulated by introducing specific heuristics for selection of the next entity on the walk (see below for details). To determine how useful a random walk can be, we measure properties like the amount of information along the walk and in its neighbourhood, or semantic similarity between the source and target entities (i.e., how semantically coherent the walk is).

The index of related articles has been chosen as a secondary means for evaluating SKIMMR. Producing links between publications is not the main purpose of our current work, however, it is closely related to the notion of skim reading. Furthermore, there are directly applicable gold standards we can use for automated evaluation of the lists of related articles generated by SKIMMR, which can provide additional perspective on the utility of the underlying data even if we do not momentarily expose the publication networks to users.

Running and measuring the random walks

To evaluate the properties of random walks in a comprehensive manner, we ran them in batches with different settings of various parameters. These are namely: (1) *heuristics* for selecting the next entity (one of the four defined below); (2) *length* of the walk (2, 5, 10 or 50 edges); (3) *radius* of the walk's envelope, i.e., the maximum distance between the nodes of the path and entities that are considered its neighbourhood (0, 1, 2); (4) number of *repetitions* (100-times for each combination of the parameter (1–3) settings).

Before we continue, we have to introduce few notions that are essential for the definition of the random walk heuristics and measurements. The first of them is a set of top-level (abstract) clusters associated with an entity in a graph (either from SKIMMR or from PubMed) according to the MeSH taxonomy. This is defined as a function $C_A : E \to M$, where E, M are the sets of entities and MeSH cluster identifiers, respectively. The second notion is a set of specific entity cluster identifiers C_S, defined on the same domain and range as C_A, *i.e.*, $C_S : E \to M$.

The MeSH cluster identifiers are derived from the tree path codes associated with each term represented in MeSH. The tree path codes have the form $L_1.L_2.\ \ldots\ .L_{n-1}.L_n$ where L_i are sub-codes of increasing specificity (i.e., L_1 is the most general and L_n most specific). For the abstract cluster identifiers, we take only the top-level tree path codes into account as the values of C_A, while for C_S we consider the complete codes. Note that for the automatically extracted entity names in SKIMMR, there are often no direct matches in the MeSH taxonomy that could be used to assign the cluster identifiers. In these situations, we try to find a match for the terms and their sub-terms using a lemmatised full-text index implemented on the top of MeSH. This helps to increase the coverage two- to three-fold on our experimental data sets.

For some required measures, we will need to consider the number and size of specific clusters associated with the nodes in random walks and their envelopes. Let us assume a set of entities $Z \subseteq E$. The number of clusters associated with the entities from Z, $cn(Z)$, is then defined as $cn(Z) = |\bigcup_{X \in Z} C(X)|$ where C is one of C_A, C_S (depending on which type of clusters are we interested in). The size of a cluster $C_i \in C(X)$, $cs(C_i)$, is defined as an absolute frequency of the mentions of C_i among the clusters associated with the entities in Z. More formally, $cs(C_i) = |\{X | X \in Z \wedge C_i \in C(X)\}|$. Finally, we need a MeSH-based semantic similarity of entities $sim_M(X, Y)$, which is defined in detail in the formula (8) in 'Similarities'.

Example 7 *To illustrate the MeSH-based cluster annotations and similarities, let us consider two entities,* `supranuclear palsy, progressive, 3` *and* `secondary parkinson disease`*. The terms correspond to the MeSH tree code sets* {`C10.228.662.700`, ..., `C23.888.592.636.447.690`, ..., `C11.590.472.500`, ...} *and* {`C10.228.662.600.700`}*, respectively, which are also the sets of specific clusters associated with the terms. The top-level clusters are* {`C10`, `C11`, `C23`} *and* {`C10`}*, respectively. The least common subsumer of the two terms is* `C10.228.662` *of depth 3 (the only possibility with anything in common is* `C10.228.662.700` *and* `C10.228.662.600.700`*). The depths of the related cluster annotations are 4 and 5, therefore the semantic similarity is* $\frac{2 \cdot 3}{4+5} = \frac{2}{3}$.

We define four heuristics used in our random walk implementations. All the heuristics select the next node to visit in the entity graph according to the following algorithm:

1. Generate the list L of neighbours of the current node.
2. Sort L according to certain criteria (heuristic-dependent).
3. Initialise a threshold e to e_i, a pre-defined number in the $(0, 1)$ range (we use 0.9 in our experiments).
4. For each node u in the sorted list L, do:
   - Generate a random number r from the $[0, 1]$ range.
   - If $r \leq e$:
     - return u as the next node to visit.
   - Else:
     - set e to $e \cdot e_i$ and continue with the next node in L.
5. If nothing has been selected by now, return a random node from L.

All the heuristics effectively select the nodes closer to the head of the sorted neighbour list more likely than the ones closer to the tail. The random factor is introduced to emulate the human way of selecting next nodes to follow, which is often rather fuzzy according to our observations of sample SKIMMR users.

The distinguishing factor of the heuristics are the criteria for sorting the neighbour list. We employed the following four criteria in our experiments: (1) giving preference to the nodes that have not been visited before ($H = 1$); (2) giving preference to the nodes connected by edges with higher weight ($H = 2$); (3) giving preference to the nodes that are

more similar, using the sim_M function introduced before ($H = 3$); (4) giving preference to the nodes that are less similar ($H = 4$). The first heuristic simulates a user that browses the graph more or less randomly, but prefers to visit previously unknown nodes. The second heuristic models a user that prefers to follow a certain topic (i.e., focused browsing). The third heuristic represents a user that wants to learn as much as possible about many diverse topics. Finally, the fourth heuristic emulates a user that prefers to follow more plausible paths (approximated by the weight of the statements computed by SKIMMR).

Each random walk and its envelope (i.e., the neighbourhood of the corresponding paths in the entity graphs) can be associated with various information-theoretic measures, graph structure coefficients, levels of correspondence with external knowledge bases, etc. Out of the multitude of possibilities, we selected several specific scores we believe to soundly estimate the value of the underlying data for users in the context of skim reading.

Firstly, we measure **semantic coherence** of the walks. This is done using the MeSH-based semantic similarity between the nodes of the walk. In particular, we measure: (A) coherence between the source S and target T nodes as $sim_M(S, T)$; (B) product coherence between all the nodes $U_1, U_2, \ldots, U_n$ of the walk as $\Pi_{i\in\{1,\ldots,n-1\}} sim_M(U_i, U_{i+1})$; (C) average coherence between all the nodes $U_1, U_2, \ldots, U_n$ of the walk as $\frac{1}{n}\sum_{i\in\{1,\ldots,n-1\}} sim_M(U_i, U_{i+1})$. This family of measures helps us to assess how convergent (or divergent) are the walks in terms of focus on a specific topic.

The second measure we used is the **information content** of the nodes on and along the walks. For this, we use the entropy of the association of the nodes with clusters defined either (a) by the MeSH annotations or (b) by the structure of the envelope. By definition, the higher the entropy of a variable, the more information the variable contains (*Shannon, 1948*). In our context, a high entropy value associated with a random walk means that there is a lot of information available for the user to possibly learn when browsing the graph. The specific entropy measures we use relate to the following sets of nodes: (D) abstract MeSH clusters, path only; (E) specific MeSH clusters, path only; (F) abstract MeSH clusters, path and envelope; (G) specific MeSH clusters, path and envelope; (H) clusters defined by biconnected components (*Hopcroft & Tarjan, 1973*) in the envelope.[10] The entropies of the sets (D–G) are defined by formulae (9) and (10) in 'Entropies'.

[10] Biconnected components can be understood as sets of nodes in a graph that are locally strongly connected and therefore provide us with a simple approximation of clustering in the entity graphs based purely on their structural properties.

The last family of random walk evaluation measures is based on the **graph structure** of the envelopes: (I) envelope size (in nodes); (J) envelope size in biconnected components; (K) average component size (in nodes); (L) envelope's clustering coefficient. The first three measures are rather simple statistics of the envelope graph. The clustering coefficient is widely used as a convenient scalar representation of the structural complexity of a graph, especially in the field of social network analysis (*Carrington, Scott & Wasserman, 2005*). In our context, we can see it as an indication of how likely it is that the connections in the entity graph represent non-trivial relationships.

To facilitate the interpretation of the results, we computed also the following auxiliary measures: (M) number of abstract clusters along the path; (N) average size of the abstract clusters along the path; (O) number of abstract clusters in the envelope; (P) average size of the abstract clusters in the envelope; (Q) number of specific clusters along the path;

(R) average size of the specific clusters along the path; (S) number of specific clusters in the envelope; (T) average size of the specific clusters in the envelope. Note that all the auxiliary measures use the MeSH cluster size and number notions, i.e., *cs*(…) and *cn*(…) as defined earlier.

Comparing the indices of related articles

The indices of related articles have quite a simple structure. We can also use the baseline indices as gold standard, and therefore evaluate the publication networks implied by the SKIMMR data using classical measures of precision and recall (*Manning, Raghavan & Schütze, 2008*). Moreover, we can also compute correlation between the ranking of the items in the lists of related articles which provides an indication of how well SKIMMR preserves the ranking imposed by the gold standard.

For the correlation, we use the standard Pearson's formula (*Dowdy, Weardon & Chilko, 2005*), taking into account only the ranking of articles occurring in both lists. The measures of precision and recall are defined using overlaps of the sets of related articles in the SKIMMR and gold standard indices. The detailed definitions of the specific notions of precision and recall we use are given in formulae (11) and (12) in 'Precision and recall'. The gold standard is selected depending on the experimental data set, as explained in the next section. In order to cancel out the influence of different average lengths of lists of related publications between the SKIMMR and gold standard indices, one can take into account only a limited number of the most relevant (i.e., top) elements in each list.

RESULTS

We have implemented the techniques described in the previous section as a set of software modules and provided them with a search and browse front-end. This forms a prototype implementation of SKIMMR, available as an open source software package through the GitHub repository (see 'Software packages' for details). We here describe the architecture of the SKIMMR software ('Architecture') and give examples on the typical use of SKIMMR in the domains of Spinal Muscular Atrophy and Parkinson's Disease ('Using SKIMMR'). 'Evaluation' presents an evaluation of the proposed approach to machine-aided skim reading using SKIMMR running on three domain-specific sets of biomedical articles.

Architecture

The SKIMMR architecture and data flow is depicted in Fig. 2. First of all, SKIMMR needs a list of PubMed identifiers (unique numeric references to articles indexed on PubMed) specified by the user of system administrator. Then it automatically downloads the abstracts of the corresponding articles and stores the texts locally. Alternatively, one can export results of a manual PubMed search as an XML file (using the *'send to file'* feature) and then use a SKIMMR script to generate text from that file. From the texts, a domain-specific SKIMMR knowledge base is created using the methods described in 'Extracting basic co-occurrence statements from texts' and 'Computing a knowledge base from the extracted statements'. The computed statements and their article provenance are then indexed as described in 'Indexing and querying the knowledge base'. This allows

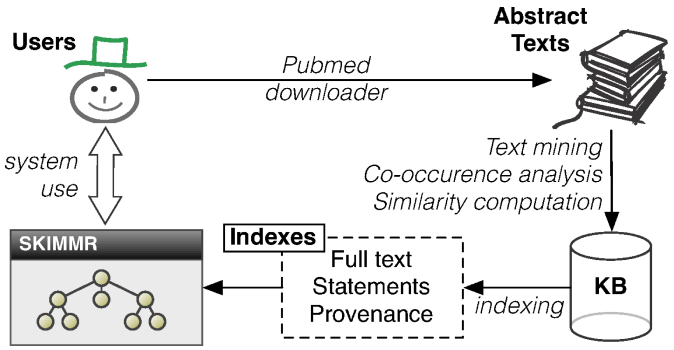


Figure 2 **Architecture of the SKIMMR system.**

users to search and browse the high-level graph summaries of the interconnected pieces of knowledge in the input articles. The degrees in the result sets (explained in detail in 'Indexing and querying the knowledge base') are used in the user interface to prioritise the more important nodes in the graphs by making their font and size proportional to the sum of the degrees of links (i.e., the number of statements) associated with them. Also, only a selected amount of the top scoring entities and links between them is displayed at a time.

Using SKIMMR

The general process of user interaction with SKIMMR can be schematically described as follows:

1. Search for an initial term of interest in a simple query text box.
2. A graph corresponding to the results of the search is displayed. The user has two options then:
   (a) Follow a link to another node in the graph, essentially browsing the underlying knowledge base along the chosen path by displaying the search results corresponding to the selected node and thus going back to step 1 above.
   (b) Display most relevant publications that have been used for computing the content of the result graph, going to step 3 below.
3. Access and study the displayed publications in detail using a re-direct to PubMed.

The following two sections illustrate the process using examples from two live instances of SKIMMR deployed on articles about Spinal Muscular Atrophy and Parkinson's Disease.[11] The last section of this part of the article gives a brief overview of the open source software packages of SKIMMR available for developers and users interested in deploying SKIMMR on their own data.

[11] The live instances are running at http://www.skimmr.org:8008 and http://www.skimmr.org:8090, respectively, as of June 2014. Canned back-up versions of them are available at http://www.skimmr.org/resources/skimmr/sma.tgz and http://www.skimmr.org/resources/skimmr/pd.tgz (SMA and Parkinson's Disease, respectively). If the SKIMMR dependencies are met (see https://github.com/vitnov/SKIMMR), the canned instances can be used locally on any machine with Python installed (versions higher than 2.4 and lower than 3.0 are supported, while 2.6.* and 2.7.* probably work best). After downloading the archives, unpack them and switch to the resulting folder. Run the re-indexing script, following Section 3.6 in the README provided in the same folder. To execute the SKIMMR front-end locally, run the server as described in Section 3.7 of the README.

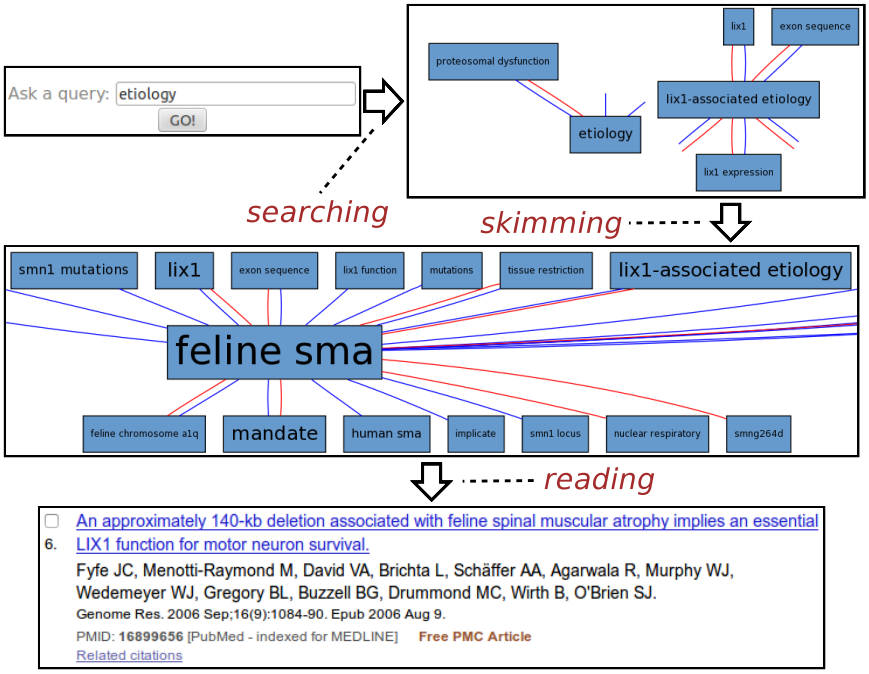


Figure 3 **Exploring SMA etiology.**

Spinal muscular atrophy

Fig. 3 illustrates a typical session with the Spinal Muscular Atrophy[12] instance of SKIMMR. The SMA instance was deployed on a corpus of 1,221 abstracts of articles compiled by SMA experts from the SMA foundation.[13]

The usage example is based on an actual session with Maryann Martone, a neuroscience professor from UCSD and a representative of the SMA Foundation who helped us to assess the potential of the SKIMMR prototype. Following the general template from the beginning of the section, the SMA session can be divided into three distinct phases:

1. **Searching:** The user was interested in the SMA etiology (studies on underlying causes of a disease). The key word `etiology` was thus entered into the search box.
2. **Skimming:** The resulting graph suggests relations between etiology of SMA, various gene mutations, and the Lix1 gene. Lix1 is responsible for protein expression in limbs which seems relevant to the SMA manifestation, therefore the `Lix1` — `associated` `etiology` path was followed in the graph, moving on to a slightly different area in the underlying knowledge base extracted from the SMA abstracts. When browsing the graph along that path, one can quickly notice recurring associations with `feline` `SMA`. According to the neuroscience expert we consulted, the cat models of the SMA disease appear to be quite a specific and interesting fringe area of SMA

[12] A genetic neurological disease caused by mutation of SMN1 gene that leads to death of motor neurons and consequent progressive muscle atrophy. It is the most common genetic cause of infant death and there is no cure as of now. See http://en.wikipedia.org/wiki/Spinal_muscular_atrophy for details.

[13] See http://www.smafoundation.org/.

research. Related articles may be relevant and enlightening even for experienced researchers in the field.

3. **Reading:** The reading mode of SKIMMR employs an in-line redirect to a specific PubMed result page. This way one can use the full set of PubMed features for exploring and reading the articles that are mostly relevant to the focused area of the graph the user skimmed until now. The sixth publication in the result was most relevant for our sample user, as it provided more details on the relationships between a particular gene mutation in a feline SMA model and the Lix1 function for motor neuron survival. This knowledge, albeit not directly related to SMA etiology in humans, was deemed as enlightening by the domain expert in the context of the general search for the culprits of the disease.

The whole session with the neuroscience expert lasted about two minutes and clearly demonstrated the potential for serendipitous knowledge discovery with our tool.

Parkinson's disease

Another example of the usage of SKIMMR is based on a corpus of 4,727 abstracts concerned with the clinical studies of Parkinson's Disease (PD). A sample session with the PD instance of SKIMMR is illustrated in Fig. 4. Following the general template from the beginning of the section, the PD session can be divided into three distinct phases again:

1. **Searching:** The session starts with typing `parkinson's` into the search box, aiming to explore the articles from a very general entry point.
2. **Skimming:** After a short interaction with SKIMMR, consisting of few skimming steps (i.e., following a certain path in the underlying graphs of entities extracted from the PD articles), an interesting area in the graph has been found. The area is concerned with `Magnetic Resonance Parkinsons Index` (MRPI). This is a numeric score calculated by multiplying two structural ratios: one for the area of the pons relative to that of the midbrain and the other for the width of the Middle Cerebellar Peduncle relative to the width of the Superior Cerebellar Peduncle. The score is used to diagnose PD based on neuroimaging data (*Morelli et al., 2011*).
3. **Reading:** When displaying the articles that were used to compute the subgraph surrounding MRPI, the user reverted to actual reading of the literature concerning MRPI and related MRI measures used to diagnose Parskinson's Disease as well a range of related neurodegenerative disorders.

This example illustrates once again how SKIMMR provides an easy way of navigating through the conceptual space of a subject that is accessible even to novices, reaching interesting and well-specified components areas of the space very quickly.

Software packages

In addition to the two live instances described in the previous sections, SKIMMR is available for local installation and custom deployment either on biomedical article abstracts from PubMed, or on general English texts. Moreover, one can expose SKIMMR

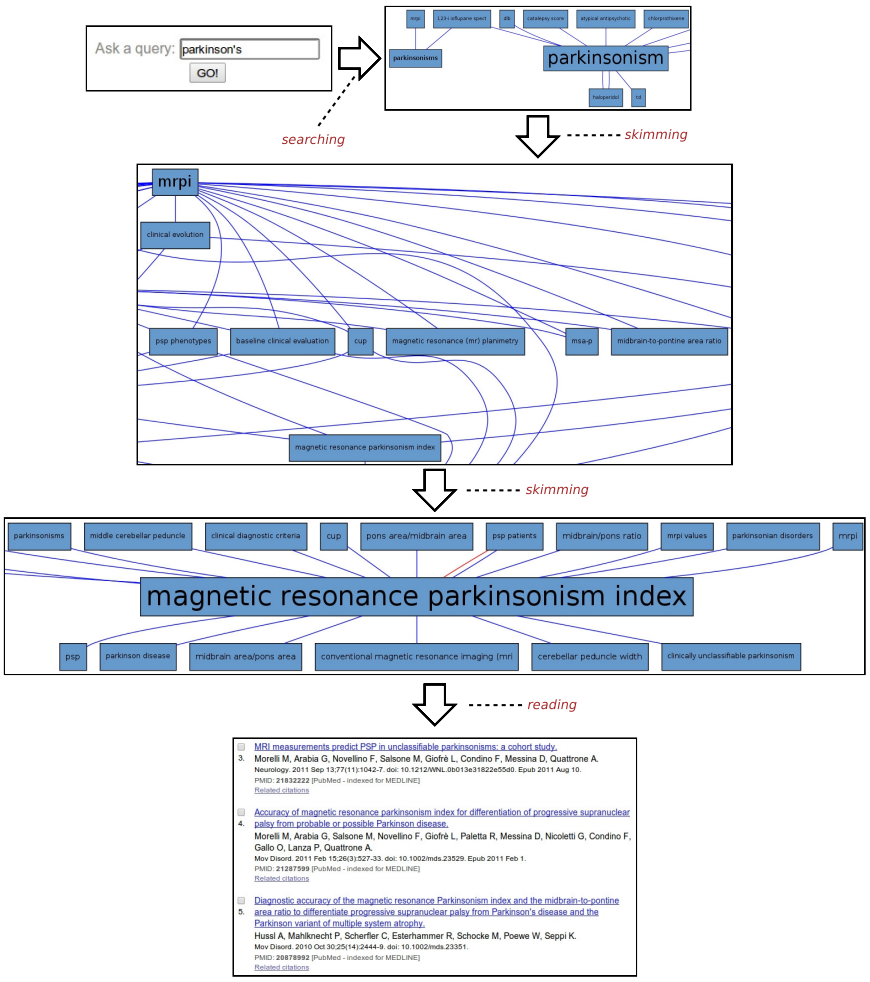


Figure 4 Exploring Parkinson's disease.

via a simple HTTP web service once the back-end has compiled a knowledge base from selected textual input. The latter is particularly useful for the development of other applications on the top of the content generated by SKIMMR. Open source development snapshots (written in the Python programming language) of SKIMMR modules are available via our GitHub repository[14] with accompanying documentation.

[14] See https://github.com/vitnov/SKIMMR.

Evaluation

In the following we report on experiments we used for evaluating SKIMMR using the method explained in 'Evaluation methodology'. The results of our experiments empirically demonstrate that the SKIMMR networks allow for more focused browsing

of the publication content than is possible with tools like PubMed. SKIMMR also has the potential for offering more information of higher complexity during the browsing process. The following sections provide details on the data sets used in the experiments and the results of the evaluation.

Evaluation data

We have evaluated SKIMMR using three corpora of domain-specific biomedical articles. The first one was SMA: a representative corpus of 1,221 PubMed abstracts dealing with Spinal Muscular Atrophy (SMA), compiled by experts from SMA Foundation. The second corpus was PD: a set of 4,727 abstracts that came as results (in February 2013) of a search for clinical studies on Parkinson's Disease on PubMed. The last corpus was TREC: a random sample[15] of 2,247 PubMed abstracts from the evaluation corpus of the TREC'04 genomics track (document categorisation task).

[15] We processed only a subset of the experimental data available from TREC so that the experimental knowledge bases are of a size within similar range of hundreds of thousands of statements.

For running the experiment with random walks, we generated two graphs for each of the corpora (using the methods described in Example 6): (1) network of SKIMMR entities; (2) network of MeSH terms based on the PubMed annotations of the articles that were used as sources for the particular SKIMMR instance.

As outlined before in the methods section, we also used some auxiliary data structures for the evaluation. The first auxiliary resource was the MeSH thesaurus (version from 2013). From the data available on the National Library of Medicine web site, we generated a mapping from all MeSH terms and their synonyms to the corresponding tree codes indicating their position in the MeSH hierarchy. We also implemented a lemmatised full-text index on the MeSH mapping keys to increase the coverage of the tree annotations when the extracted entity names do not exactly correspond to the MeSH terms.

The second type of auxiliary resource (a gold standard) were indices of related articles based on the corresponding PubMed service. For the other type of gold standard, we used the TREC'04 category associations from the genomics track data. This is essentially a mapping between PubMed IDs, category identifiers and a degree of membership of the specific IDs in the category (definitely relevant, possibly relevant, not relevant). From that mapping, we generated the index of related articles as a gold standard for the secondary evaluation method (the details of the process are described in the previous section).

Note that for the TREC corpus, the index of related articles based on the TREC data is applicable as a gold standard for the secondary evaluation. However, for the other two data sets (SMA and PD), we used the gold standard based on the PubMed service for fetching related articles. This is due to almost zero overlap between the TREC PubMed IDs and the SMA, PD corpora, respectively.

Data statistics

Corpus and knowledge base statistics. Basic statistics of the particular text corpora are given in Table 1, with column explanations as follows: (1) $|SRC|$ is the number of the source documents; (2) $|TOK|$ is the number of tokens (words) in the source documents; (3) $|BC|$ is the number of base co-occurrence statements extracted from the sources (see 'Extracting basic co-occurrence statements from texts' for details); (4) $|LEX|$ is the vocabulary size

Table 1 Basic statistics of the SKIMMR instances.

Data set ID	$\|SRC\|$	$\|TOK\|$	$\|BC\|$	$\|LEX\|$	$\|KB_{cooc}\|$	$\|KB_{sim}\|$
SMA	1,221	223,257	333,124	15,288	308,626	23,167
PD	4,727	943,444	1,096,037	43,410	965,753	57,876
TREC	2,247	439,202	757,762	39,431	745,201	65,510

Table 2 Derived statistics of the SKIMMR instances.

Data set ID	T/S	B/S	L/T	SM/KB	KB/S	KB/L
SMA	182.848	272.829	0.068	0.07	271.739	21.703
PD	199.586	231.867	0.046	0.057	216.549	23.58
TREC	195.462	337.233	0.09	0.081	360.797	20.56

(i.e., the number of unique entities occurring in the basic co-occurrence statements); (5) $|KB_{cooc}|$ is the number of aggregate co-occurrence statements in the corresponding SKIMMR knowledge base (see 'Corpus-wide co-occurrence'); (6) $|KB_{sim}|$ is the number of similarity statements in the corresponding SKIMMR knowledge base (see 'Similarity').

Derived statistics on the SKIMMR instances are provided in Table 2, with column explanations as follows: (1) T/S is an average number of tokens per a source document; (2) B/S is an average number of basic co-occurrence statements per a source document; (3) L/T is a ratio of the size of the lexicon with respect to the overall number of tokens in the input data; (4) SM/KB is a ratio of the similarity statements to the all statements in the knowledge base; (5) KB/S is an average number of statements in the knowledge base per a source document; (6) KB/L is an average number of statements in the knowledge base per a term in the lexicon. The values in the columns are computed from the basic statistics as follows:

$$T/S = \frac{|TOK|}{|SRC|}, \quad B/S = \frac{|BC|}{|SRC|}, \quad L/T = \frac{|LEX|}{|TOK|}, \quad SM/KB = \frac{|KB_{sim}|}{|KB_{sim}| + |KB_{cooc}|},$$
$$KB/S = \frac{|KB_{sim}| + |KB_{cooc}|}{|SRC|}, \quad KB/L = \frac{|KB_{sim}| + |KB_{cooc}|}{|LEX|}.$$

The statistics of the data sets are relatively homogeneous. The TREC data contains more base co-occurrence statements per article, and has an increased ratio of (unique) lexicon terms per absolute number of (non-unique) tokens in the documents. TREC knowledge base also contains more statements per article than the other two, but the ratios of number of statements in it per lexicon term are more or less balanced. We believe that the statistics do not imply the need to treat each of the data sets differently when interpreting the results reported in the next section.

Graph statistics. The statistics of the graph data that are utilised in the random walks experiment are given in Tables 3 and 4 for PubMed and SKIMMR, respectively. The specific

Table 3 Statistics of the PubMed graphs for random walks.

Data set ID	$\|V\|$	$\|E\|$	$\frac{\|E\|}{\|V\|}$	D	d	l_G	$\|C\|$
SMA	5,364	78,608	14.655	$5.465 \cdot 10^{-3}$	5.971	3.029	2
PD	8,622	133,188	15.447	$3.584 \cdot 10^{-3}$	6	2.899	2
TREC	10,734	161,838	15.077	$2.809 \cdot 10^{-3}$	7.984	3.146	3

Table 4 Statistics of the SKIMMR graphs for random walks.

Data set ID	$\|V\|$	$\|E\|$	$\frac{\|E\|}{\|V\|}$	D	d	l_G	$\|C\|$
SMA	15,287	305,077	19.957	$2.611 \cdot 10^{-3}$	5	2.642	1
PD	43,411	952,296	21.937	$1.011 \cdot 10^{-3}$	5	2.271	2
TREC	37,184	745,078	20.038	$1.078 \cdot 10^{-3}$	5.991	2.999	12

statistics provided on the graphs are: (1) number of nodes ($|V|$); (2) number of edges[16] ($|E|$); (3) average number of edges per a node ($\frac{|E|}{|V|}$); (4) density ($D = \frac{2 \cdot |E|}{|V|(|V|-1)}$, i.e., a ratio of the actual bidirectional connections between nodes relative to the maximum possible number of connections); (5) diameter (d, computed as an arithmetic mean of the longest possible paths in the connected components of the graph, weighted by the size of the components in nodes); (6) average shortest path length (l_G, computed similarly to d as an average weighted mean of the value for each connected component); (7) number of connected components ($|C|$).

The statistics demonstrate that the SKIMMR graphs are larger and have higher absolute number of connections per a node, but are less dense than the PubMed graphs. All the graphs exhibit the "small-world" property (*Watts & Strogatz, 1998*), since the graphs have small diameters and there are also very short paths between the connected nodes despite the low density and relatively large size of the graphs.

Auxiliary data statistics. The MeSH data contained 719,877 terms and 54,935 tree codes, with ca. 2.371 tree code annotations per term in average. The statistics of the indices of related publications for SKIMMR and for gold standards are provided in Table 5. We provide values for the size of the index in numbers of publications covered ($|P|$) and an average number of related publications associated with each key ($\bar{R}$). The average length of the lists of related publications is much higher for all three instances of SKIMMR. This is a result of the small-world property of the SKIMMR networks which makes most of the publications connected with each other (although the connections mostly have weights close to zero).

Evaluation results

In the following we report on the results measured using the specific SKIMMR knowledge bases and corresponding baseline data. Each category of the evaluation measures is covered

[16] Note that the number of edges is lower in the SKIMMR graphs than in the corresponding SKIMMR knowledge bases due to the fact that we do not distinguish between the different relationships. Therefore, if two nodes are connected by more than one statements, there is still only one edge for those nodes in the graph.

Table 5 Statistics of the indices of related publications.

	Gold standard		SKIMMR	
Data set ID	$\|P\|$	$\bar{R}$	$\|P\|$	$\bar{R}$
SMA	1,221	36.15	1,220	959.628
PD	4,727	28.61	4,724	4327.625
TREC	434	18.032	2,245	1251.424

in a separate section. Note that we mostly provide concise plots and summaries of the results here in the article, however, full results can be found online (Data Deposition).

Semantic coherence. Figure 5 shows the values of the aggregated semantic coherence measures (i.e., source-target coherence, product path coherence and average path coherence) for the PD, SMA and TREC data sets. The values were aggregated by computing their arithmetic means and are denoted by the *y*-axis of the plots. The *x*-axis corresponds to different combinations of the heuristics and path lengths for the execution of the random walks (as the coherence does not depend on the envelope size, this parameter is zero all the time in this case).[17] The combinations are grouped by heuristics (random preference, weight preference, similarity preference, dissimilarity preference from left to right). The path length parameter increases from left to right for each heuristic group on the *x*-axis. The green line is for the SKIMMR results and the blue line is for the PubMed baseline.

For any combination of the random walk execution parameters, SKIMMR outperforms the baseline by quite a large relative margin. The most successful heuristic in terms of coherence is the one that prefers more similar nodes to visit next (third quarter of the plots), and the coherence is generally lower for longer paths, which are all observations corresponding to intuitive assumptions.

Information content. Figure 6 shows the values of the arithmetic mean of all types of information content measures for the particular combinations of the random walk execution parameters (including also envelope sizes in increasing order for each heuristic). Although the relative difference is not as significant as in the semantic coherence case, SKIMMR again performs consistently better than the baseline. There are no significant differences between the specific heuristics. The information content increases with longer walks and larger envelopes, which is due to generally larger numbers of clusters occurring among more nodes involved in the measurement.

Graph Structure. Figure 7 shows the values of the clustering coefficient, again with green and blue lines for the SKIMMR and PubMed baseline results, respectively. SKIMMR exhibits larger level of complexity than the baseline in terms of clustering coefficient, with moderate relative margin in most cases. There are no significant differences between the particular walk heuristics. The complexity generally increases with the length of the path, but, interestingly enough, does not so with the size of the envelopes. The highest complexity is typically achieved for the longest paths without any envelope. We suspect

[17] The exact form of labels on the *x*-axis is a combination of heuristic (H), envelope diameter (E) and path length (L) parameters with their numeric identifiers (in case of heuristics) or values (for envelope size and path length). For instance, $\mathtt{H}=2.\mathtt{E}=1.\mathtt{L}=10$ stands for a measurement using the weight preference heuristic (identifier 2), envelope of diameter 1 and path of length 10.

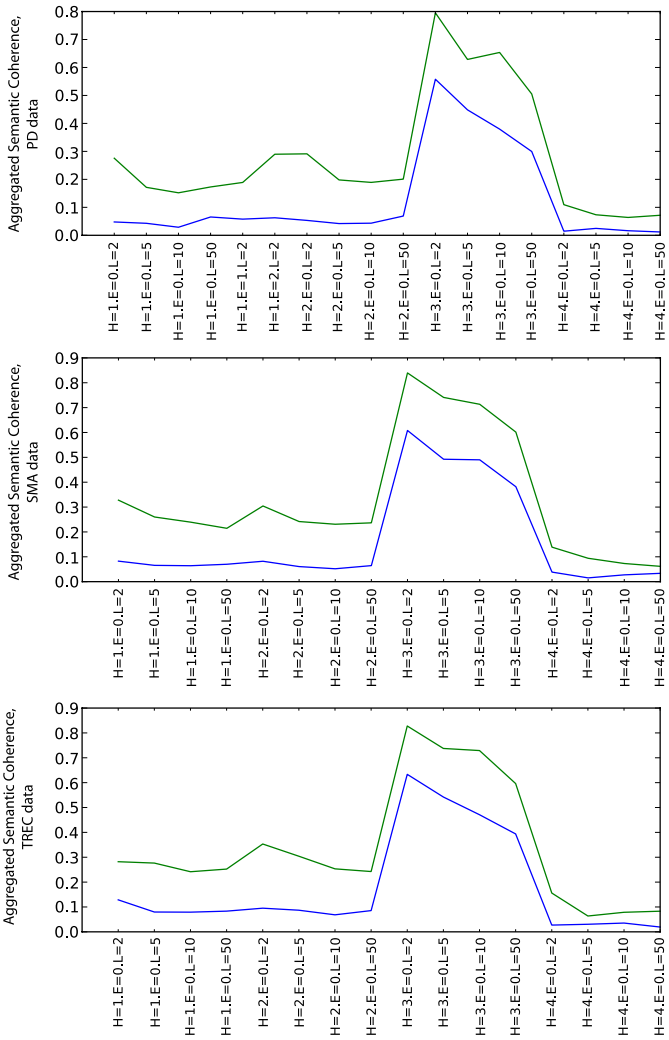


Figure 5 **Aggregated semantic coherence (blue: PubMed, green: SKIMMR).**

this to be related to the small world property of the graphs–adding more nodes from the envelope may not contribute to the actual complexity due to making the graph much more "uniformly" dense and therefore less complex.

Auxiliary measures. The number of clusters associated with the nodes on the paths (measures M and Q) is always higher for SKIMMR than for the PubMed baseline. The number of clusters associated with the whole envelopes (measures O and S) is almost always higher for SKIMMR with few exceptions of rather negligible relative differences in

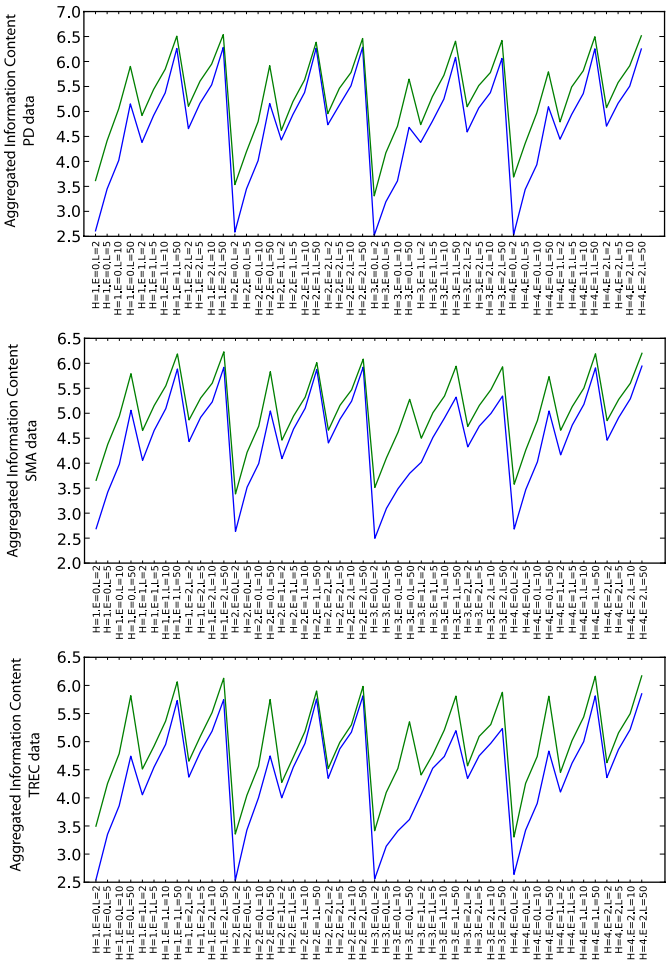


Figure 6 **Aggregated information content (blue: PubMed, green: SKIMMR).**

favour of the baseline. The average numbers of nodes per cluster on the path (measures N and R) are higher for SKIMMR except for the heuristic that prefers similar nodes to visit next. This can be explained by the increased likelihood of populating already "visited" clusters with this heuristic when traversing paths with lower numbers of clusters along them. Finally, the average number of nodes per cluster in the envelope (measures P and T) is higher for SKIMMR in most cases.

The general patterns observed among the auxiliary measure values indicates higher topical variability in the SKIMMR graphs, as there are more clusters that have generally higher cardinality than in the PubMed baselines. This is consistent with the observation of

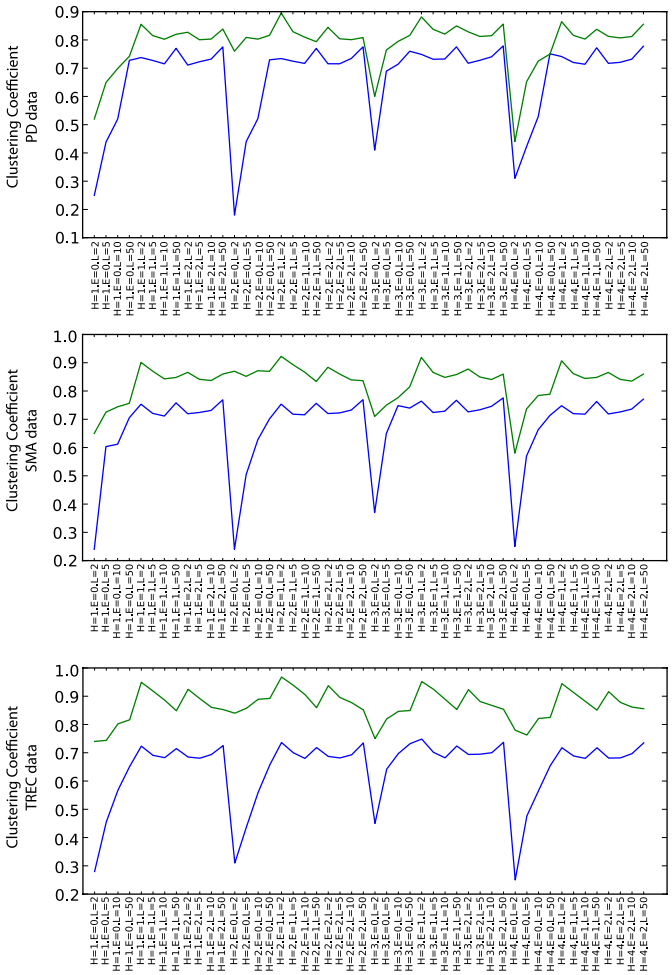


Figure 7 **Clustering coefficient (blue: PubMed, green: SKIMMR).**

the generally higher information content associated with the random walks in SKIMMR graphs.

Related articles. The results of the evaluation measures based on the lists of related articles generated by SKIMMR and by related baselines are summarised in Table 6. Note that as explained in 'Evaluation data', we used actual TREC evaluation data for the TREC dataset, while for PD and SMA, we used the related articles provided by PubMed due to negligible overlap with the TREC gold standard.

The pre_{avg} and rec_{avg} columns in Table 6 contain the precision and recall values for each data set, respectively, and the $C \geq 0.7$ contains the ratio of SKIMMR results that have

Table 6 **Results for the related articles.**

PD			SMA			TREC		
pre_{avg}	rec_{avg}	$C \geq 0.7$	pre_{avg}	rec_{avg}	$C \geq 0.7$	pre_{avg}	rec_{avg}	$C \geq 0.7$
0.0095	0.0240	0.5576	0.0139	0.0777	0.5405	0.0154	0.0487	0.5862

significant correlation (i.e., at least 0.7) with the corresponding baseline. The absolute values of the average precision and recall are very poor, in units of percents. The correlation results are more promising, showing that more than half of the related document rankings produced by SKIMMR are reasonably aligned with the gold standard. Moreover, the correlation is highest for the TREC data set based on the only gold standard that is manually curated.

DISCUSSION

SKIMMR provides a computational instantiation of the concept of 'skim reading.' In the early prototype stage, we generally focussed on delivering as much of the basic functionality as possible in a lightweight interface. Lacking enough representative data collected from ongoing user studies, we have designed a series of automated experiments to simulate several skim reading modes one can engage in with SKIMMR. We evaluated these experiments using gold standards derived from manually curated biomedical resources. Here we offer a discussion of the results in relation to the concept of machine-aided skim reading as realised by the SKIMMR prototype. The discussion is followed by an overview of related work and an outline of possible future directions.

Interpreting the results

The secondary evaluation using lists of related publications induced by the SKIMMR knowledge bases did not bring particularly good results in terms of precision and recall. However, the correlation with the related document ranking provided by baselines was more satisfactory. This indicates that with better methods for pruning the rather extensive lists of related publications produced with SKIMMR, we may be able to improve the precision and recall substantially. Still, this evaluation was indirect since generating lists of related publications is not the main purpose of SKIMMR. Apart from indirect evaluation, we were also curious whether the data produced by SKIMMR could not be used also for a rather different task straightaway. The lesson learned is that this may be possible, however, some post-processing of the derived publication lists would be required to make the SKIMMR-based related document retrieval more accurate for practical applications.

Our main goal was to show that our approach to machine-aided skim reading can be efficient in navigating high-level conceptual structures derived from large numbers of publications. The results of the primary evaluation experiment—simulations of various types of skimming behaviour by random walks—demonstrated that our assumption may indeed be valid. The entity networks computed by SKIMMR are generally more *semantically coherent*, more *informative* and more *complex* than similar networks based on

the manually curated PubMed article annotations. This means that users will typically be able to browse the SKIMMR networks in a more focused way. At the same time, however, they will learn more interesting related information from the context of the browsing path, and can also potentially gain additional knowledge from more complex relationships between the concepts encountered on the way. This is very promising in the context of our original motivations for the presented research.

Experiments with actual users would have brought many more insights regarding the practical relevance of the SKIMMR prototype. Still, the simulations we have proposed cover four distinct classes of possible browsing behaviour, and our results are generally consistent regardless of the particular heuristic used. This leads us to believe that the evaluation measures computed on paths selected by human users would not be radically different from the patterns observed within our simulations.

Related work

The text mining we use is similar to the techniques mentioned in *Yan et al. (2009)*, but we use a finer-grained notion of co-occurrence. Regarding biomedical text mining, tools like BioMedLEE (*Friedman et al., 2004*), MetaMap (*Aronson & Lang, 2010*) or SemRep (*Liu et al., 2012*) are closely related to our approach. The tools mostly focus on annotation of texts with concepts from standard biomedical vocabularies like UMLS which is very useful for many practical applications. However, it is relatively difficult to use the corresponding software modules within our tool due to complex dependencies and lack of simple APIs and/or batch scripts. The tools also lack the ability to identify concepts not present in the biomedical vocabularies or ontologies. Therefore we decided to use LingPipe's batch entity recogniser in SKIMMR. The tool is based on a relatively outdated GENIA corpus, but is very easy to integrate, efficient and capable of capturing unknown entities based on the underlying statistical model, which corresponds well to our goal of delivering a lightweight, extensible and easily portable tool for skim-reading.

The representation of the relationships between entities in texts is very close to the approach of *Baroni & Lenci (2010)*, however, we have extended the tensor-based representation to tackle a broader notion of text and data semantics, as described in detail in *Nováček, Handschuh & Decker (2011)*. The indexing and querying of the relationships between entities mentioned in the texts is based on fuzzy index structures, similarly to *Zadrozny & Nowacka (2009)*. We make use of the underlying distributional semantics representation, though, which captures more subtle features of the meaning of original texts.

Graph-based representations of natural language data have previously been generated using dependency parsing (*Ramakrishnan et al., 2008*; *Biemann et al., 2013*). Since these representations are derived directly from the parse structure, they are not necessarily tailored for the precise task of skim-reading but could provide a valuable intermediate representation. Another graph-based representation that is derived from the text of documents are similarity-based approaches derived from 'topic models' of document corpora (*Talley et al., 2011*). Although these analyses typically provide a visualization of the organization of documents, not of their contents, the topic modeling methods provide

statistical representation of the text that can then be leveraged to examine other aspects of the context of the document, such as its citations (*Foulds & Smyth, 2013*).

A broad research area of high relevance to the presented work is the field of 'Machine Reading' that can be defined as "*the autonomous understanding of text*" (*Etzioni, Banko & Cafarella, 2006*). It is an ambitious goal that has attracted much interest from NLP researchers (*Mulkar et al., 2007*; *Strassel et al., 2010*; *Poon & Domingos, 2010*). By framing the reading task as 'skimming' (which provides a little more structure than simply navigating a set of documents, but much less than a full representation of the semantics of documents), we hope to leverage machine reading principles into practical tools that can be used by domain experts straightforwardly.

Our approach shares some similarities with applications of spreading activation in information retrieval which are summarised for instance in the survey (*Crestani, 1997*). These approaches are based on associations between search results computed either off-line or based on the "live" user interactions. The network data representation used for the associations is quite close to SKIMMR, however, we do not use the spreading activation principle to actually retrieve the results. We let the users to navigate the graph by themselves which allows them to discover even niche and very domain-specific areas in the graph's structure that may not be reached using the spreading activation.

Works in literature based discovery using either semantic relationships (*Hristovski et al., 2006*) or corresponding graph structures (*Wilkowski et al., 2011*) are conceptually very similar to our approach to skim reading. However, the methods are quite specific when deployed, focusing predominantly on particular types of relationships and providing pre-defined schema for mining instances of the relationships from the textual data. We keep the process lightweight and easily portable, and leave the interpretation of the conceptual networks on the user. We do lose some accuracy by doing so, but the resulting framework is easily extensible and portable to a new domain within minutes, which provides for a broader coverage compensating the loss of accuracy.

From the user perspective, SKIMMR is quite closely related to GoPubMed (*Dietze et al., 2008*), a knowledge-based search engine for biomedical texts. GoPubMed uses Medical Subject Headings and Gene Ontology to speed up finding of relevant results by semantic annotation and classification of the search results. SKIMMR is oriented more on browsing than on searching, and the browsing is realised via knowledge bases inferred from the texts automatically in a bottom-up manner. This makes SKIMMR independent on any pre-defined ontology and lets users to combine their own domain knowledge with the data present in the article corpus.

Tools like DynaCat (*Pratt, 1997*) or QueryCat (*Pratt & Wasserman, 2000*) share the basic motivations with our work as they target the information overload problem in life sciences. They focus specifically on automated categorisation of user queries and the query results, aiming at increasing the precision of document retrieval. Our approach is different in that it focuses on letting users explore the content of the publications instead of the publications themselves. This provides an alternative solution to the information overload by leading

users to interesting information spanning across multiple documents that may not be grouped together by *Pratt (1997)* and *Pratt & Wasserman (2000)*.

Another related tool is Exhibit (*Huynh, Karger & Miller, 2007*), which can be used for faceted browsing of arbitrary datasets expressed in JSON (*Crockford, 2006*). Using Exhibit one can dynamically define the scope from which they want to explore the dataset and thus quickly focus on particular items of interest. However, Exhibit does not provide any solution on how to get the structured data to explore from possibly unstructured resources (such as texts).

Textpresso (*Müller, Kenny & Sternberg, 2004*) is quite similar to SKIMMR concerning searching for relations between concepts in particular chunks of text. However, the underlying ontologies and their instance sets have to be provided manually which often requires years of work, whereas SKIMMR operates without any such costly input. Moreover, the system's scale regarding the number of publications' full-texts and concepts covered is generally lower than the instances of SKIMMR that can be set up in minutes.

CORAAL (*Nováček et al., 2010*) is our previous work for cancer publication search, which extracts relations between entities from texts, based on the verb frames occurring in the sentences. The content is then exposed via a multiple-perspective search and browse interface. SKIMMR brings the following major improvements over CORAAL: (1) more advanced back-end (built using our distributional data semantics framework introduced in *Nováček, Handschuh & Decker, 2011*); (2) simplified modes of interaction with the data leading to increased usability and better user experience; (3) richer, more robust fuzzy querying; (4) general streamlining of the underlying technologies and front-ends motivated by the simple, yet powerful metaphor of machine-aided skim reading.

Future work

Despite the initial promising results, there is still much to do in order to realise the full potential of SKIMMR as a machine-aided skim reading prototype. First of all, we need to continue our efforts in recruiting coherent and reliable sample user groups for each of the experimental SKIMMR instances in order to complement the presented evaluation by results of actual user studies. Once we get the users' feedback, we will analyse it and try to identify significant patterns emerging from the tracked behaviour data in order to correlate them with the explicit feedback, usability assessments and the results achieved in our simulation experiments. This will provide us with a sound basis for the next iteration of the SKIMMR prototype development, which will reflect more representative user requirements.

Regarding the SKIMMR development itself, the most important things to improve are as follows. We need to extract more types of relations than just co-occurrence and rather broadly defined similarity. One example of domain specific complex relation are associations of potential side effects with drugs. Another, more general example, is taxonomical relations (super-concept, sub-concept), which may help provide additional perspective to browsing the entity networks (i.e., starting with high-level overview of the relations between more abstract concepts and then focusing on the structure of the

connections between more specific sub-concepts of selected nodes). Other improvements related to the user interface are: (1) smoother navigation in the entity networks (the nodes have to be active and shift the focus of the displayed graph upon clicking on them, they may also display additional metadata, such as summaries of the associated source texts); (2) support of more expressive (conjunctive, disjunctive, etc.) search queries not only in the back-end, but also in the front-end, preferably with a dedicated graphical user interface that allows to formulate the queries easily even for lay users; (3) higher-level visualisation features such as evolution of selected concepts' neighbourhoods in time on a sliding scale. We believe that realisation of all these features will make SKIMMR a truly powerful tool for facilitating knowledge discovery (not only) in life sciences.

ACKNOWLEDGEMENTS

We would like to thank to our former colleagues Eduard H. Hovy and Drashti Dave for their generously shared insights regarding the NLP and biomedical aspects, respectively, of the presented work. Last but not least, we are indebted to Maryann Martone for her guidance concerning the Spinal Muscular Atrophy domain and for multiple testing sessions during SKIMMR development which helped us to refine the tool in order to meet the actual requirements of life scientists.

APPENDIX. FORMULAE DEFINITIONS

In this appendix we give full account on definitions of some of the formal notions used throughout the main article but not covered in detail there.

Co-occurrences

The basic co-occurrence score $cooc((e_x, e_y), PubMed_{PMID})$ for two entities e_x, e_y in an article $PubMed_{PMID}$, introduced in 'Extracting basic co-occurrence statements from texts', is computed as

$$cooc((e_x, e_y), PubMed_{PMID}) = \sum_{i,j \in S(e_x, e_y)} \frac{1}{1 + |i - j|} \tag{1}$$

where $S(e_x, e_y)$ is a set of numbers of sentences that contain the entity e_x or e_y (assuming the sentences numbered sequentially from the beginning of the text). In practice, one may impose a limit on the maximum allowed distance of entities to be taken into account in the co-occurrence score computation (we disregard entities occurring more than 3 sentences apart from the score sum).

The non-normalised formula for corpus-wide co-occurrence for two outcomes (i.e., terms in our specific use case) x, y, using a base-2 logarithm (introduced in 'Corpus-wide co-occurrence'), is:

$$fpmi(x, y) = F(x, y) log_2 \frac{p(x, y)}{p(x)p(y)} \tag{2}$$

where $F(x, y)$ is the absolute frequency of the x, y co-occurrence and $p(x, y), p(x), p(y)$ are the joint and individual distributions, respectively. In our case, the distributions are the

weighted relative frequencies of the entity terms in the basic co-occurrence tuples generated from the input texts which are computed as follows. Let us assume a set T of tuples

$$t_1 = (e_{1,x}, e_{1,y}, cooc((e_{1,x}, e_{1,y}), PubMed_{PMID_1}), PubMed_{PMID_1}),$$
$$t_2 = (e_{2,x}, e_{2,y}, cooc((e_{2,x}, e_{2,y}), PubMed_{PMID_2}), PubMed_{PMID_2}),$$
$$\vdots$$
$$t_n = (e_{n,x}, e_{n,y}, cooc((e_{n,x}, e_{n,y}), PubMed_{PMID_n}), PubMed_{PMID_n})$$

as a result of the basic co-occurrence statement extraction described in the previous section. The joint distribution of terms x, y specific to our case can then be computed as:

$$p(x,y) = \frac{\sum_{w \in W(x,y,T)} w}{|T|} \tag{3}$$

where $W(x,y,T) = \{w | \exists e_1, e_2, w, i.(e_1, e_2, w, i) \in T \wedge ((e_1 = x \wedge e_2 = y) \vee (e_1 = y \wedge e_2 = x))\}$ is the set of weights in the basic co-occurrence tuples that contain both x, y as entity arguments. Finally, the individual distribution of a term z is computed as:

$$p(z) = \frac{\sum_{w \in W(z,T)} w}{|T|} \tag{4}$$

where $W(z,T) = \{w | \exists e_1, e_2, w, i.(e_1, e_2, w, i) \in T \wedge (e_1 = z \vee e_2 = z)\}$ is the set of weights in the basic co-occurrence tuples that contain z as any one of the entity arguments. In the eventual result, all co-occurrence tuples with score lower than zero are omitted, while the remaining ones are normalised as follows:

$$npmi(x,y) = \nu(fpmi(x,y), P) \tag{5}$$

where ν is a function that divides the scores by the P-th percentile of all the scores and truncates the resulting value to 1 if it is higher than that. The motivation for such definition of the normalisation is that using the percentile, one can flexibly reduce the influence of possibly disproportional distributions in the scores (i.e., when there are few very high values, normalisation by the sum of all values or by the maximal value would result in most of the final scores being very low, whereas the carefully selected percentile can balance that out, reducing only relatively low number of very high scores to crisp 1).

Similarities

Firstly we define the cosine similarity introduced in 'Similarity'. For that we need few auxiliary notions. First of them is a so called 'co-occurrence complement' $\bar{x}$ of an entity x:

$$\bar{x} = \{(e,w) | \exists e, w.(e, cooc, x, w) \in KB \vee (x, cooc, e, w) \in KB\} \tag{6}$$

where KB is the knowledge base, i.e., the set of the aggregated co-occurrence statements computed as shown in 'Corpus-wide co-occurrence'. Additionally, we define an element-set projection of an entity's co-occurrence complement $\bar{x}$ as $\bar{x}_1 = \{y | \exists w.w \neq 0 \wedge (y,w) \in \bar{x}\}$,

i.e., set of all the entities in the co-occurrence complement abstracting from the corresponding co-occurrence weights. Finally, we use a shorthand notation $\bar{x}[y] = w$ such that $(y, w) \in \bar{x}$ for a quick reference to the weight corresponding to an entity in a co-occurrence complement. If an entity y is missing in the co-occurrence complement of x, we define $\bar{x}[y] = 0$.

Example 8 *Assuming that the knowledge base consists only from one co-occurrence tuple* $(\texttt{parkinsonism}, \texttt{cooc}, \texttt{DRD}, 0.545)$ *from the previous Example 2, we can define two co-occurrence complements on the entities in it:*

$$\overline{\texttt{parkinsonism}} = \{(\texttt{DRD}, 0.545)\}, \qquad \overline{\texttt{DRD}} = \{(\texttt{parkinsonism}, 0.545)\}.$$

The element-set projection of $\overline{\texttt{parkinsonism}}$ *is then a set* $\{\texttt{DRD}\}$*, while* $\overline{\texttt{parkinsonism}}[\texttt{DRD}]$ *equals* 0.545.

Now we can define the similarity between two entities a, b in a SKIMMR knowledge base as:

$$sim(a, b) = \frac{\sum_{z \in \bar{a}_1 \cap \bar{b}_1} \bar{a}[z]\bar{b}[z]}{\sqrt{\sum_{x \in \bar{a}_1} \bar{a}[x]^2}\sqrt{\sum_{y \in \bar{b}_1} \bar{b}[y]^2}} \tag{7}$$

where $\bar{a}, \bar{b}$ are the co-occurrence complements of a, b, and $\bar{a}_1, \bar{b}_1$ their element-set projections. It can be easily seen that the formula directly corresponds to the definition of cosine distance: its top part is the dot product of the co-occurrence context vectors corresponding to the entities a, b, while the lower part is multiplication of the vectors' sizes (Euclidean norms in particular).

The MeSH-based semantic similarity of entities, introduced in 'Running and measuring the random walks', is defined as

$$sim_M(X, Y) = \max_{u \in C_S(X), v \in C_S(Y)} \frac{2 \cdot dpt(lcs(u, v))}{dpt(u) + dpt(v)} \tag{8}$$

where the specific tree codes in the $C_S(X), C_S(Y)$ are interpreted as nodes in the MeSH taxonomy, the *lcs* stands for the least common subsumer of two nodes in the taxonomy and *dpt* is the depth of a node in the taxonomy (defined as zero if no node is supplied as an argument, i.e., if *lcs* has no result). The formula we use is essentially based on a frequently used taxonomy-based similarity measure defined in *Wu & Palmer (1994)*. We only maximise it across all possible cluster annotations of the two input entities to find the best match. Note that this strategy is safe in case of a resource with as low ambiguity as MeSH – while there are often more annotations of a term, they do not refer to different senses but rather to different branches in the taxonomy. Therefore using the maximum similarity corresponds to finding the most appropriate branch in the MeSH taxonomy along which the terms can be compared.

Entropies

'Running and measuring the random walks' introduced entropies for expressing information value of SKIMMR evaluation samples (*i.e.*, random walks and their contexts). The entropies are defined using the notion of MeSH cluster size ($cs(\ldots)$) introduced in the main part of the article. Given a set Z of nodes of interest, the entropy based on MeSH cluster annotations, $H_M(Z)$, is computed as

$$H_M(Z) = -\sum_{C_i \in C(Z)} \frac{cs(C_i)}{\sum_{C_j \in C(Z)} cs(C_j)} \cdot \log_2 \frac{cs(C_i)}{\sum_{C_j \in C(Z)} cs(C_j)} \tag{9}$$

where C is one of C_A, C_S, depending whether we consider the abstract or the specific nodes. Similarly, the component-based entropy $H_C(Z)$ is defined as

$$H_C(Z) = -\sum_{C_i \in B(Z)} \frac{|C_i|}{\sum_{C_j \in B(Z)} |C_j|} \cdot \log_2 \frac{|C_i|}{\sum_{C_j \in B(Z)} |C_j|} \tag{10}$$

where $B(Z)$ is a function returning a set of biconnected components in the envelope Z, which is effectively a set of subsets of nodes from Z.

Precision and recall

The indices of related articles are compared using precision and recall measures, as stated in 'Comparing the indices of related articles'. Let $I_S : P \to 2^P, I_G : P \to 2^P$ be the SKIMMR and gold standard indices of related publications, respectively (P being a set of publication identifiers). Then the precision and recall for a publication $p \in P$ are computed as

$$pre(p) = \frac{|I_S(p) \cap I_G(p)|}{|I_S(p)|}, \qquad rec(p) = \frac{|I_S(p) \cap I_G(p)|}{|I_G(p)|} \tag{11}$$

respectively. To balance the possibly quite different lengths of the lists of related articles, we limit the computation of the precision and recall up to at most 50 most relevant items in the lists. The average values of precision and recall for a corpus of articles $X \subseteq P$ are computed as

$$pre_{avg}(X) = \frac{\sum_{p \in X} pre(p)}{|X|}, \qquad rec_{avg}(X) = \frac{\sum_{p \in X} rec(p)}{|X|} \tag{12}$$

respectively.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This publication has emanated from research supported in part by research grants from Science Foundation Ireland (SFI) under Grant Numbers SFI/08/CE/I1380, SFI/08/CE/I1380 – STTF 11 (2), and SFI/12/RC/2289. Work was also supported under NIH grants RO1-GM083871 and RO1-MH079068-01A2. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Science Foundation Ireland (SFI): SFI/08/CE/I1380, SFI/08/CE/I1380–STTF 11 (2), SFI/12/RC/2289.
NIH: RO1-GM083871, RO1-MH079068-01A2.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Vít Nováček conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, prepared figures and/or tables, reviewed drafts of the paper, implemented the SKIMMR prototype and corresponding experimental validation scripts.
- Gully A.P.C. Burns conceived and designed the experiments, wrote the paper, reviewed drafts of the paper.

Ethics

The following information was supplied relating to ethical approvals (i.e., approving body and any reference numbers):

IRB name: USC UPIRB, approval number: UP-12-00414.

Data Deposition

The following information was supplied regarding the deposition of related data:

http://skimmr.org/resources/skimmr/pd.tgz (Parskinson's Disease instance of SKIMMR, canned archive version)

http://skimmr.org/resources/skimmr/sma.tgz (Spinal Muscular Atrophy instance of SKIMMR, canned archive version)

http://skimmr.org/resources/skimmr/trec.tgz (TREC instance of SKIMMR, canned archive version)

http://skimmr.org/resources/skimmr/plots.tgz (complete plots of the results).

REFERENCES

Aronson AR, Lang F-M. 2010. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association* **17(3)**:229–236 DOI 10.1136/jamia.2009.002733.

Baroni M, Lenci A. 2010. Distributional memory: a general framework for corpus-based semantics. *Computational Linguistics* **36(4)**:673–721 DOI 10.1162/coli_a_00016.

Biemann C, Coppola B, Glass MR, Gliozzo A, Hatem M, Riedl M. 2013. JoBimText visualizer: a graph-based approach to contextualizing distributional similarity. In: *Proceedings of TextGraphs-8 graph-based methods for natural language processing, Seattle, Washington, USA*. Association for Computational Linguistics, 6–10.

Carrington PJ, Scott J, Wasserman S. 2005. *Models and methods in social network analysis*. Cambridge: Cambridge University Press.

Cohen AM, Hersh WR. 2006. The trec 2004 genomics track categorization task: classifying full text biomedical documents. *Journal of Biomedical Discovery and Collaboration* **1(1)**: Article 4 DOI 10.1186/1747-5333-1-4.

Crestani F. 1997. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review* **11(6)**:453–482 DOI 10.1023/A:1006569829653.

Crockford D. 2006. The application/json media type for JavaScript Object Notation (JSON). *Available at http://www.ietf.org/rfc/rfc4627.txt* (accessed July 2013).

d'Amato C. 2007. Similarity-based learning methods for the semantic web. PhD Thesis.

Dietze H, Alexopoulou D, Alvers MR, Barrio-Alvers B, Doms A, Hakenberg J., Mönnich J, Plake C, Reischuk A, Royer L, Wächter T, Zschunke M, Schroeder M. 2008. GoPubMed: exploring pubmed with ontological background knowledge. In: Ashburner M, Leser U, Rebholz-Schuhmann D, eds. *Ontologies and text mining for life sciences: current status and future perspectives*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Dowdy S, Weardon S, Chilko D. 2005. *Statistics for research*, 3rd edition. John Wiley & Sons, Inc.

Etzioni O, Banko M, Cafarella MJ. 2006. Machine reading. In: *Proceedings of the 2007 AAAI spring symposium on machine reading*. AAAI Press.

Foulds J, Smyth P. 2013. Modeling scientific impact with topical influence regression. In: *Proceedings of the 2013 conference on empirical methods in natural language processing, Seattle, Washington, USA*. Association for Computational Linguistics, 113–123.

Friedman C, Shagina L, Lussier Y, Hripcsak G. 2004. Automated encoding of clinical documents based on natural language processing. *Journal of the American Medical Informatics Association* **11(5)**:392–402 DOI 10.1197/jamia.M1552.

Hájek P. 1998. *Metamathematics of fuzzy logic*. Dordrecht: Kluwer.

Hopcroft J, Tarjan R. 1973. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM* **16(6)**:372–378 DOI 10.1145/362248.362272.

Hristovski D, Friedman C, Rindflesch TC, Peterlin B. 2006. Exploiting semantic relations for literature-based discovery. In: *AMIA annual symposium proceedings*, vol. 2006. American Medical Informatics Association, 349–353.

Huynh DF, Karger DR, Miller RC. 2007. Exhibit: lightweight structured data publishing. In: *Proceedings of the 16th international conference on World Wide Web*. 737–746.

Lin J, Wilbur WJ. 2007. PubMed related articles: a probabilistic topic-based model for content similarity. *BMC Bioinformatics* **8(1)**:423 DOI 10.1186/1471-2105-8-423.

Liu Y, Bill R, Fiszman M, Rindflesch T, Pedersen T, Melton GB, Pakhomov SV. 2012. Using semrep to label semantic relations extracted from clinical text. In: *AMIA annual symposium proceedings*, vol. 2012. American Medical Informatics Association, 587.

Lovász L. 1993. *Random walks on graphs: a survey, Bolyai society mathematical studies,* vol. 2. 1–46.

Manning CD, Raghavan P, Schütze H. 2008. *Introduction to information retrieval.* Cambridge: Cambridge University Press.

Morelli M, Arabia G, Salsone M, Novellino F, Giofrè L, Paletta R, Messina D, Nicoletti G, Condino F, Gallo O, Lanza P, Quattrone A. 2011. Accuracy of magnetic resonance parkinsonism index for differentiation of progressive supranuclear palsy from probable or possible parkinson disease. *Movement Disorders* **26(3)**:527–533 DOI 10.1002/mds.23529.

Mulkar R, Hobbs JR, Hovy E, Chalupsky H, Lin C-Y. 2007. Learning by reading: two experiments. In: *Proceedings of the IJCAI workshop on knowledge and reasoning for answering questions (KRAQ)*. Hyderabad, India.

Müller HM, Kenny EE, Sternberg PW. 2004. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biology* **2(11)**:e309 DOI 10.1371/journal.pbio.0020309.

Nováček V, Groza T, Handschuh S, Decker S. 2010. CORAAL–dive into publications, bathe in the knowledge. *Web Semantics: Science, Services and Agents on the World Wide Web* **8(2–3)**:176–181 DOI 10.1016/j.websem.2010.03.008.

Nováček V, Handschuh S, Decker S. 2011. Getting the meaning right: a complementary distributional layer for the web semantics. In: *Proceedings of ISWC'11*. Springer.

Poon H, Domingos P. 2010. Machine reading: a "killer app" for statistical relational AI. In: *AAAI workshop on statistical relational artificial intelligence*. AAAI Press.

Pratt W. 1997. Dynamic organization of search results using the umls. In: *Proceedings of the AMIA annual fall symposium*. American Medical Informatics Association, 480.

Pratt W, Wasserman H. 2000. Querycat: automatic categorization of medline queries. In: *Proceedings of the AMIA symposium*. American Medical Informatics Association, 655.

Ramakrishnan C, Mendes PN, da Gama RATS, Ferreira GCN, Sheth AP. 2008. Joint extraction of compound entities and relationships from biomedical literature. In: *Web intelligence*. IEEE, 398–401.

Renear AH, Palmer CL. 2009. Strategic reading, ontologies, and the future of scientific publishing. *Science* **325(5942)**:828–832 DOI 10.1126/science.1157784.

Shannon CE. 1948. A mathematical theory of communication. *Bell System Technical Journal* **27**:379–423 DOI 10.1002/j.1538-7305.1948.tb01338.x.

Singhal A. 2001. Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* **24**:35–43.

Strassel S, Adams D, Goldberg H, Herr J, Keesing R, Oblinger D, Simpson H, Schrag R, Wright J. 2010. The DARPA machine reading program-encouraging linguistic and reasoning research with a series of reading tasks. In: *Proceedings of the international conference on language resources and evaluation*. Valletta, Malta.

Talley EM, Newman D, Mimno D, Herr BW, Wallach HM, Burns GAPC, Leenders AGM, McCallum A. 2011. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods* **8(6)**:443–444 DOI 10.1038/nmeth.1619.

Watts DJ, Strogatz SH. 1998. Collective dynamics of 'small-world' networks. *Nature* **393(6684)**:440–442 DOI 10.1038/30918.

Wilkowski B, Fiszman M, Miller CM, Hristovski D, Arabandi S, Rosemblat G, Rindflesch TC. 2011. Graph-based methods for discovery browsing with semantic predications. In: *AMIA annual symposium proceedings*, vol. 2011. American Medical Informatics Association, 1514–1523.

Wu Z, Palmer M. 1994. Verbs semantics and lexical selection. In: *Proceedings of the 32nd annual meeting on association for computational linguistics, ACL'94, Stroudsburg, PA, USA*. Association for Computational Linguistics, 133–138.

Yan Y, Okazaki N, Matsuo Y, Yang Z, Ishizuka M. 2009. Unsupervised relation extraction by mining wikipedia texts using information from the web. In: *Proceedings of ACL/AFNLP'09*. Association for Computational Linguistics, 1021–1029.

Zadrozny S, Nowacka K. 2009. Fuzzy information retrieval model revisited. *Fuzzy Sets and Systems* **160(15)**:2173–2191 DOI 10.1016/j.fss.2009.02.012.

Part III

. . . to Knowledge Graph Embeddings

Chapter 8

Injecting Axioms into Knowledge Graphs

Regularizing Knowledge Graph Embeddings via Equivalence and Inversion Axioms

Pasquale Minervini[1(✉)], Luca Costabello[2], Emir Muñoz[1,2], Vít Nováček[1], and Pierre-Yves Vandenbussche[2]

[1] Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
{pasquale.minervini,emir.munoz,vit.novacek}@insight-centre.org
[2] Fujitsu Ireland Ltd., Galway, Ireland
{luca.costabello,emir.munoz,pierre-yves.vandenbussche}@ie.fujitsu.com

Abstract. Learning embeddings of entities and relations using neural architectures is an effective method of performing statistical learning on large-scale relational data, such as knowledge graphs. In this paper, we consider the problem of regularizing the training of neural knowledge graph embeddings by leveraging external background knowledge. We propose a principled and scalable method for leveraging equivalence and inversion axioms during the learning process, by imposing a set of model-dependent soft constraints on the predicate embeddings. The method has several advantages: *(i)* the number of introduced constraints does not depend on the number of entities in the knowledge base; *(ii)* regularities in the embedding space effectively reflect available background knowledge; *(iii)* it yields more accurate results in link prediction tasks over non-regularized methods; and *(iv)* it can be adapted to a variety of models, without affecting their scalability properties. We demonstrate the effectiveness of the proposed method on several large knowledge graphs. Our evaluation shows that it consistently improves the predictive accuracy of several neural knowledge graph embedding models (for instance, the MRR of TransE on WordNet increases by 11%) without compromising their scalability properties.

1 Introduction

Knowledge graphs are graph-structured knowledge bases, where factual knowledge is represented in the form of relationships between entities: they are powerful instruments in search, analytics, recommendations, and data integration. This justified a broad line of research both from academia and industry, resulting in projects such as DBpedia (Auer et al. 2007), Freebase (Bollacker et al. 2007), YAGO (Suchanek et al. 2012), NELL (Carlson et al. 2010), and Google's Knowledge Graph and Knowledge Vault projects (Dong et al. 2014).

However, despite their size, knowledge graphs are often very far from being complete. For instance, 71% of the people described in Freebase have no known place of birth, 75% have no known nationality, and the coverage for less used relations can be even lower (Dong et al. 2014). Similarly, in DBpedia, 66% of

M. Ceci et al. (Eds.): ECML PKDD 2017, Part I, LNAI 10534, pp. 668–683, 2017.
https://doi.org/10.1007/978-3-319-71249-9_40

the persons are also missing a place of birth, while 58% of the scientists are missing a fact stating what they are known for (Krompaß et al. 2015).

In this work, we focus on the problem of *predicting missing links* in large knowledge graphs, so to discover new facts about the world. In the literature, this problem is referred to as *link prediction* or *knowledge base population*: we refer to Nickel et al. (2016) for a recent survey on machine learning-driven solutions to this problem.

Recently, *neural knowledge graph embedding models* (Nickel et al. 2016) – neural architectures for embedding entities and relations in continuous vector spaces – have received a growing interest: they achieve state-of-the-art link prediction results, while being able to scale to very large and highly-relational knowledge graphs. Furthermore, they can be used in a wide range of applications, including entity disambiguation and resolution (Bordes et al. 2014), taxonomy extraction (Nickel et al. 2016), and query answering on probabilistic databases (Krompaß et al. 2014). However, a limitation in such models is that they only rely on existing facts, without making use of any form of background knowledge. At the time of this writing, how to efficiently leverage preexisting knowledge for learning more accurate neural knowledge graph embeddings is still an open problem (Wang et al. 2015).

Contribution – In this work, we propose a principled and scalable method for leveraging external background knowledge for regularising neural knowledge graph embeddings. In particular, we leverage background axioms in the form $p \equiv q$ and $p \equiv q^-$, where the former denotes that relations p and q are *equivalent*, such as in the case of relations PARTOF and COMPONENTOF, while the latter denotes that the relation p is the *inverse* of the relation q, such as in the case of relations PARTOF and HASPART. Such axioms are used for defining and imposing a set of model-dependent soft constraints on the relation embeddings during the learning process. Such constraints can be considered as regularizers, reflecting available prior knowledge on the distribution of embedding representations of relations.

The proposed method has several advantages: *(i)* the number of introduced constraints is independent on the number of entities, allowing it to scale to large and Web-scale knowledge graphs with millions of entities; *(ii)* relationships between relation types in the embedding space effectively reflect available background schema knowledge; *(iii)* it yields more accurate results in link prediction tasks than state-of-the-art methods; and *(iv)* it is a general framework, applicable to a variety of embedding models. We demonstrate the effectiveness of the proposed method in several link prediction tasks: we show that it consistently improves the predictive accuracy of the models it is applied to, without negative impact on their scalability properties.

2 Preliminaries

Knowledge Graphs – A knowledge graph is a graph-structured knowledge base, where factual information is stored in the form of relationships

between entities. Formally, a knowledge graph $\mathcal{G} \triangleq \{\langle s, p, o\rangle\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of $\langle s, p, o\rangle$ triples, each consisting of a *subject* s, a *predicate* p and an *object* o, and encoding the statement "s has a relationship p with o". The subject and object $s, o \in \mathcal{E}$ are entities, $p \in \mathcal{R}$ is a relation type, and $\mathcal{E}, \mathcal{R}$ respectively denote the sets of all entities and relation types in the knowledge graph.

Example 1. Consider the following statement: *"Ireland is located in Northern Europe, and shares a border with the United Kingdom."* It can be expressed by the following triples:

Subject	Predicate	Object
Ireland	locatedIn	Northern Europe
Ireland	neighborOf	United Kingdom

A knowledge graph can be represented as a labelled directed multigraph, in which each triple is represented as an edge connecting two nodes: the source and target nodes represent the subject and object of the triple, and the edge label represents the predicate.

Knowledge graph adhere to the *Open World Assumption* (Hayes and Patel-Schneider 2014): a missing triple does not necessarily imply that the corresponding statement holds false, but rather that its truth value is *unknown*, *i.e.* it cannot be observed in the graph. For instance, the fact that the triple $\langle$United Kingdom, neighborOf, Ireland$\rangle$ is missing from the graph in Example 1 does not imply that the United Kingdom does not share a border with Ireland, but rather that we do not know whether this statement is true or not.

Equivalence and Inversion Axioms – Knowledge graphs are usually endowed with additional background knowledge, describing classes of entities and their properties and characteristics, such as equivalence and symmetry. In this work, we focus on two types of logical axioms in the form $p \equiv q$ and $p \equiv q^{-}$, where $p, q \in \mathcal{R}$ are predicates.

A largely popular knowledge representation formalism for expressing schema axioms is the OWL 2 Web Ontology language (Schneider 2012). According to the OWL 2 RDF-based semantics, the axiom $p \equiv q$ implies that predicates p and q share the same property extension, *i.e.* if $\langle s, p, o\rangle$ is true then $\langle s, q, o\rangle$ is also true (and vice-versa). Similarly, the axiom $p \equiv q^{-}$ implies that the predicate q is the inverse of the predicate p, *i.e.* if $\langle s, p, o\rangle$ is true then $\langle o, q, s\rangle$ is also true (and vice-versa). It is possible to express that a predicate $p \in \mathcal{R}$ is *symmetric* by using the axiom $p \equiv p^{-}$. Such axioms can be expressed by the OWL 2 `owl:equivalentProperty` and `owl:inverseOf` constructs.

Example 2. Consider the following statement: *"The relation* locatedIn *is the inverse of the relation* locationOf*, and the relation* neighborOf *is symmetric."* It can be encoded by the axioms locatedIn $\equiv$ locationOf^{-} and neighborOf $\equiv$ neighborOf^{-}.

Link Prediction – As mentioned earlier, real world knowledge graphs are often largely incomplete. *Link prediction* in knowledge graphs consists in identifying missing triples (facts) in order to discover new facts about a domain of interest. This task is also referred to as *knowledge base population* in literature. We refer to Nickel et al. (2016) for a recent survey on link prediction methods.

The link prediction task can be cast as a *learning to rank* problem, where we associate a *prediction score* ϕ_{spo} to each triple $\langle s, p, o \rangle$ as follows:

$$\phi_{spo} \triangleq \phi(\langle s, p, o \rangle; \Theta),$$

where the score ϕ_{spo} represents the confidence of the model that the statement encoded by the triple $\langle s, p, o \rangle$ holds true, $\phi(\cdot; \Theta)$ denotes a *triple scoring function*, with $\phi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \rightarrow \mathbb{R}$, and Θ represents the parameters of the scoring function and thus of the link prediction model. Triples associated with a higher score by the link prediction model have a higher probability of encoding a true statement, and are thus considered for a completion of the knowledge graph $\mathcal{G}$.

3 Neural Knowledge Graph Embedding Models

Recently, *neural* link prediction models received a growing interest (Nickel et al. 2016). They can be interpreted as simple multi-layer neural networks, where given a triple $\langle s, p, o \rangle$, its score $\phi(\langle s, p, o \rangle; \Theta)$ is given by a two-layer neural network architecture, composed by an *encoding layer* and a *scoring layer*.

Encoding Layer – in the encoding layer, the subject and object entities s and o are mapped to distributed vector representations $\mathbf{e}_s$ and $\mathbf{e}_o$, referred to as *embeddings*, by an encoder $\psi : \mathcal{E} \mapsto \mathbb{R}^k$ such that $\mathbf{e}_s \triangleq \psi(s)$ and $\mathbf{e}_o \triangleq \psi(o)$. Given an entity $s \in \mathcal{E}$, the encoder ψ is usually implemented as a simple embedding layer $\psi(s) \triangleq [\boldsymbol{\Psi}]_s \in \mathbb{R}^k$, where $\boldsymbol{\Psi} \in \mathbb{R}^{|\mathcal{E}| \times k}$ is an embedding matrix (Nickel et al. 2016).
The distributed representations in this layer can be either pre-trained (Baroni et al. 2012) or, more commonly, learnt from data by back-propagating the link prediction error to the embeddings (Bordes et al. 2013; Yang et al. 2015; Trouillon et al. 2016; Nickel et al. 2016).
Scoring Layer – in the scoring layer, the subject and object representations $\mathbf{e}_s$ and $\mathbf{e}_o$ are scored by a predicate-dependent function $\phi_p^\theta(\mathbf{e}_s, \mathbf{e}_o) \in \mathbb{R}$, parametrised by θ.

The architecture of neural link prediction models can be summarized as follows:

$$\begin{aligned} \phi(\langle s, p, o \rangle; \Theta) &\triangleq \phi_p^\theta(\mathbf{e}_s, \mathbf{e}_o) \\ \mathbf{e}_s, \mathbf{e}_o &\triangleq \psi(s), \psi(o), \end{aligned} \tag{1}$$

and the set of parameters Θ corresponds to $\Theta \triangleq \{\theta, \boldsymbol{\Psi}\}$. Neural link prediction model generate distributed embedding representations for all entities in a knowledge graph, as well as a model of determining whether a triple is more likely than

others, by means of a neural network architecture. For such a reason, they are also referred to as *neural knowledge graph embedding models* (Yang et al. 2015; Nickel et al. 2016).

Several neural link prediction models have been proposed in the literature. For brevity, we overview a small subset of these, namely the Translating Embeddings model TransE (Bordes et al. 2013); the Bilinear-Diagonal model DistMult (Yang et al. 2015); and its extension in the complex domain ComplEx (Trouillon et al. 2016). Unlike previous models, such models can scale to very large knowledge graphs, thanks to: (i) a space complexity that grows *linearly* with the number of entities $|\mathcal{E}|$ and relations $|\mathcal{R}|$; and (ii) efficient and scalable scoring functions and parameters learning procedures. In the following, we provide a brief and self-contained overview of such neural knowledge graph embedding models.

TransE – The scoring layer in TransE is defined as follows:

$$\phi_p(\mathbf{e}_s, \mathbf{e}_o) \triangleq -\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| \in \mathbb{R},$$

where $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$ represent the subject and object embeddings, $\mathbf{r}_p \in \mathbb{R}^k$ is a predicate-dependent translation vector, $\|\cdot\|$ denotes either the L_1 or the L_2 norm, and $\|\mathbf{x} - \mathbf{y}\|$ denotes the distance between vectors $\mathbf{x}$ and $\mathbf{y}$. In TransE, the score $\phi_p(\mathbf{e}_s, \mathbf{e}_o)$ is then given by the *similarity* between the translated subject embedding $\mathbf{e}_s + \mathbf{r}_p$ and the object embedding $\mathbf{e}_o$.

DistMult – The scoring layer in DistMult is defined as follows:

$$\phi_p(\mathbf{e}_s, \mathbf{e}_o) \triangleq \langle \mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o \rangle \in \mathbb{R},$$

where, given $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^k$, $\langle \mathbf{x}, \mathbf{y}, \mathbf{z} \rangle \triangleq \sum_{i=1}^{k} \mathbf{x}_i \mathbf{y}_i \mathbf{z}_i$ denotes the standard component-wise multi-linear dot product, and $\mathbf{r}_p \in \mathbb{R}^k$ is a predicate-dependent vector.

ComplEx – The recently proposed ComplEx is related to DistMult, but uses complex-valued embeddings while retaining the mathematical definition of the dot product. The scoring layer in ComplEx is defined as follows:

$$\begin{aligned}\phi_p(\mathbf{e}_s, \mathbf{e}_o) &\triangleq \mathrm{Re}\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right) \\ &= \langle \mathrm{Re}\left(\mathbf{r}_p\right), \mathrm{Re}\left(\mathbf{e}_s\right), \mathrm{Re}\left(\mathbf{e}_o\right) \rangle + \langle \mathrm{Re}\left(\mathbf{r}_p\right), \mathrm{Im}\left(\mathbf{e}_s\right), \mathrm{Im}\left(\mathbf{e}_o\right) \rangle \\ &\quad + \langle \mathrm{Im}\left(\mathbf{r}_p\right), \mathrm{Re}\left(\mathbf{e}_s\right), \mathrm{Im}\left(\mathbf{e}_o\right) \rangle - \langle \mathrm{Im}\left(\mathbf{r}_p\right), \mathrm{Im}\left(\mathbf{e}_s\right), \mathrm{Re}\left(\mathbf{e}_o\right) \rangle \in \mathbb{R},\end{aligned}$$

where given $\mathbf{x} \in \mathbb{C}^k$, $\overline{\mathbf{x}}$ denotes the complex conjugate of $\mathbf{x}$[1], while $\mathrm{Re}\left(\mathbf{x}\right) \in \mathbb{R}^k$ and $\mathrm{Im}\left(\mathbf{x}\right) \in \mathbb{R}^k$ denote the real part and the imaginary part of $\mathbf{x}$, respectively.

4 Training Neural Knowledge Graph Embedding Models

In neural knowledge graph embedding models, the parameters Θ of the embedding and scoring layers are learnt from data. A widely popular strategy for

[1] Given $x \in \mathbb{C}$, its complex conjugate is $\overline{x} \triangleq \mathrm{Re}\left(x\right) - i\mathrm{Im}\left(x\right)$.

Algorithm 1. Learning the model parameters Θ via Projected SGD

Require: Batch size n, epochs τ, learning rates $\boldsymbol{\eta} \in \mathbb{R}^{\tau}$
Ensure: Optimal model parameters $\hat{\Theta}$
for $i = 1, \ldots, \tau$ **do**
  $\mathbf{e}_e \leftarrow \mathbf{e}_e / \|\mathbf{e}_e\|, \ \forall e \in \mathcal{E}$
  {Sample a batch of positive and negative examples $\mathcal{B} = \{(t, \tilde{t})\}$}
  $\mathcal{B} \leftarrow \textsc{SampleBatch}(\mathcal{G}, n)$
  {Compute the gradient of the loss function $\mathcal{J}$ on examples $\mathcal{B}$}
  $g_i \leftarrow \nabla \sum_{(t,\tilde{t}) \in \mathcal{B}} \left[\gamma - \phi(t; \Theta_{i-1}) + \phi(\tilde{t}; \Theta_{i-1})\right]_+$
  {Update the model parameters via gradient descent}
  $\Theta_i \leftarrow \Theta_{i-1} - \eta_i g_i$
end for
return Θ_τ

learning the model parameters is described in Bordes et al. (2013); Yang et al. (2015); Nickel et al. (2016). In such works, authors estimate the optimal parameters by minimizing the following pairwise margin-based ranking loss function $\mathcal{J}$ defined on parameters Θ:

$$\mathcal{J}(\Theta) \triangleq \sum_{t^+ \in \mathcal{G}} \sum_{t^- \in \mathcal{C}(t^+)} \left[\gamma - \phi(t^+; \Theta) + \phi(t^-; \Theta)\right]_+ \tag{2}$$

where $[x]_+ = \max\{0, x\}$, and $\gamma \geq 0$ specifies the width of the margin. Positive examples t^+ are composed by all triples in $\mathcal{G}$, and negative examples t^- are generated by using the following *corruption process*:

$$\mathcal{C}(\langle s, p, o \rangle) \triangleq \{\langle \tilde{s}, p, o \rangle \mid \tilde{s} \in \mathcal{E}\} \cup \{\langle s, p, \tilde{o} \rangle \mid \tilde{o} \in \mathcal{E}\},$$

which, given a triple, generates a set of corrupt triples by replacing its subject and object with all other entities in $\mathcal{G}$. This method of sampling negative examples is motivated by the *Local Closed World Assumption* (LCWA) (Dong et al. 2014). According to the LCWA, if a triple $\langle s, p, o \rangle$ exists in the graph, other triples obtained by corrupting either the subject or the object of the triples not appearing in the graph can be considered as negative examples. The optimal parameters can be learnt by solving the following minimization problem:

$$\begin{array}{ll} \underset{\Theta}{\text{minimize}} & \mathcal{J}(\Theta) \\ \text{subject to} & \forall e \in \mathcal{E} : \ \|\mathbf{e}_e\| = 1, \end{array} \tag{3}$$

where Θ denotes the parameters of the model. The norm constraints on the entity embeddings prevent to trivially solve the optimization problem by increasing the norm of the embedding vectors (Bordes et al. 2014). The loss function in Eq. (2) will reach its global minimum 0 iff, for each pair of positive and negative examples t^+ and t^-, the score of the (true) triple t^+ is higher with a margin of at least γ than the score of the (missing) triple t^-. Following Yang et al. (2015), we use the Projected Stochastic Gradient Descent (SGD) algorithm (outlined in

Algorithm 1) for solving the loss minimization problem in Eq. (3), and AdaGrad (Duchi et al. 2011) for automatically selecting the optimal learning rate η at each iteration.

5 Regularizing via Background Knowledge

We now propose a method for incorporating background schema knowledge, provided in the form of equivalence and inversion axioms between predicates, in neural knowledge graph embedding models. Formally, let $\mathcal{A}_1$ and $\mathcal{A}_2$ denote the following two sets of equivalence and inversion axioms between predicates:

$$\mathcal{A}_1 \triangleq \{p_1 \equiv q_1, \ldots, p_m \equiv q_m\} \qquad \mathcal{A}_2 \triangleq \{p_{m+1} \equiv q_{m+1}^-, \ldots, p_n \equiv q_n^-\} \tag{4}$$

where $1 \leq m \leq n$, and $\forall i \in \{1, \ldots, n\} : p_i, q_i \in \mathcal{R}$. Recall that each axiom $p \equiv q$ encodes prior knowledge that predicates p and q are equivalent, *i.e.* they share the same extension. Similarly, each axiom $p \equiv q^-$ encodes prior knowledge that the predicate p and the *inverse* of the predicate q are equivalent.

Equivalence Axioms – Consider the case in which predicates $p \in \mathcal{R}$ and $q \in \mathcal{R}$ are equivalent, as encoded by the axiom $p \equiv q$. This implies that a model with scoring function $\phi(\cdot\,; \Theta)$ and parameters Θ should assign the same scores to the triples $\langle s, p, o \rangle$ and $\langle s, q, o \rangle$, for all entities $s, o \in \mathcal{E}$:

$$\phi(\langle s, p, o \rangle; \Theta) = \phi(\langle s, q, o \rangle; \Theta) \quad \forall s, o \in \mathcal{E}. \tag{5}$$

A simple method for enforcing the constraint in Eq. (5) during the parameter learning process consists in solving the loss minimization problem in Eq. (3) under the additional equality constraints in Eq. (5). However, this solution results in introducing $\mathcal{O}(|\mathcal{E}|^2)$ constraints in the optimization problem in Eq. (3), a quantity that grows *quadratically* with the number of entities $|\mathcal{E}|$. This solution may not be feasible for very large knowledge graphs, which typically contain millions of entities or more, while $|\mathcal{R}|$ is usually several orders of magnitude lower. A more efficient method consists in enforcing the model to associate *similar embedding representations* to both p and q, *i.e.* $\mathbf{r}_p = \mathbf{r}_q$. This solution can be encoded by *a single constraint*, satisfying all identities in Eq. (5).

Inversion Axioms – Consider the case in which the predicate p (*e.g.* PARTOF) and the inverse of the predicate q (*e.g.* HASPART) are equivalent, as encoded by the axiom $p \equiv q^-$. This implies that a model with scoring function $\phi(\cdot\,; \Theta)$ and parameters Θ should assign the same scores to the triples $\langle s, p, o \rangle$ and $\langle o, q, s \rangle$, for all entities $s, o \in \mathcal{E}$:

$$\phi(\langle s, p, o \rangle; \Theta) = \phi(\langle o, q, s \rangle; \Theta) \quad \forall s, o \in \mathcal{E}. \tag{6}$$

Also in this case we can enforce the identity in Eq. (6) through a single constraint on the embeddings of predicates p and q. In the following, we derive the constraints for the models TRANSE, DISTMULT and COMPLEX. The constraints

rely on a function $\Phi(\cdot)$ that applies a model-dependent transformation to the predicate embedding $\mathbf{r}_q$.

TransE: We want to enforce that, for any pair of s and o embedding vectors $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$, the score associated to the triples $\langle s, p, o\rangle$ and $\langle o, q, s\rangle$ are the same. Formally:

$$\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| = \|\mathbf{e}_o + \mathbf{r}_q - \mathbf{e}_s\|, \quad \forall \mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k \tag{7}$$

where $\|\cdot\|$ denotes either the L_1 or the L_2 norm.

Theorem 1. *The identity in Eq. (7) is satisfied by imposing:*

$$\mathbf{r}_p = \Phi(\mathbf{r}_q) \quad \textit{such that} \quad \Phi(\mathbf{r}_q) \triangleq -\mathbf{r}_q.$$

Proof. *For any $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k$, the following result holds:*

$$\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| = \|\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s\|,$$

where $\|\cdot\|$ is a norm on $\mathbb{R}^k$. Because of the absolute homogeneity property *of norms we have that, for any $\alpha \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^k$:*

$$\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|.$$

It follows that:

$$\begin{aligned}\|\mathbf{e}_s + \mathbf{r}_p - \mathbf{e}_o\| &= \|-1\,(\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s)\,\| \\ &= |-1|\|\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s\| \qquad \textit{(absolute homogeneity property)} \\ &= \|\mathbf{e}_o - \mathbf{r}_p - \mathbf{e}_s\|.\end{aligned}$$

DistMult: We want to enforce that:

$$\langle \mathbf{r}_p, \mathbf{e}_s, \mathbf{e}_o\rangle = \langle \mathbf{r}_q, \mathbf{e}_o, \mathbf{e}_s\rangle, \quad \forall \mathbf{e}_s, \mathbf{e}_o \in \mathbb{R}^k \tag{8}$$

A limitation in DistMult, addressed by ComplEx, is that its scoring function is *symmetric*, *i.e.* it assigns the same score to $\langle s, p, o\rangle$ and $\langle o, p, s\rangle$, due to the commutativity of the element-wise product.

The identity in Eq. (8) is thus satisfied by imposing $\mathbf{r}_p = \Phi(\mathbf{r}_q)$ such that $\Phi(\mathbf{r}_q) \triangleq \mathbf{r}_q$.

ComplEx: We want to enforce that:

$$\mathrm{Re}\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o}\rangle\right) = \mathrm{Re}\left(\langle \mathbf{r}_q, \mathbf{e}_o, \overline{\mathbf{e}_s}\rangle\right), \quad \forall \mathbf{e}_s, \mathbf{e}_o \in \mathbb{C}^k. \tag{9}$$

The identity in Eq. (9) can be satisfied as follows:

Theorem 2. *The identity in Eq. (9) is satisfied by imposing:*

$$\mathbf{r}_p = \Phi(\mathbf{r}_q) \quad \textit{such that} \quad \Phi(\mathbf{r}_q) \triangleq \overline{\mathbf{r}_q}.$$

Proof. *For any* $\mathbf{e}_s, \mathbf{e}_o \in \mathbb{C}^k$*, the following result holds:*

$$Re\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right)) = Re\left(\langle \overline{\mathbf{r}_p}, \mathbf{e}_o, \overline{\mathbf{e}_s} \rangle\right)) .$$

Consider the following steps:

$$\begin{aligned} Re\left(\langle \mathbf{r}_p, \mathbf{e}_s, \overline{\mathbf{e}_o} \rangle\right) &= Re\left(\overline{\langle \overline{\mathbf{r}_p}, \overline{\mathbf{e}_s}, \mathbf{e}_o \rangle}\right) && \textit{(since} \overline{(\overline{\mathbf{x}})} = \mathbf{x}\textit{)} \\ &= Re\left(\overline{\langle \overline{\mathbf{r}_p}, \mathbf{e}_o, \overline{\mathbf{e}_s} \rangle}\right) && \textit{(commutative property)} \\ &= Re\left(\langle \overline{\mathbf{r}_p}, \mathbf{e}_o, \overline{\mathbf{e}_s} \rangle\right) && \textit{(since} Re\left(\overline{\mathbf{x}}\right) = Re\left(\mathbf{x}\right)\textit{).} \end{aligned}$$

Similar procedures for deriving the function $\Phi(\cdot)$ can be used in the context of other knowledge graph embedding models.

5.1 Regularizing via Soft Constraints

One solution for integrating background schema knowledge consists in solving the loss minimization problem in Eq. (3) under additional hard equality constraints on the predicate embeddings, for instance by enforcing $\mathbf{r}_p = \mathbf{r}_q$ for all $p \equiv q \in \mathcal{A}_1$, and $\mathbf{r}_p = \Phi(\mathbf{r}_q)$ for all $p \equiv q^- \in \mathcal{A}_2$. However, this solution does not cover cases in which two predicates are not strictly equivalent but still share very similar semantics, such as in the case of predicates MARRIEDWITH and PARTNEROF.

A more flexible solution consists in relying on *soft constraints* (Meseguer et al. 2006), which are used to formalize *desired properties* of the model rather than requirements that cannot be violated: we propose relying on weighted soft constraints for encoding our background knowledge on latent predicate representations.

Formally, we extend the loss function $\mathcal{J}$ described in Eq. (2) with an additional penalty term $\mathcal{R}_\mathcal{S}$ for enforcing a set of desired relationships between the predicate embeddings. This process leads to the following novel loss function $\mathcal{J}_\mathcal{S}$:

$$\begin{aligned} \mathcal{R}_\mathcal{S}(\Theta) &\triangleq \sum_{p \equiv q \in \mathcal{A}_1} D\left[\mathbf{r}_p \| \mathbf{r}_q\right] + \sum_{p \equiv q^- \in \mathcal{A}_2} D\left[\mathbf{r}_p \| \Phi(\mathbf{r}_q)\right] \\ \mathcal{J}_\mathcal{S}(\Theta) &\triangleq \mathcal{J}(\Theta) + \lambda \mathcal{R}_\mathcal{S}(\Theta), \end{aligned} \tag{10}$$

where $\lambda \geq 0$ is the weight associated with the soft constraints, and $D\left[\mathbf{x} \| \mathbf{y}\right]$ is a divergence measure between two vectors $\mathbf{x}$ and $\mathbf{y}$. In our experiments, we use the Euclidean distance as divergence measure, *i.e.* $D\left[\mathbf{x} \| \mathbf{y}\right] \triangleq \|\mathbf{x} - \mathbf{y}\|_2^2$.

In particular, $\mathcal{R}_\mathcal{S}$ in Eq. (10) can be thought of as a schema-aware *regularization term*, which encodes our prior knowledge on the distribution of predicate embeddings. Note that the formulation in Eq. (10) allows us to freely interpolate between *hard constraints* ($\lambda = \infty$) and the original models represented by the loss function $\mathcal{J}$ ($\lambda = 0$), permitting to adaptively specify the relevance of each logical axiom in the embedding model.

6 Related Works

How to effectively improve neural knowledge graph embeddings by making use of background knowledge is a largely unexplored field. Chang et al. (2014); Krompass et al. (2014); Krompaß et al. (2015) make use of type information about entities for only considering interactions between entities belonging to the domain and range of each predicate, assuming that type information about entities is complete. In Minervini et al. (2016), authors assume that type information can be incomplete, and propose to adaptively decrease the score of each missing triple depending on the available type information. These works focus on type information about entities, while we propose a method for leveraging background knowledge about relation types which can be used jointly with the aforementioned methods.

Dong et al. (2014); Nickel et al. (2014); Wang et al. (2015) propose combining observable patterns in the form of rules and latent features for link prediction tasks. However, rules are not used *during* the parameters learning process, but rather *after*, in an ensemble fashion. Wang et al. (2015) suggest investigating how to incorporate logical schema knowledge during the parameters learning process as a future research direction. Rocktäschel et al. (2015) regularize relation and entity representations by grounding first-order logic rules. However, as they state in their paper, adding a very large number of ground constraints does not scale to domains with a large number of entities and predicates.

In this work we focus on *2-way* models rather than *3-way* models (García-Durán et al. 2014), since the former received an increasing attention during the last years, mainly thanks to their scalability properties (Nickel et al. 2016). According to García-Durán et al. (2014), 3-way models such as RESCAL (Nickel et al. 2011; 2012) are more prone to overfitting, since they typically have a larger number of parameters. It is possible to extend the proposed model to RESCAL, whose score for a $\langle s, p, o \rangle$ triple is $\mathbf{e}_s^T \mathbf{W}_p \mathbf{e}_o$. For instance, it is easy to show that $\mathbf{e}_s^T \mathbf{W}_p \mathbf{e}_o = \mathbf{e}_o^T \mathbf{W}_p^T \mathbf{e}_s$. However, extending the proposed method to more complex 3-way models, such as the latent factor model proposed by Jenatton et al. (2012) or the ER-MLP model (Dong et al. 2014) can be less trivial.

7 Evaluation

We evaluate the proposed schema-based soft constraints on three datasets: WordNet, DBpedia and YAGO3. Each dataset is composed by a *training*, a *validation* and a *test* set of triples, as summarized in Table 1. All material needed for reproducing the experiments in this paper is available online[2].

WordNet (Miller 1995) is a lexical knowledge base for the English language, where entities correspond to word senses, and relationships define lexical relations between them: we use the version made available by Bordes et al. (2013).

[2] At https://github.com/pminervini/neural-schema-regularization.

Table 1. Statistics for the datasets used in experiments

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#training	#validation	#test
WordNet	40,943	18	141,442	5,000	5,000
DBpedia	32,510	7	289,825	5,000	5,000
YAGO3	123,182	37	1,079,040	5,000	5,000

YAGO3 (Mahdisoltani et al. 2015) is a large knowledge graph automatically extracted from several sources: our dataset is composed by facts stored in the YAGO3 Core Facts component of YAGO3.

DBpedia (Auer et al. 2007) is a knowledge base created extracting structured, multilingual knowledge from Wikipedia, and made available using Semantic Web and Linked Data standards. We consider a fragment extracted following the indications from Krompaß et al. (2014), by considering relations in the music domain[3].

The axioms we used in experiments are simple common-sense rules, and are listed in Table 1.

Evaluation Metrics – For evaluation, for each test triple $\langle s, p, o \rangle$, we measure the quality of the ranking of each test triple among all possible subject and object substitutions $\langle \tilde{s}, p, o \rangle$ and $\langle s, p, \tilde{o} \rangle$, with $\tilde{s}, \tilde{o} \in \mathcal{E}$. Mean Reciprocal Rank (MRR) and Hits@k as described by Bordes et al. (2013); Nickel et al. (2016); Trouillon et al. (2016) are widely adopted evaluation measures for evaluating knowledge graph completion algorithms. The measures are reported in the *raw* and *filtered* settings (Bordes et al. 2013). In the *filtered* setting, metrics are computed after removing all the other positive (true) triples that appear in either training, validation or test set from the ranking, whereas in the *raw* setting these are not removed. The filtered setting is motivated by observing that ranking a positive test triple after another true triple should not be considered a mistake (Bordes et al. 2013).

Evaluation Setting – In our experiments we consider three knowledge graph embedding models – TransE, ComplEx and DistMult, as described in Sect. 3. For evaluating the effectiveness of the proposed method, we train them using both the standard loss function $\mathcal{J}$, defined in Eq. (2), and the proposed schema-aware loss function $\mathcal{J}_{\mathcal{S}}$, defined in Eq. (10). Models trained by using the proposed method are denoted by the R superscript.

For each model and dataset, hyper-parameters were selected on the validation set by grid search. Specifically, we selected the embedding size $k \in \{20, 50, 100, 150\}$, the regularization weight $\lambda \in \{0, 10^{-4}, 10^{-2}, \ldots, 10^{6}\}$ and, in TransE, the norm $\|\cdot\|$ is selected across the L_1 and the L_2 norm.

[3] Following Krompass et al. (2014), such relations are album, associated band, associated musical artist, genre, musical artist, musical band, and record-Label.

Table 2. Link prediction results (Hits@k and Mean Reciprocal Rank, filtered setting) on WordNet, DBpedia and YAGO3.

	WordNet				**DBpedia**				**YAGO3**			
	Hits@N (%)			MRR	Hits@N (%)			MRR	Hits@N (%)			MRR
	3	5	10		3	5	10		3	5	10	
TransE	79.9	87.3	91.1	0.452	44.3	52.6	59.0	0.245	32.4	40.7	50.5	0.214
TransE^R	**86.9**	**91.6**	**93.3**	**0.566**	**47.8**	**54.0**	**60.0**	**0.256**	**33.4**	**42.5**	**52.0**	**0.248**
DistMult	91.7	93.2	94.2	0.840	44.6	**50.6**	55.7	0.371	**29.9**	**37.2**	46.3	**0.260**
DistMult^R	**92.4**	**93.8**	**94.9**	**0.851**	**44.9**	**50.6**	**55.8**	**0.381**	**29.9**	**37.2**	**46.4**	**0.260**
ComplEx	94.2	94.4	94.6	0.939	52.7	54.2	55.8	0.486	**34.8**	41.5	49.9	**0.304**
ComplEx^R	**94.3**	**94.5**	**94.7**	**0.940**	**53.1**	**54.3**	**55.9**	**0.503**	34.7	**41.6**	**50.0**	**0.304**

Similarly to Yang et al. (2015) we set the margin $\gamma = 1$ and, for each combination of hyper-parameters, we train each model for 1000 epochs. The learning rate in Stochastic Gradient Descent was initially set to 0.1, and then adapted during training by AdaGrad.

Results – We report test results in terms of raw and filtered Mean Reciprocal Rank (MRR), and filtered Hits@k in Table 2. For both the MRR and Hits@k metrics, the higher the results on the test set, the better.

We can see that, in every case, the proposed method – which relies on regularizing relation embeddings by leveraging background knowledge – improves the generalization abilities for each of the models. Results are especially evident for TransE, which largely benefits from the novel regularizer. For instance we can see that, in the WordNet case, the Hits@10 improves from 91.1 to 93.3, while the Mean Reciprocal Rank improves from 0.452 to 0.566. For the remaining models we can only notice marginal improvements, probably because they already able to capture the patterns encoded by the background knowledge.

In Fig. 2 we can see a set of trained WordNet predicate embeddings (using the model TransE), where relationships predicates are described in the axioms in Fig. 1. We can immediately see that, if $p \equiv q^-$, *i.e.* p is the inverse of q, then $\mathbf{r}_p \approx -\mathbf{r}_q$, which means that their embeddings $\mathbf{r}_p$ and $\mathbf{r}_q$ will be similar but will have opposite sign. On the left we set $\lambda = 0$, *i.e.* we do not enforce any soft constraint: we can see that the model is naturally inclined to assign opposite sign embeddings to relations such as PART OF and HAS PART, and HYPONYM and HYPERNYM; however, there is still some error margin in such an assignment, possibly due to the incompleteness of the knowledge graph. On the right we set $\lambda = 10^6$, *i.e.* we enforce the relationships between predicate embeddings via soft constraints: we can see that the aforementioned error margin in modeling the relationships between predicate embeddings is greatly reduced, improving the generalization properties of the model and establishing new state-of-the-art link prediction results on several datasets.

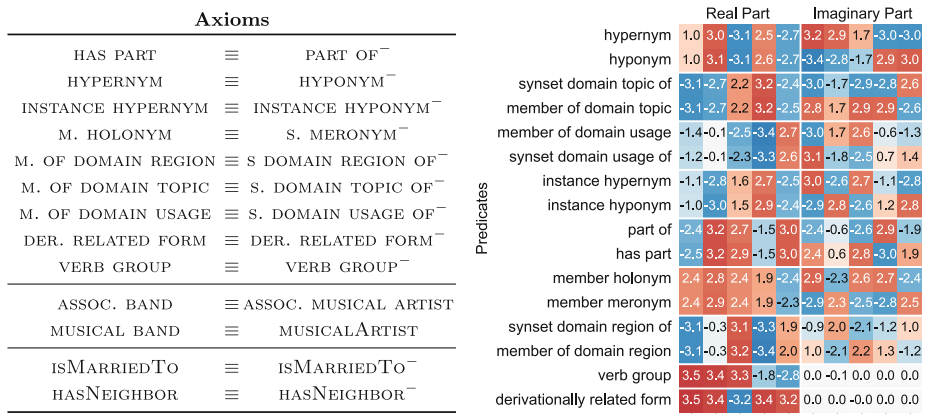


Fig. 1. Axioms used with WordNet, DBpedia and YAGO3 (left) and WordNet predicate embeddings learned by ComplEx (right). Note that if $p \equiv q^-$ (*e.g.* PART OF and HAS PART) then $\mathbf{r}_p \approx \overline{\mathbf{r}_q}$, *i.e.* $\mathbf{r}_p$ and $\mathbf{r}_q$ have similar real parts and similar but opposite sign imaginary parts.

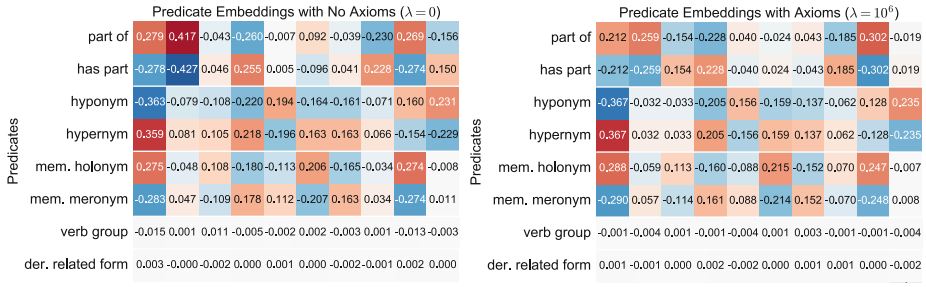


Fig. 2. WordNet predicate embeddings learned using the TransE model, with $k = 10$ and regularization weight $\lambda = 0$ (left) and $\lambda = 10^6$ (right) – embeddings are represented as a heatmap, with values ranging from larger (red) to smaller (blue). Note that, assuming the axiom $p \equiv q^-$ holds, using the proposed method leads to predicate embeddings such that $\mathbf{r}_p \approx -\mathbf{r}_q$. (Color figure online)

A similar phenomenon in Fig. 1 (right), where predicated embeddings have been trained using ComplEx: we can see that the model is naturally inclined to assign complex conjugate embeddings to inverse relations and, as a consequence, nearly-zero imaginary parts to the embeddings of symmetric predicates – since it is the only way of ensuring $\mathbf{r}_p \approx \overline{\mathbf{r}_p}$. However, we can enforce such relationships explicitly by means of model-specific regularizers, for increasing the predictive accuracy and generalization abilities of the models.

Table 3. Average number of seconds required for training.

	Plain	Regularized
WordNet	31.7 s	32.0 s
DBpedia	57.9 s	58.5 s
YAGO3	220.7 s	221.3 s

We also benchmarked the computational overhead introduced by the novel regularizers by timing the training time for unregularized (plain) models and for

regularized ones – results are available in Table 3. We can see that the proposed method for leveraging background schema knowledge during the learning process adds a negligible overhead to the optimization algorithm – less than 10^{-1} s per epoch.

8 Conclusions and Future Works

In this work we introduced a novel and scalable approach for leveraging background knowledge into neural knowledge graph embeddings. Specifically, we proposed a set of background knowledge-driven regularizers on the relation embeddings, which effectively enforce a set of desirable algebraic relationships among the distributed representations of relation types. We showed that the proposed method improves the generalization abilities of all considered models, yielding more accurate link prediction results without impacting on the scalability properties of neural link prediction models.

Future Works

A promising research direction consists in leveraging more sophisticated background knowledge – *e.g.* in the form of First-Order Logic rules – in neural knowledge graph embedding models. This can be possible by extending the model in this paper to regularize over subgraph pattern embeddings (such as *paths*), so to leverage relationships between such patterns, rather than only between predicates. Models for embedding subgraph patterns have been proposed in the literature – for instance, see (Niepert 2016; Guu et al. 2015). For instance, it can be possible to enforce an equivalency between the path $\text{PARENTOF} \circ \text{PARENTOF}$ and GRANDPARENTOF, effectively incorporating a First-Order rule in the model, by regularizing over their embeddings.

Furthermore, a future challenge is also extending the proposed method to more complex models, such as ER-MLP (Dong et al. 2014), and investigating how to mine rules by extracting regularities from the latent representations of knowledge graphs.

Acknowledgements. This work was supported by the TOMOE project funded by Fujitsu Laboratories Ltd., Japan and Insight Centre for Data Analytics at National University of Ireland Galway (supported by the Science Foundation Ireland grant 12/RC/2289).

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52

Baroni, M., Bernardi, R., Do, N-Q., Shan, C.: Entailment above the word level in distributional semantics. In: EACL, pp. 23–32. The Association for Computer Linguistics (2012)

Bollacker, K.D., Cook, R.P., Tufts, P.: Freebase: a shared database of structured general human knowledge. In: AAAI, pp. 1962–1963. AAAI Press (2007)
Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Mach. Learn. **94**(2), 233–259 (2014)
Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI. AAAI Press (2010)
Chang, K.-W., Yih, W., Yang, B., Meek, C.: Typed tensor decomposition of knowledge bases for relation extraction. In: EMNLP, pp. 1568–1579. ACL (2014)
Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: KDD, pp. 601–610. ACM (2014)
Duchi, J.C., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
García-Durán, A., Bordes, A., Usunier, N.: Effective blending of two and three-way interactions for modeling multi-relational data. In: ECML-PKDD, pp. 434–449 (2014)
Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: EMNLP, pp. 318–327. The Association for Computational Linguistics (2015)
Hayes, P., Patel-Schneider, P.: RDF 1.1 semantics. W3C recommendation, W3C, February 2014. http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/
Jenatton, R., Roux, N.L., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: NIPS, pp. 3176–3184 (2012)
Krompass, D., Nickel, M., Tresp, V.: Large-scale factorization of type-constrained multi-relational data. In: DSAA, pp. 18–24. IEEE (2014)
Krompaß, D., Nickel, M., Tresp, V.: Querying factorized probabilistic triple databases. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 114–129. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_8
Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9366, pp. 640–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_37
Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: a knowledge base from multilingual wikipedias. In: CIDR (2015). www.cidrdb.org
Meseguer, P., Rossi, F., Schiex, T.: Soft constraints. In: Handbook of Constraint Programming, of Foundations of Artificial Intelligence, vol. 2, pp. 281–328. Elsevier (2006)
Miller, G.A.: WordNet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
Minervini, P., d'Amato, C., Fanizzi, N., Esposito, F.: Leveraging the schema in latent factor models for knowledge graph completion. In: SAC, pp. 327–332. ACM (2016)
Nickel, M., Tresp, V., Kriegel, H.-P.: A three-way model for collective learning on multi-relational data. In: ICML, pp. 809–816 (2011)
Nickel, M., Tresp, V., Kriegel, H.-P.: Factorizing YAGO: scalable machine learning for linked data. In: WWW, pp. 271–280. ACM (2012)
Nickel, M., Jiang, X., Tresp, V.: Reducing the rank in relational factorization models by including observable patterns. In: NIPS, pp. 1179–1187 (2014)
Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)

Niepert, M.: Discriminative Gaifman models. In: NIPS, pp. 3405–3413 (2016)
Rocktäschel, T., Singh, S., Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction. In: HLT-NAACL, pp. 1119–1129. The Association for Computational Linguistics (2015)
Schneider, M.: OWL 2 web ontology language RDF-based semantics, 2nd edn. W3C recommendation, W3C, December 2012. http://www.w3.org/TR/2012/REC-owl2-rdf-based-semantics-20121211/
Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: WWW, pp. 697–706. ACM (2007)
Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: ICML, of JMLR Workshop and Conference Proceedings, vol. 48, pp. 2071–2080. JMLR. org (2016)
Wang, Q., Wang, B., Guo, L.: Knowledge base completion using embeddings and rules. In: IJCAI, pp. 1859–1866. AAAI Press (2015)
Yang, B., Yih, W-t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015, May 2015

Chapter 9

Prediction of Adverse Drug Reactions

Briefings in Bioinformatics, 2017, 1–13

doi: 10.1093/bib/bbx099
Paper

Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models

Emir Muñoz, Vít Nováček and Pierre-Yves Vandenbussche

Corresponding author: Emir Muñoz, Fujitsu Ireland Ltd., Insight Building, IDA Business Park, Lower Dangan, Newcastle, Galway, Ireland. E-mail: emir.munoz@ie.fujitsu.com or emir.munoz@gmail.com

Abstract

Timely identification of adverse drug reactions (ADRs) is highly important in the domains of public health and pharmacology. Early discovery of potential ADRs can limit their effect on patient lives and also make drug development pipelines more robust and efficient. Reliable *in silico* prediction of ADRs can be helpful in this context, and thus, it has been intensely studied. Recent works achieved promising results using machine learning. The presented work focuses on machine learning methods that use drug profiles for making predictions and use features from multiple data sources. We argue that despite promising results, existing works have limitations, especially regarding flexibility in experimenting with different data sets and/or predictive models. We suggest to address these limitations by generalization of the key principles used by the state of the art. Namely, we explore effects of: (1) using knowledge graphs—machine-readable interlinked representations of biomedical knowledge—as a convenient uniform representation of heterogeneous data; and (2) casting ADR prediction as a multi-label ranking problem. We present a specific way of using knowledge graphs to generate different feature sets and demonstrate favourable performance of selected off-the-shelf multi-label learning models in comparison with existing works. Our experiments suggest better suitability of certain multi-label learning methods for applications where ranking is preferred. The presented approach can be easily extended to other feature sources or machine learning methods, making it flexible for experiments tuned toward specific requirements of end users. Our work also provides a clearly defined and reproducible baseline for any future related experiments.

Key words: adverse drug reactions (ADR); drug similarity; knowledge graphs; multi-label learning

Introduction

Adverse drug reactions (ADRs) can cause significant clinical problems and represent a major challenge for public health and the pharmaceutical industry. During a drug development process, pharmacology profiling leads to the identification of potential drug-induced biological system perturbations including primary effects (intended drug–target interactions) as well as secondary effects (off-target–drug interactions) mainly responsible for ADRs [1]. Many ADRs are discovered during preclinical and clinical trials before a drug is released on the market. However, the use of a registered drug within a large population (demonstrating a wider range of clinical genotypes and phenotypes than considered in the clinical trials) can result in serious ADRs that have not been identified before. This has a large impact on patient safety and quality of life, and also has significant financial consequences for the pharmaceutical industry [2].

Emir Muñoz is a PhD student at Insight Centre for Data Analytics, National University of Ireland Galway, and a Researcher at Fujitsu Ireland Ltd. His main interests lie within the areas of databases and machine learning. He is currently focused on representational learning and knowledge graphs mining.
Vít Nováček holds a PhD from National University of Ireland, Galway. Vit has background in NLP, Semantic Web and knowledge representation, and his current research revolves around knowledge discovery from biomedical texts and data. He works as a project leader at the Insight Centre for Data Analytics in Galway.
Pierre-Yves Vandenbussche holds a PhD in Information Technology from Paris VI University. Currently leading the Knowledge Engineering and Discovery research team in Fujitsu Ireland working with the Insight Centre, his research interest concerns methods to improve semantic data representation, knowledge extraction and knowledge graph mining.
Submitted: 12 April 2017; **Received (in revised form):** 17 July 2017

The result of a recent review of epidemiological studies in Europe states that 3.5% of hospital admissions are because of ADRs and 10% of patients experience an ADR during their hospitalization [3]. ADRs are a major cause of morbidity (and associated reduction of quality of life) and mortality [4, 2]. Recent estimates set the number of yearly drug-induced fatalities to 100 000 in the United States and almost 200 000 in Europe, making it the fourth cause of death before pulmonary diseases or diabetes [5, 3]. In addition to the significance for the public health, ADRs are associated with an important economic burden imposed for public health systems and pharmaceutical industry. The extra costs are caused mainly by the withdrawal of dangerous drugs from the market, litigations and further hospitalizations to treat the adverse effects. The annual cost of ADRs in the United States is estimated at $136 billion [6].

Any improvements in the early identification of ADRs can decrease the high attrition rate in the drug discovery and development process. After the drug registration, better prediction of ADRs can alleviate associated clinical problems and decrease the adverse effect-induced extra costs. *In silico* approaches to predict ADRs of candidate drugs are now commonly used to complement costly and time-consuming *in vitro* methods [7]. Computational methods differ by the drug development/deployment stage they are applied at, and by the features used for the prediction of ADRs. Pharmacovigilance systems (monitoring the effects of drugs after they have been licensed for use) mine statistical evidence of ADRs from spontaneous reports by physicians, such as the Food and Drug Administration (FDA) Adverse Event Reporting System (FAERS) [8–10]; from patient records [11]; or more recently from non-traditional sources, such as logs of search engine activity or social media [12, 13]. While these methods limit the risk of serious public health issues by identifying early occurrences of ADRs, they assume that such adverse effects are already demonstrated within a population.

Computational prediction of ADRs during the development cycle of a drug (before the drug is licenced for use) can reduce the cost of drug development and provide a safer therapy for patients [14]. Most state-of-the-art techniques adopt a drug-centric approach and rely on the assumption that similar drugs share the same properties, mechanism of action and therefore also ADRs [15, 16] (there are also methods that focus on the ADR information to overcome certain specific problems like drugs with little or no features at all, or ADRs with low number of related drugs [15, 9]. The methods are, however, less numerous and also harder to evaluate in a comparative manner). Predictions of new ADRs are then based on a drug–drug similarity network. In most of the early works, this network was based on the similarity of the substructures within the active ingredients of drugs [17–20]. More recent approaches combine data covering both chemical space of drugs and biological interaction-based features such as drug target, pathways, enzymes, transporters or protein–protein interactions [21–23]. Lately, integrative methods take into account also phenotypic observation-based features such as drug indications [24–27]. The availability of multi-source structured data has allowed for integration of complementary aspects of drugs and their links to side effects leading to higher accuracy [28].

The scope of this review is given by recent state-of-the-art methods (from 2011 on) that satisfy two key requirements. First, we consider methods that take advantage of multi-source structured data. Secondly, we focus on techniques that use machine learning to predict the likelihood of a side effect being caused by a given drug (drug-centric approach). Table 1 lists the reviewed approaches along with the features they use.

Table 1. Multi-source feature sets used by state-of-the-art methods

Feature space	Atias and Sharan, 2011 [17]	Pauwels *et al.* (2011) [18]	Mizutani *et al.* (2012) [21]	Yamanishi *et al.* (2012) [22]	Liu *et al.* (2012) [24]	Bresso *et al.* (2013) [19]	Huang *et al.* (2013) [23]	Jahid and Ruan (2013) [20]	Zhang *et al.* (2015) [25, 28, 26]	Rahmani *et al.* (2016) [29]	Muñoz *et al.* (2016) [27]
Chemical space											
Drug compound substructure	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Biological space											
Drug target				✓	✓		✓		✓	✓	✓
Pathway			✓		✓				✓		✓
Enzymes					✓				✓		✓
Transporters					✓				✓		✓
Protein-protein interaction (PPi)							✓			✓	
Phenotypic space											
Indication					✓				✓		✓
Cell line response	✓										

While many of the state-of-the-art approaches produce results that have great potential for being used in drug development pipelines, there are still things to improve. A limitation that is most relevant as a motivation for the presented review is the lack of flexibility that prevents users who are not proficient in machine learning from easily using the predictive models. This makes it difficult for people like biologists, pharmacologists or clinicians to experiment with data and models fine-tuned towards their specific requirements on the ADR prediction (such as increasing the sensitivity or specificity of the model). The main issue of existing models is that they typically work with data sets that have been manually preprocessed, and the particular prediction methods are adapted to the experimental data in a focused manner.

We review the key points and limitations of existing approaches and introduce their generalization based on: (1) tapping into many diverse interlinked knowledge bases (i.e. knowledge graphs) related to drugs and adverse effects that substantially limits the manual effort required for data integration and feature engineering. (2) Rigorous formulation of the ADR prediction problem as a multi-label learning-to-rank problem that allows for easy experimentation with many off-the-shelf machine learning models.

We show that specific applications of these two principles can lead to performance comparable with existing methods. Moreover, the proposed approach produces ranked predictions by default, with many relevant predictions present at the top of the ranked result lists. This is potentially useful in scenarios where experts (e.g. pharmaceutical researchers or public health officials) have limited time and/or resources, and thus they can only process a few prediction candidates out of many possibilities (there can often be hundreds of predictions for a single drug).

The main contributions of this work are as follows. We propose a specific way of using knowledge graphs to generate different feature sets for ADR prediction and demonstrate the favourable performance of selected off-the-shelf multi-label learning models in comparison with existing works. In addition to that, we show how the approach can be easily extended to other feature sources or machine learning methods. This makes the method flexible for experiments tuned towards specific requirements of end users. Our results and data also provide a clearly defined and reproducible baseline for any future related experiments.

Materials

Various publicly available data sources can be used to define similarity between drugs [14]. Each data source describes a specific aspect of the pharmacological space of a drug such as its chemical, biological or phenotypic properties. For instance, SIDER database [30] presents information of side effects and indication for marketed drugs. PubChem Compound data [31] contain chemical structure description of drugs. DrugBank [32] provides detailed information about drugs such as their binding proteins and targets, enzymes or transporters, thus informing on drugs' mechanism of action and metabolism. KEGG Genes, Drug, Compound and Disease databases [33] describe further information about molecular interaction of drugs and their signalling pathways.

In the following, we review the materials—results of data integration using multiple data sources, provided by the authors of the state-of-the-art methods. Because previous data integration activities were expensive and mostly carried out manually, here, we propose a different data source and representation, which can be considered a superset of all previous data sets used. This data source is represented using a graph database, a model in which it is simpler to integrate different data sources such as the ones already mentioned. We also provide an algorithm to generate the required drugs' profile, similarly to the ones provided by the reviewed methods (Supplementary Section D). For comparisons, we use Liu's data set [24] and Zhang *et al.* [25] data set termed 'SIDER 4' as benchmarks. As presented in Table 1, Liu's data set contains six types of features covering the chemical, biological and phenotypic spaces of drugs combined with information on their associated ADRs (cf. Table 2). We use this data set as primary means to compare the reviewed methods. SIDER 4 data set introduced by Zhang *et al.* [25] is an update of Liu's data set integrating the fourth version of SIDER. This data set is interesting, as it introduces newly approved drugs for which fewer post-market ADR have been detected. We use the SIDER 4 data set as secondary means to compare the methods.

Table 2. The data set characteristics

Data set	Number of drugs	Number of side effects
Liu's data set	832	1385
Bio2RDF data set	1824	5880
SIDER 4 data set	1080	5579
Aeolus data set	750	181

A new alternative multi-source graph data have recently become via the Bio2RDF project [34]. Bio2RDF publishes the pharmacological databases used in many ADR prediction experiments in the form of a knowledge graph—a standardized, interlinked knowledge representation based on labelled relationships between entities of interest. Bio2RDF data were first used for the prediction of ADRs by Muñoz *et al.* [27], where drug similarities were computed by measuring the shared connections between drugs in the graph. Here, we build on top of that and evaluate the use of the BioRDF knowledge graph as a means to facilitate the generation of more expressive features for computing similarity between drugs. Such automatically generated data can be used to replace or enrich existing manually integrated feature sets, and be used to evaluate prediction methods as per normal machine learning pipelines.

Finally, to get another perspective for interpreting the evaluation results, we use the FDA FAERS [8, 10]. FAERS publishes recent ADR reports coming from population-wide post-marketing drug effect surveillance activities. Extracting the most recent ADRs for newly marketed drugs helps us to evaluate the ability of various methods to predict ADRs of drugs after their release on the market. We extract this information from the Aeolus data set [35], which is a curated and annotated, machine-readable version of the FAERS database. We use Aeolus to generate an updated version of the SIDER 4 data set that includes also the latest ADRs as observed in the population.

For details on the generation of Liu's data set [24] and the SIDER 4 data set [25], we refer the readers to the original articles. We will now detail the construction of the 'Bio2RDF data set' and the 'Aeolus data set'.

Bio2RDF data set

The Bio2RDF project (http://bio2rdf.org/) aims at simplifying the use of publicly available biomedical databases by representing them in a form of an interconnected multigraph [34, 36].

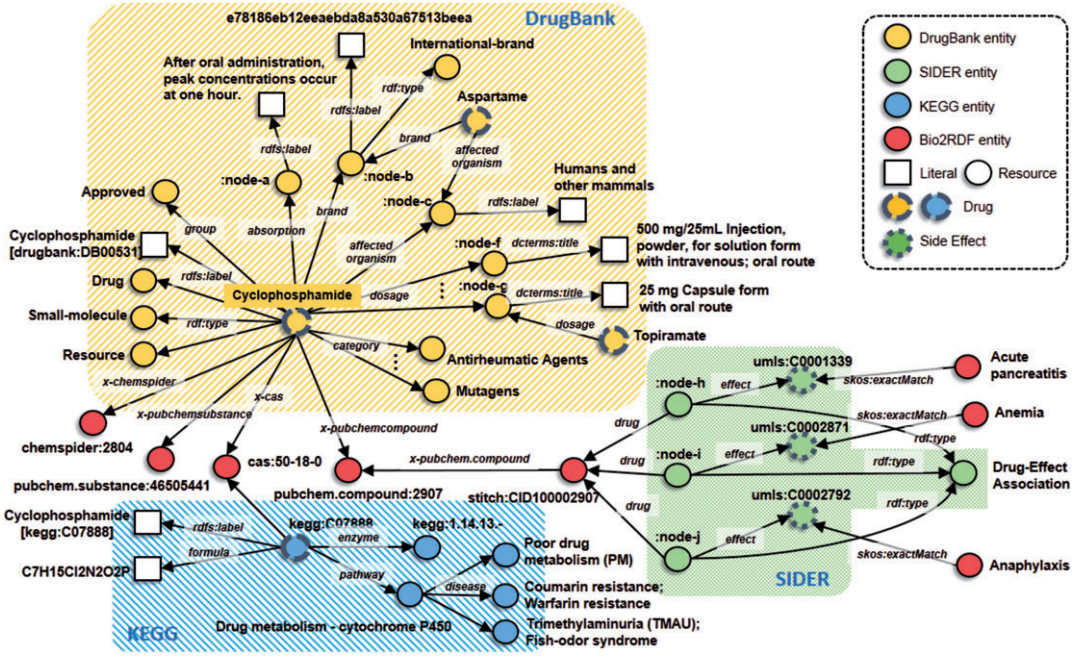


Figure 1. A Bio2RDF fragment around the cyclophosphamide drug, showing the connections between three main databases: DrugBank, SIDER and KEGG.

The project provides a set of scripts (https://github.com/bio2rdf) to convert from the typically relational or tabular databases (e.g. DrugBank; SIDER) to a more flexible triple-based RDF (Resource Description Framework) format. The project also makes available the output of their conversions, and in its version 4, published in December 2015, Bio2RDF represented 30 databases including PubChem, DrugBank, SIDER and KEGG, among others with valuable drug-related information. Each information such as drug, protein or side effect is represented as a node entity with a unique identifier, and each relation between them as an edge with a qualified label, generating a network of great value for bioinformatics [37]. Here, we use the release 4 of Bio2RDF and represent its data using a knowledge graph $\mathcal{G} = \{\langle s, p, o\rangle\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, which is a set of $\langle s, p, o\rangle$ triples each consisting of a subject s, a predicate p and an object o, and encoding the statement 's has a relationship p with o'. The subject and object $s, o \in \mathcal{E}$ are entities, $p \in \mathcal{R}$ is a relation type and $\mathcal{E}$, $\mathcal{R}$ denote the sets of all entities and relation types in the knowledge graph, respectively. Figure 1 shows a fragment of the Bio2RDF knowledge graph that integrates three databases, namely, DrugBank, SIDER and KEGG. Usually, connections between databases are made using identifiers such as PubChem compound or Chemical Abstracts Service (CAS) number. Notice that a knowledge graph can also be built from the original databases using different methods or scripts, and here, we select Bio2RDF because it already contains mappings for most of the relevant databases.

A knowledge graph $\mathcal{G}$ can contain biomedical facts [note that the URIs in the examples are used as unique identifiers; the availability of corresponding records through an HTTP request (such as in a browser) depends on a third-party service (Bio2RDF.org)] such as:

Table 3. Number of $\langle s, p, o\rangle$ triples in the Bio2RDF data set used in our experiments

Data set	Type of information	Number of triples
DrugBank	Drug types, chemical information	5 151 999
SIDER	Side effects of drugs	5 578 286
KEGG	Drugs, genes and pathway maps	4 387 541

⟨http://bio2rdf.org/drugbank:DB00531, *label*, "Cyclophosphamide"⟩

or

⟨http://bio2rdf.org/drugbank:DB00531, *enzyme*, http://bio2rdf.org/kegg:1.14.13.-⟩.

This format allows for easy representation of equivalences or external links between data sources as an additional relation/edge. For instance, a relationship between a DrugBank drug and a PubChem compound can be expressed as:

⟨http://bio2rdf.org/drugbank:DB00531, *x-pubchemcompound*, http://bio2rdf.org/pubchem.compound:2907⟩.

By simply merging multiple data sources from Bio2RDF, we are able to build an integrated knowledge graph with links between databases materialized. During the integration, the PubChem compound of a drug is used to link DrugBank and SIDER, while the CAS number is used to link DrugBank and KEGG. This flexibility for generating training and testing data is currently impossible with the manual integration pipelines used by the reviewed methods. In our experiments, we shall use a knowledge graph integrating the DrugBank, SIDER and KEGG databases (cf. Table 3).

Aeolus data set

Aeolus [35] is a curated and standardized adverse drug events resource meant to facilitate research in drug safety. The data in Aeolus come from the publicly available US FDA FAERS, but is extensively processed to allow for easy use in experiments. In particular, the cases (i.e. ADR events) in the FAERS reports are deduplicated and the drug and outcome (i.e. effect) concepts are mapped to standard vocabulary identifiers (RxNorm and SNOMED-CT, respectively). A similar approach for extracting ADR terms from FDA-approved drug labels was applied in [38] to group similar drugs by topics. However, Aeolus is preferred because of its curated status.

The Aeolus data set is presented in a convenient comma-separated values (CSV) format, from which we can easily extract pairs of drugs and their adverse effects ranked by the statistical significance of their occurrences within the FAERS reports. We map the identifiers for drugs and for adverse effects in Aeolus to the ones in DrugBank, which are used in our experiments. This means that we are able to use the FDA FAERS data as an additional manually curated resource for validating any adverse effect prediction method, as detailed later on in the description of our experiments.

Methods

In this section, we present details of the reviewed approaches for ADR prediction, on the basis of a multi-label learning setting.

Multi-label learning framework

As a drug can generally have multiple adverse reactions, the ADR prediction can be naturally formulated as a multi-label learning problem [39]. Multi-label learning addresses a special variant of the classification problem in machine learning, where multiple labels (i.e. ADRs) are assigned to each example (i.e. drug). The problem can be solved either by transforming the multi-label problem into a set of binary classification problems or by adapting existing machine learning techniques to the full multi-label problem (see https://en.wikipedia.org/wiki/Multi-label_classification for more details and a list of examples).

Most of the current ADR prediction methods, however, do not fully exploit the convenient multi-label formulation, as they simply convert the main problem into a set of binary classification problems [40]. This is problematic for two main reasons. First, transforming the multi-label problem into a set of binary classification problems is typically computationally expensive for large numbers of labels (which is the case in predicting thousands of ADRs). Secondly, using binary classifiers does not accurately model the inherently multi-label nature of the main problem. We validate these two points empirically in 'Results and discussion' section. Here, we follow the philosophy of algorithm adaptation: fit algorithms to data [40].

Yet, there are exceptions, such as the work in [25], presenting the multi-label learning method FS-MLKNN that integrates feature selection and k-nearest neighbours (kNN). Unlike most previous works, Zhang *et al.* [25] propose a method that does not generate binary classifiers per label but uses an ensemble learning instead. (We shall provide more details of this and other methods in 'Learning models' section.) Also, Muñoz *et al.* [27] proposed a multi-label solution for the prediction problem using a constrained propagation of ADRs between neighbouring drugs, making clear the benefits of a graph structure of data (cf. Supplementary Section F).

In the following, we formalize the learning framework with Q-labels as in [41, 42]. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be the instance space of N different data points (i.e. drugs) in $\mathbb{R}^d$, and let $\mathcal{Y} = \{y_1, y_2, \ldots, y_Q\}$ be the finite set of labels (i.e. ADRs). Given a training set $\mathcal{D} = \{(\mathbf{x}_i, Y_i) | 1 \leq i \leq N\}$, where $\mathbf{x}_i \in \mathcal{X}$ is a d-dimensional drug feature vector $[x_{i1}, x_{i2}, \ldots, x_{id}]^\top$ and $Y_i \in 2^{\mathcal{Y}}$ is a vector of labels associated with $\mathbf{x}_i$, the goal of the learning system is to output a multi-label classifier $h : \mathcal{X} \to 2^{\mathcal{Y}}$, which optimizes some specific evaluation metric. In most cases, however, the learning system will not output a multi-label classifier but instead will produce a real-valued function (aka. regressor) of the form $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $f(\mathbf{x}, y)$ can be regarded as the confidence of $y \in \mathcal{Y}$ being a proper label of $\mathbf{x}$. It is expected that for a given instance $\mathbf{x}$ and its associated label set Y, a successful learning system will tend to output larger values for labels in Y than those not in Y, i.e. $f(\mathbf{x}, y_1) > f(\mathbf{x}, y_2)$ for any $y_1 \in Y$ and $y_2 \notin Y$. In other words, the model should consistently be more 'confident' about true positives (actual ADRs) than about false positives. Intuitively, the regressor $f(\cdot, \cdot)$ can be transformed into a ranking function $rank_f(\cdot, \cdot)$, which maps the outputs of $f(\mathbf{x}, y)$ for any $y \in \mathcal{Y}$ to $\{y_1, y_2, \ldots, y_Q\}$ such that if $f(\mathbf{x}, y_1) > f(\mathbf{x}, y_2)$, then $rank_f(\mathbf{x}, y_1) < rank_f(\mathbf{x}, y_2)$. The ranking function can naturally be used for instance for selecting top-k predictions for any given drug, which can be useful in cases where only limited numbers of prediction candidates can be further analysed by the experts.

Our learning problem can now be formally stated as: given a drug $\mathbf{x}$ and a finite-size vector Y with its initially known adverse reactions (i.e. labels) seek to find a discriminant function $f(\mathbf{x}, Y) = \widehat{Y}$, where $\widehat{Y}$ is a finite-size vector representation of the labelling function $\widehat{Y} = [f(\mathbf{x}, y_1), \ldots, f(\mathbf{x}, y_Q)]^\top$ for $y_i \in \mathcal{Y}$. For instance, headache (C0018681) and vomiting (C0042963) are common adverse reactions of atomoxetine (DB00289), a drug used for the treatment of attention deficit hyperactivity disorder, and they should be ranked higher than conjunctivitis (C0009763) or colitis (C0009319), which are rare or unregistered ADRs for atomoxetine (cf. Supplementary Section E for features manipulation guidelines).

Learning models

To complement most previous works, we formulate ADR prediction as a multi-label ranking problem, and train different machine learning models. This allows for approaching the problem more naturally in many practical use cases, such as when one prefers to explore only a few, i.e. the most relevant adverse effect candidates out of many possible for a certain drug. Multi-label learning models learn how to assign sets of ADRs/labels to each drug/example. The main motivation for our model choices was to have a representative sample of the different multi-label learning families described in the machine learning literature (ranging from decision trees through instance-based learning or regression to neural networks). Such an approach demonstrates the broad range of possibilities when adopting off-the-shelf models. We investigate state-of-the-art multi-label learning models, namely, decision trees, random forests, kNN and multilayer perceptron. We also investigate the use of logistic regression binary classifiers for multi-label following the one-vs-all strategy in which the system builds as many binary classifiers as input labels, where samples having label y are considered as positive, and negative otherwise (cf. Supplementary Section B for a description of each model).

Among the methods for predicting ADRs that accept multi-source data are Liu's method, FS-MLKNN (feature selection-based multi-label k-nearest neighbour) [25], the linear neighbourhood similarity methods (LNSMs) with two different data integration approaches, similarity matrix integration (LNSM-SMI) and cost minimization integration (LNSM-CMI) [28] and, finally, knowledge graph similarity propagation (KG-SIM-PROP) [27]. Liu *et al.* [24] proposed a multi-source method using chemical, biological and phenotypic information about drugs and built an SVM classifier for each ADR. FS-MLKNN is a method that simultaneously determines critical feature dimensions and builds multi-label prediction models. An FS-MLKNN model is composed of five MLKNN models constructed from a selected subset of features selected using a genetic algorithm. In the learning step, the LNSM-SMI method generates K similarity matrices from K different data sources and combines them using θ_i weights (for all $1 \leq i \leq K$), while the LNSM-CMI learns the LNSM independently on each data source. LNSM is itself a method that can train models and make predictions based on single-source data, and takes the assumption that a data point (i.e. drug) can be optimally reconstructed by using a linear combination of its neighbours. Because of this, LNSM methods usually require a large number of neighbours to deliver better results. Both LNSM-SMI and LNSM-CMI are formulated as convex optimization problems using the similarity between drugs to later make predictions. On the other hand, KG-SIM-PROP [27] proposes to exploit a graph structure built from the similarity matrix of drugs to propagate ADR labels from one drug to other drugs in its neighbourhood. Later, we will see that such propagation, unlike LNSM-based methods, requires a smaller number of neighbours to deliver efficient predictions. KG-SIM-PROP has been modified to not limit the number of predictions as stated in [27], and adopt the evaluation protocol defined for this review, ensuring a fair comparison with the other models.

A comparative review of existing multi-source machine learning models and selected off-the-shelf multi-label learning models trained on knowledge graphs allows for assessing not only the performance but also the flexibility of the various approaches. The performance of the off-the-shelf methods can also be used as a baseline for more focused experiments in ADR prediction, which is something that has been missing before. An additional contribution of this review is the analysis of the model performance not only on the hand-crafted feature sets used by existing approaches but also on drug features automatically extracted from knowledge graphs (cf. Supplementary Section E). This is to demonstrate the feasibility of this particular approach to increasing practical applicability of automated ADR prediction pipelines.

We perform a comparison of the above models (Liu's method; FS-MLKNN; LNSM-SMI; LNSM-CMI; KG-SIM-PROP; decision trees; random forests; multi-layer perceptron; linear regression) in terms of performance based on several multi-label ranking evaluation metrics. All models are given a design matrix $\mathbf{X}$ with binary features as input, where the row i of $\mathbf{X}$ represents drug-i using a vector $\mathbf{x}_i$ (for $1 \leq i \leq N$) with a 1 or 0 in column j to indicate whether drug-i has feature j (for $1 \leq j \leq d$), respectively. In the same way, labels are represented using a binary matrix $\mathbf{Y}$, where row i contains either 1 or 0 in column j indicating whether drug-i has ADR-j, respectively. For instance, considering the following three features: ($j = 0$) enzyme P05181, ($j = 1$) indication abdominal cramp (C0000729) and ($j = 2$) pathway hsa00010, we can have the vector $\mathbf{x}_1 = [1, 0, 1]$ for the drug fomepizole (DB01213), meaning that fomepizole interacts with enzyme P05181, is not used to treat abdominal cramps and is part of pathway hsa00010.

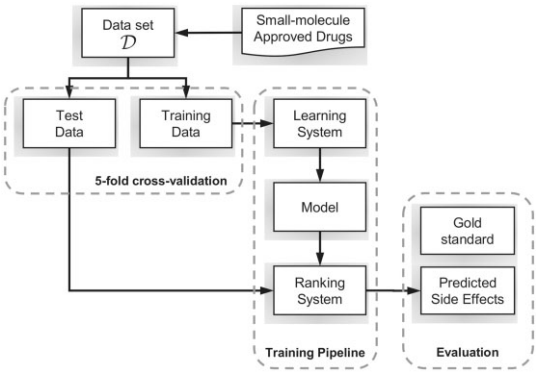


Figure 2. Machine learning flow chart for training and testing of a model.

In Figure 2, we show a typical flow chart for the processes of training, testing and evaluating machine learning models. For a given model, its output is used to generate ADR predictions. These predictions are evaluated using Liu's, SIDER 4 and Aeolus data sets as gold standards.

Most of the reviewed models work directly with the drug feature matrices. However, two models, namely, KG-SIM-PROP and kNN, require a similarity graph as input, which in this case is generated from the similarity between drugs using either the original data sets features or the Bio2RDF knowledge graph. Such a similarity graph encodes the similarity relations between drugs, where the value of the i-th row with the j-th column is the similarity score between drug-i and drug-j. In Supplementary Section A, we describe a method to generate such similarity network of drugs from a knowledge graph.

Results and discussion

Experimental configuration and evaluation metrics

All five multi-label learning models plus KG-SIM-PROP were implemented using the Scikit-Learn Python package [43] (http://scikit-learn.org/stable/), whereas, when available, we use the implementations provided by the reviewed methods. (General details on training and using the models are provided in the Supplementary Section B.) In many cases, we used the default hyper-parameters values, as our main focus was to compare the performance of different models and not to find the optimal hyper-parameter settings for each of them. Some specific hyper-parameters, however, proved to have an obvious impact on the model results, and therefore, we changed some of the default values and performed limited hyper-parameter optimization via grid search. In particular: (a) the KG-SIM-PROP model uses the 3w-Jaccard similarity metric [44], with 10–100 neighbours size; (b) the kNN model is tested with 10–100 neighbours, with uniform and distance-based weights using the Minkowski, Manhattan, Chebyshev, Jaccard and Rogers Tanimoto distance metrics [44]; (c) the decision trees and random forests models use the mean squared error criterion, which is the only one supporting multi-label learning; (d) the multi-layer perceptron model is set with a unique hidden layer with 64, 128, 256 and 512 hidden units, a batch size equals to the 20% of drugs (which was chosen from an independent grid search), a logistic sigmoid activation and the Adam solver; (e) the logistic regression model uses a L_2 penalty function, $C = 1.0$, stochastic average gradient as solver and 200 maximum iterations.

Table 4. Predictive power of the models using Liu's data set

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
Liu's method [24]	0.2610	0.2514	0.8850	0.0927	0.9291	837.4579
FS-MLKNN [28]	0.5134	0.4802	**0.9034**	0.0703	0.1202	795.9435
LNSM-SMI [28]	**0.5476**	**0.5053**	0.8986	0.0670	**0.1154**	789.8486
LNSM-CMI [28]	0.5329	0.4909	0.9091	**0.0652**	0.1250	**776.3053**
KG-SIM-PROP [27]	0.4895±0.0058	0.4295±0.0078	0.8860±0.0075	0.1120±0.0139	0.1610±0.0164	1100.9985±65.8834
kNN	0.5020±0.0078	0.4417±0.0081	0.8892±0.0085	0.1073±0.0053	0.1538±0.0181	1102.3548±41.4641
Decision trees	0.2252±0.0137	0.1989±0.0181	0.6634±0.0316	0.6519±0.0242	0.5493±0.0374	1377.1316±8.3936
Random forests	0.4626±0.0163	0.4331±0.0261	0.8342±0.0218	0.2525±0.0176	0.2007±0.0154	1284.3111±27.0454
Multi-layer perceptron	0.5196±0.0069	0.4967±0.0204	0.9003±0.0057	0.0874±0.0009	0.1454±0.0166	954.0372±22.2870
Linear regression	0.2854±0.0088	0.2595±0.0196	0.6724±0.0232	0.6209±0.0137	0.4267±0.0103	1380.0763±4.0209

Note: For each metric, we report the SD values (when available). The values for the first four models were taken from [28]. The evaluation metrics are AP, AUC-PR curve, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.) Bold values represent the best performing methods across a given metric.

To compare all the models, we adopt common metrics for ranking in multi-label learning [40]. We also compute example-based ranking metrics [40] used in related works, namely, one-error (One-error), coverage (Cov-error), ranking loss (R-loss) and average precision (AP). Summary and details of all metrics we use are given in the Supplementary Section C. The performance of all models is evaluated using a 5-fold cross-validation. First, all drugs are randomly split into five equal sized subsets. Then, for each of the k folds, one part is held out for testing, and the learning algorithm is trained on the remaining four parts. In this way, all parts are used exactly once as validation data. The selection of the best hyper-parameters for each model is performed in each fold on the training set during the 5-fold cross-validation, and the best model is applied over the test set for validation (cf. Supplementary Section G). The five validation results are then averaged over all rounds. We also use common evaluation metrics for ranking systems, the area under the receive operator curve (AUC-ROC) and the area under the precision–recall curve (AUC-PR) (as defined by Davis and Goadrich in [45], when dealing with highly skewed data sets) to evaluate the models because they can be used to evaluate models, regardless of any threshold. However, because of the existing unbalance of the labels (i.e. an ADR is more commonly found as a negative value than as a positive one among drugs), the AUC-PR gives a more informative picture of the model's performance [45]. Thus, we set the AUC-PR as our target metric (in the grid searches) for each of the rounds. Additionally, we compute other example-based metrics [46, 40], namely, AP, one error, Cov-error and R-loss. The last type of measures we use are the general ranking evaluation metrics Hits at K (Hits@K) and Precision at K (P@K). Among the measures we used, the Hits@K and P@K are arguably the most accurate scores in terms of evaluating the benefit of ADR discovery for certain types of end users like clinical practitioners. As explained in [47], these scores are easily grasped by non-informaticians and are therefore apt for explaining the reliability of the system to them. Moreover, in settings where quick decisions are needed, like in clinical practice, users do not tend to perform comprehensive search among many possible alternatives to find the relevant ones [47]. The Hits@K and P@K scores reflect the likelihood that such users will find relevant results quickly at the top of the list of possibly relevant results.

Comparison on Liu's data set

In this section, we present the evaluation of all methods using Liu's data set, which includes multi-source data with different types of features about drugs. Specifically, we compare the methods considering the features and labels in Liu's data set, which was introduced in [24] and has been considered as a benchmark in [25, 28].

We compare the results reported in [28] for four existing methods (Liu's method, FS-MLKNN, LNSM-SMI and LNSM-CMI) with the KG-SIM-PROP [27] and the five off-the-shelf multi-label learning models selected by us. Table 4 shows the values of evaluation metrics for each model, highlighting the best-performing methods per metric in bold. We found out that the methods FS-MLKNN, LNSM-SMI and LNSM-CMI proposed by Zhang *et al.* recently [25, 28] perform best on Liu's data set. The multi-layer perceptron comes second by a rather small margin in all but one metric. The methods FS-MLKNN [25], LNSM-SMI and LNSM-CMI [28] exploit the notion of drug–drug similarity for propagating side effects from a drug to its neighbours. A similar approach is followed by the KG-SIM-PROP and kNN models, which can be considered a simplified version of the ones presented in [28]. The difference between the KG-SIM-PROP and kNN methods and the FS-MLKNN, LNSM-SMI and LNSM-CMI methods is that the last three require large numbers of neighbours to work properly (400 as reported in [25, 28]), while the KG-SIM-PROP and kNN methods can work with as few as 30 neighbours. This makes them more applicable to sparse data sets. As hypothesized by the authors [28], the better results of LNSM-SMI and LNSM-CMI may be attributed to their consideration of neighbourhood as an optimization problem via the linear neighbourhood similarity used. This is confirmed by the observed results and leads to better accuracy in the similarity computation but at the cost of efficiency because of the generation of neighbourhoods. The benefits of treating the similarity as an optimization problem are also shown in the competitive results of multi-layer perceptron, where a logistic sigmoid function was used as kernel. On the other hand, KG-SIM-PROP and kNN use widely used off-the-shelf similarity metrics between feature vectors to determine the neighbourhoods. Methods that do not consider a similarity, namely, decision trees, random forests and linear regression, are among the worst-performing methods. In terms of efficiency, we report that FS-MLKNN was

Table 5. Ranking performance of the models using Liu's data set

Model	Evaluation criterion						
	P@3	P@5	P@10	HITS@1	HITS@3	HITS@5	HITS@10
KG-SIM-PROP [27]	0.9333±0.1333	0.8400±0.2332	0.9200±0.1166	0.8390±0.0164	2.4351±0.0240	3.8691±0.0671	7.0734±0.0746
kNN	0.9333±0.1333	0.9200±0.0980	0.9400±0.0800	0.8450±0.0173	2.4568±0.0316	3.9027±0.0452	7.1744±0.0581
Decision trees	0.4667±0.2667	0.4400±0.2653	0.4800±0.1470	0.4171±0.0176	1.1971±0.0570	1.9651±0.0940	3.8076±0.1941
Random forests	0.9333±0.1333	0.9200±0.0400	0.9200±0.0400	0.8101±0.0088	2.3353±0.0594	3.7451±0.0779	6.9434±0.0982
Multi-layer perceptron	**1.0000±0.0000**	**0.9600±0.0800**	**0.9600±0.0490**	**0.8546±0.0166**	**2.4676±0.0295**	**3.9773±0.0544**	**7.3633±0.1451**
Linear regression	0.3333±0.2981	0.4000±0.1265	0.4400±0.1347	0.5745±0.0469	1.6262±0.0716	2.6394±0.0782	5.1851±0.0823

Note: The evaluation metrics are P@X (precision at 3, 5 and 10), and HITS@X (hits at 1, 3, 5 and 10). (For all metrics, the higher the value of the metric, the better). Bold values represent the best performing methods across a given metric.

Table 6. Predictive power of the models using drugs in Liu's data set with features from Bio2RDF v1 (DrugBank + SIDER)

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
KG-SIM-PROP [27]	0.5011±0.0106	0.4485±0.0115	0.8935±0.0096	0.1058±0.0122	0.1586±0.0177	1095.3082±55.47904
kNN	0.4977±0.0107	0.4210±0.0228	0.8848±0.0062	0.1211±0.0113	0.1658±0.0206	1127.7254±45.6342
Decision trees	0.1964±0.0116	0.1710±0.0138	0.6301±0.0250	0.7220±0.0194	0.5673±0.0144	1377.2001±6.9189
Random forests	0.4317±0.0107	0.3843±0.0143	0.8097±0.0102	0.3037±0.0088	0.2212±0.0139	1314.5006±17.6714
Multi-layer perceptron	**0.5099±0.0159**	**0.4546±0.0169**	**0.9010±0.0061**	**0.0791±0.0022**	**0.1430±0.0160**	**892.8340±20.4758**
Linear regression	0.2847±0.0083	0.2482±0.0137	0.6404±0.0248	0.6726±0.0141	0.3467±0.0238	1383.3808±3.2383

Note: The evaluation metrics are AP, AUC-PR, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.) Bold values represent the best performing methods across a given metric.

the slowest method with >2 weeks running time on a single machine with commodity hardware. This is mainly because of its multiple feature selection steps based on genetic algorithms. From the multi-label ranking methods, the slowest was kNN with 13 h and 18 min, followed by linear regression with 9 h and 26 min. Both multi-layer perceptron and KG-SIM-PROP took ∼2 h and 16 min, while the decision trees were the fastest with only 16 min. We can see that even the slowest among multi-label learning models we have tested is orders of magnitude faster than the best-performing previously published method. This is important information in the context of applicability of different models that is not obvious from previously published work. In cases where quick experimentation with different data sets or model parameters is required, the multi-layer perceptron may well be the best choice, as its results are close to the best performance of existing tools.

In addition to the metrics reported in previous works, we report the ranking performance of the multi-label learning to rank methods in Table 5. Results show that multi-layer perceptron gives the best rankings across all metrics. This may indicate that non-linear methods (such as deep neural nets) are better suited to the ADR prediction problem. Deep learning methods have shown to excel in applications, where there is an abundance of training data, and sources such as Bio2RDF could serve for this purpose. The use of deep learning methods for the prediction of ADRs is still an open problem. Further studies in this area may lead to significant performance improvements as indicated by the preliminary results presented in this review.

Comparison on the Bio2RDF data set

Several authors have found that combining information from different sources can lead to improved performance of computational approaches in bioinformatics (see [48, 49] among others). In 'Materials' section, we introduced the Bio2RDF data set, which is a multi-source knowledge graph. An important aspect of increasing the practicality of ADR prediction we suggest in this review is automation of the feature extraction process. A possible way of doing it is to use heterogeneous knowledge graphs to represent entities such as drugs. This makes experimentation with different feature sets easier than with the existing reviewed works. To show the benefits of combining diverse data sources in terms of performance, we tested the multi-label learning models against two versions of the Bio2RDF data set: (v1) containing DrugBank and SIDER, and (v2) containing DrugBank, SIDER and KEGG. Table 6 shows the performance of six multi-label learning methods (unfortunately, there were no implementations available for LNSM-SMI, LNSM-CMI [28] for comparison at the time of this writing, and FS-MLKNN was discarded because of its intractability on larger feature sets) using the set of 832 drugs and 1385 side effects from Liu's data set, but replacing the feature vectors of drugs with those extracted from the Bio2RDF v1 (or Bio2RDF v2) data set. Originally, Liu's data set contained a set of 2892 manually integrated features coming from six sources. These are replaced by 30 161 and 37 368 features in Bio2RDF v1 and v2, respectively. Both sets are automatically generated using the method described in Supplementary Section A, and represent a drug according to its incoming and outgoing relations with other entities in the knowledge graph.

Table 7. Predictive power of the models using drugs in Liu's data set with features from Bio2RDF v2 (DrugBank + SIDER + KEGG)

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
KG-SIM-PROP [27]	0.5118±0.0101	0.4604±0.0097	0.8954±0.0054	0.1051±0.0109	0.1466±0.0214	1091.9749±51.4537
kNN	0.5083±0.0124	0.4341±0.0277	0.8835±0.0086	0.1281±0.0031	0.1478±0.0027	1155.2053±36.5165
Decision trees	0.2069±0.0176	0.1742±0.0266	0.6258±0.0242	0.7140±0.0233	0.5469±0.0385	1370.7402±7.5913
Random forests	0.4438±0.0162	0.3993±0.0256	0.8153±0.0171	0.2883±0.0225	0.2103±0.0169	1295.7516±20.2287
Multi-layer perceptron	**0.5278±0.0106**	**0.4725±0.0284**	**0.9002±0.0074**	**0.0795±0.0028**	**0.1322±0.0298**	**909.7297±19.7920**
Linear regression	0.2919±0.0109	0.2587±0.0165	0.6441±0.0261	0.6665±0.0166	0.3557±0.0306	1383.3796±3.2407

Note: The evaluation metrics are AP, AUC-PR, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.) Bold values represent the best performing methods across a given metric.

Table 8. Predictive power of the models using a combination of features from both Liu's data set and Bio2RDF v2 data set

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
KG-SIM-PROP [27]	0.5012±0.0079	0.4471±0.0097	0.8882±0.0089	0.1184±0.0139	0.1526±0.0177	1127.3234±51.2769
kNN	0.5020±0.0808	0.4482±0.0101	0.8883±0.0089	0.1184±0.0139	0.1502±0.0208	1127.1279±51.3701
Decision trees	0.2080±0.0190	0.1728±0.0149	0.6306±0.0239	0.6944±0.0215	0.5444±0.0289	1372.1095±9.6089
Random forests	0.4609±0.0174	0.4331±0.0127	0.8357±0.0117	0.2627±0.0134	0.1995±0.0241	1308.7285±24.9798
Multi-layer perceptron	**0.5281±0.0088**	**0.4870±0.0269**	**0.8946±0.0067**	**0.0835±0.0034**	**0.1418±0.0158**	**937.8773±36.9387**
Linear regression	0.3031±0.0108	0.2681±0.0169	0.6578±0.02424	0.6431±0.0147	0.3617±0.0273	1381.7218±4.0156

Note: The evaluation metrics are AP, AUC-PR, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.) Bold values represent the best performing methods across a given metric.

Results show that in both cases (Bio2RDF v1 and v2), the methods perform better with the Bio2RDF features than with Liu's original data set features, confirming our assumption that combination of various feature sources may increase the performance. This can be explained by the fact that Bio2RDF provides a richer representation of drugs and their relationships than the traditional feature sets. This is an important finding, as the Bio2RDF features can be constructed automatically, while the features in the Liu's and Zhang's data sets require non-trivial manual efforts. Furthermore, our results also indicate that having extra information about pathways provides better performance as shown in Table 7, where Bio2RDF v2 is built by adding KEGG data set [33] to Bio2RDF v1. To further explore the influence of possible feature set combinations on the results, we integrated the original Liu's data set [24] features with Bio2RDF v2, leading to 40 260 features in total. Table 8 shows the performance results obtained when combining feature sets from Liu's and Bio2RDF v2 data sets. This yields slightly better results in terms of the AP and AUC-PR metrics.

Comparison on the SIDER 4 data set

To further evaluate the practical applicability of the multi-label learning models, we performed an experiment using SIDER 4 [25]. The intuition behind this experiment is to test the predictive power of the models under a simple train and test set-up. SIDER 4 data set contains 771 drugs used for training, which are also present in Liu's data set, and 309 newly added drugs used for testing. First, we run all methods on the original SIDER 4 data set features and labels, and compare them against the results provided by Zhang *et al.* [28]. Table 9 shows the results of the different methods over the SIDER 4 data set. The state-of-the-art method LNSM-SMI gives the best AP and AUC-PR, while LNSM-CMI produces the best Cov-error. However, multi-layer perceptron is the best-performing model in the AUC-ROC, R-loss and one-error metrics. These results suggest better relative suitability of some multi-label learning methods for applications, where a ranking function is preferred over classification. Examples of such applications are use cases, where experts can only review a few prediction candidates and need the relevant ones to appear at the top of the list. Such use cases are indeed realistic, as there are often hundreds of predictions for every single drug. The results of multi-layer perceptron show some improvements when using features coming from the Bio2RDF v2 data set (cf. Table 10).

Comparison on the SIDER4 and Aeolus data sets

We further evaluate the models considering both the SIDER 4 and Aeolus data sets [35]. Aeolus data set provides us with relations between drugs and ADRs that were not previously known during the training or testing steps. The reason for the experiments using the SIDER 4 and Aeolus data sets is the evolving nature of the knowledge about drugs—generally, new ADRs can always be discovered for a drug, either by new studies or via pharmacovigilance in the post-market stage. The classic approach for validating ADR predictions follows the closed-world assumption (i.e. missing predictions are false), but the actual problem follows the open-world assumption (i.e. missing predictions may be just unknown). Therefore, it is always possible that predictions that are currently deemed false positives can be considered true positives if more knowledge becomes available in the future. We hope to reflect this phenomenon by using the complementary Aeolus data that is frequently updated and contains information based on manually validated reports. For these reasons, we believe it will be beneficial to use this data set for complementary validations also in future studies in this domain.

Table 9. Predictive power of the models using SIDER 4 data set

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
Liu's method [24]	0.1816	0.1766	0.8772	0.1150	0.9870	1587.5663
FS-MLKNN [28]	0.3649	0.3109	0.8722	0.1038	0.1851	1535.9223
LNSM-SMI [28]	**0.3906**	**0.3465**	0.8786	0.0969	0.2013	1488.2977
LNSM-CMI [28]	0.3804	0.3332	0.8852	0.0952	0.1916	**1452.7184**
KG-SIM-PROP [27]	0.3375	0.2855	0.8892	0.1398	0.2233	4808.3689
kNN	0.3430	0.2898	0.8905	0.1392	0.2168	4086.0777
Random forests	0.3004	0.2599	0.8235	0.3318	0.2848	5362.6117
Multi-layer perceptron	0.3546	0.2899	**0.8943**	**0.0922**	**0.1309**	4054.0356

Note: The values for the first four models were taken from [28]. The evaluation metrics are AP, AUC-PR curve, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.) Bold values represent the best performing methods across a given metric.

Table 10. Predictive power of the models using drugs in SIDER 4 data set and Bio2RDF v2 data set features

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
KG-SIM-PROP [27]	0.3438	0.2876	0.8764	0.17460	0.2427	4969.0647
kNN	0.3416	0.2835	0.8728	0.1777	0.2395	5002.6084
Random forests	0.2384	0.2061	0.7651	0.4567	0.4304	5440.0712
Multi-layer perceptron	0.3529	0.2857	0.9043	0.0852	0.1909	3896.3625

Note: The evaluation metrics are AP, AUC-PR, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.)

Table 11. Predictive power of the models using SIDER 4 data set, and updating the ADRs with Aeolus data set

Model	Evaluation criterion					
	AP ↑	AUC-PR ↑	AUC-ROC ↑	R-loss ↓	One-error ↓	Cov-error ↓
KG-SIM-PROP [27]	0.3272	0.2791	0.8796	0.1619	0.2233	5040.06149
kNN	0.3324	0.2834	0.8808	0.1613	0.2168	5038.6570
Random forests	0.2883	0.2447	0.8059	0.3717	0.3366	5478.8479
Multi-layer perceptron	0.3437	0.2836	0.8858	0.1050	0.1909	4339.7540

Note: The evaluation metrics are AP, AUC-PR, AUC-ROC, R-loss, one-error and Cov-error. ('↑' indicates that the higher the metric value, the better, and '↓' indicates that the lower the metric value, the better.)

To test this point, we updated the SIDER 4 matrix **Y** of ADRs of the test set using a version of Aeolus data set generated after the release of the SIDER 4 data set. We found 142 drugs in the intersection of the SIDER 4 test set and Aeolus. Whenever a new drug–ADR relationship is reported in the Aeolus data set for any of the 309 drugs in the test set, this is reflected by modifying the SIDER 4 data set. Aeolus introduces 615 new ADR relations in total with an average of 4.3 per drug. For example, Aeolus provides two new ADRs for triclosan (DB08604), an aromatic ether widely used as a preservative and antimicrobial agent in personal care products: odynophagia and paraesthesia oral. While these changes because of the Aeolus data set are not crucial for drugs with many previously known ADRs (for instance, nilotinib (DB04868) has 333 ADRs in SIDER 4, and Aeolus only adds 3 new ADRs), they can have high impact on drugs with few known ADRs (such as triclosan or mepyramine both with only one ADR). In total, Aeolus provides at least one new ADR for 46% of drugs in the SIDER 4 test set. Interestingly, most of the new ADRs added by Aeolus data set are related to the digestive system (e.g. intestinal obstruction, gastric ulcer, etc.), which we believe is because of the disproportionate FAERS reporting [8, 10] frequency for this type of events.

We ran the models once more and evaluated them against the new gold standard with the updates provided by the Aeolus data set. Table 11 shows the results of the updated data set using the Aeolus data for the four best-performing multi-label models, and when compared against values in Table 9 results are marginally lower across all metrics. For instance, the AP of multi-layer perceptron drops by 0.92% and AUC-ROC by 1.85%. This observation is not consistent with our assumption that new knowledge about relations between drugs and ADRs can increase the true-positive rate by confirming some of the previous false positives as being true. We believe that this could be because of two reasons. (A) The added ADRs are underrepresented across drugs. We observed this in SIDER 4, where 37.5% (2093 of 5579) of ADRs are present at most once in either

the training or test set. This makes those ADRs hard to predict. (B) There is a 'weak' relation between the drugs and the introduced ADRs. This weak relation comes from the original split in training and test set provided in SIDER 4 data set; we found out that 50.15% (2798 of 5579) ADRs are only present in the training set and not in the test set, compared with a 7% (392 of 5579) of ADRs that are only present in the test set.

Advantages of using Aeolus data set are illustrated, for example, by the drug eribulin (DB08871) that contains 123 ADRs in SIDER 4, most of which have been discovered in the post-marketing stage. Aeolus introduced seven new ADRs for eribulin, where one of them, namely, pharyngitis (C0031350), was ranked number 36 among all 5579 ADRs, which is a high ranking considering 123 ADRs. This means the models are able to perform well for reactions that are true based on the recent data in Aeolus, but not present among positives in the primary validation data like SIDER (and thus they could only be interpreted as false positives during the primary evaluation). Such encouraging results were observed on several of the analysed drugs for which predictions previously considered as false positives were indeed shown to be true by Aeolus.

All analysed methods consider a static view over the data and do not consider the changes in data, e.g. new ADRs discovered in a post-marketing stage. Therefore, a future research direction could study the effects of learning under evolving data sets (i.e. new drug–ADR relations), which is known as incremental learning (see [50, 51, 52] among others).

Comments on the behaviour of the models

To illustrate the flexibility and robustness of the approach we suggest to complement the existing predictive models, we enriched the Liu's data set using Bio2RDF data set features, which in general are numerous. Intuitively, by having more features for a drug, we can achieve a better representation of it, which should lead to better performance results. However, we observed mixed small positive and negative changes in the results shown in Table 8 when compared with the performance previously reported in Tables 6 and 7. This can be attributed to the famous curse of dimensionality, where the performance degrades as a function of dimensionality. This issue may have large impact on models like multi-layer perceptron, where the large number of inputs hampers the training performance if the first hidden layer is too small. This is the case of our experiments, as we limit the size of the first hidden layer for the multi-layer perceptron. However, it is possible to cope with the curse of dimensionality, using methods such as embeddings into low-rank feature spaces. Embedding models aim to find a dimensionality reduction, generating latent representations of the data that preserve structural details as much as possible [53]. This is something that represents a new research direction, by considering learning of drug representations for tasks such as comparison. We believe this could substantially improve the performance of some of the models here reviewed.

We also observed that when merging Liu's data set with Bio2RDF, some features can be considered as duplicated features. Certain models deal with this situation better, and others would apparently require a filtering of duplicated features. During our experiments, we did not filter features, and assumed that deduplication is performed by the models.

Regarding scalability, despite the substantial increase of the feature space (up to almost 13-fold), we only noticed up to double execution times of the multi-label learning methods. All running times are still far better than the time required by the previously existing methods, which is another argument for higher practical applicability of the suggested approach.

Conclusion

We have shown that using knowledge graphs for automated feature extraction and casting the problem of ADR prediction as multi-label ranking learning can be used for building models that are comparable with or better than existing related methods. Moreover, the off-the-shelf models are orders of magnitude faster than existing related ADR prediction systems. We argue that because of the demonstrated speed-up and automation of most of the steps in building the prediction pipelines, this review provides a broad range of possibilities for biomedical experts to build their own ADR prediction systems more easily than before. This is supported by extensive documentation of all necessary steps provided in the article (cf. Supplemental Material).

The applicability of some of the reviewed models is further supported by good results in ranking metrics. This can be useful in many practical scenarios, where experts cannot possibly explore all computed predictions, but require ranked results and highly relevant candidates appearing at the top of the list. Last but not least, the review presents results of the off-the-shelf machine learning modules in a way that can be used as a well-documented baseline for future experiments in this area.

In our future work, we want to investigate the influence of embeddings (i.e. latent feature models and feature extractors) on the performance of multi-label learning models for the ADR prediction. We also want to analyse the influence of various hyper-parameters on the prediction results more thoroughly. This will bring more insight into the most promising directions for further improvements of the performance of ADR prediction models. Another area we want to target is development of more stratified and comprehensive benchmark data sets that could increase the interpretability of ADR prediction results in future studies. Last but not least, we would like to perform not only quantitative validation but also qualitative trials with actual sample users. This will let us assess the real-world usability of the reviewed approaches and gain valuable feedback for further developments in this field.

Key Points

- Knowledge graphs allow for easy, automated integration of multiple diverse data sets to extract features for ADR prediction.
- Approaching the ADR prediction as a multi-label learning problem facilitates easy experimentation with a diverse range of off-the-shelf algorithms. It also produces results that can be used as a well-documented baseline for future, more sophisticated experiments.
- Applying these two principles (i.e. knowledge graphs and multi-label learning) leads to results that are comparable with or better than existing related approaches, while the training is orders of magnitude faster on the same data. Also, the resulting models provide ranked predictions by default, which further contributes to their practical applicability.
- Interested stakeholders can straightforwardly use the review for building their own ADR prediction pipelines and fine-tuning them based on their specific requirements (such as increasing particular classification or ranking performances).

Supplementary Data

Supplementary data are available online at http://bib.oxfordjournals.org/.

Acknowledgements

The authors kindly acknowledge Pasquale Minervini for contributing to the Python implementation in an early stage of this work. The authors thank the three anonymous reviewers for their valuable comments and suggestions that helped us improve the manuscript.

Funding

The TOMOE project funded by Fujitsu Laboratories Ltd., Japan and Insight Centre for Data Analytics at National University of Ireland Galway (supported by the Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289).

Availability of data and material

We make available all design matrices with drug features and labels (ADRs) as MATLAB files. All data files are available for download at http://purl.com/bib-adr-prediction. Further details on the feature extraction step and manipulation of data sets are provided in the Supplemental Material.

References

1. Bowes J, Brown AJ, Hamon J, *et al*. Reducing safety-related drug attrition: the use of *in vitro* pharmacological profiling. *Nat Rev Drug Discov* 2012;**11**:909–22.
2. Sultana J, Cutroneo P, Trifiró G, *et al*. Clinical and economic burden of adverse drug reactions. *J Pharmacol Pharmacother* 2013;**4**:73–7.
3. Bouvy JC, De Bruin ML, Koopmanschap MA. Epidemiology of adverse drug reactions in Europe: a review of recent observational studies. *Drug Saf* 2015;**38**:437–53.
4. Edwards IR, Aronson JK. Adverse drug reactions: definitions, diagnosis, and management. *Lancet* 2000;**356**:1255–9.
5. Giacomini KM, Krauss RM, Roden DM, *et al*. When good drugs go bad. *Nature* 2007;**446**:975–7.
6. Johnson J, Booman L. Drug-related morbidity and mortality. *J Manag Care Pharm* 1996;**2**:39–47.
7. Kola I, Landis J. Can the pharmaceutical industry reduce attrition rates? *Nat Rev Drug Discov* 2004;**3**:711–16.
8. Szarfman A, Machado SG, O'Neill RT. Use of screening algorithms and computer systems to efficiently signal higher-than-expected combinations of drugs and events in the US FDA's spontaneous reports database. *Drug Saf* 2002;**25**:381–92.
9. Mammadov MA, Rubinov AM, Yearwood J. The study of drug-reaction relationships using global optimization techniques. *Optim Methods Softw* 2007 Feb;**22**:99–126.
10. Harpaz R, Vilar S, DuMouchel W, *et al*. Combing signals from spontaneous reports and electronic health records for detection of adverse drug reactions. *J Am Med Inform Assoc* 2013;**20**:413–19.
11. Karimi S, Wang C, Metke-Jimenez A, *et al*. Text and data mining techniques in adverse drug reaction detection. *ACM Comput Surv* 2015;**47**:56:1–56.
12. Ginsberg J, Mohebbi MH, Patel RS, *et al*. Detecting influenza epidemics using search engine query data. *Nature* 2009;**457**:1012–14.
13. White RW, Wang S, Pant A, *et al*. Early identification of adverse drug reactions from search log data. *J Biomed Inform* 2016;**59**:42–8.
14. Tan Y, Hu Y, Liu X, *et al*. Improving drug safety: from adverse drug reaction knowledge discovery to clinical implementation. *Methods* 2016;**110**:14–25.
15. Campillos M, Kuhn M, Gavin AC, *et al*. Drug target identification using side-effect similarity. *Science* 2008;**321**:263–6.
16. Vilar S, Hripcsak G. The role of drug profiles as similarity metrics: applications to repurposing, adverse effects detection and drug–drug interactions. *Brief Bioinform* 2017;**18**:670–81. doi: 10.1093/bib/bbw048.
17. Atias N, Sharan R. An algorithmic framework for predicting side effects of drugs. *J Comput Biol* 2011;**18**:207–18.
18. Pauwels E, Stoven V, Yamanishi Y. Predicting drug side-effect profiles: a chemical fragment-based approach. *BMC Bioinformatics* 2011;**12**:169.
19. Bresso E, Grisoni R, Marchetti G, *et al*. Integrative relational machine-learning for understanding drug side-effect profiles. *BMC Bioinformatics* 2013;**14**(1):207.
20. Jahid MJ, Ruan J. An ensemble approach for drug side effect prediction. In: *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, 2013*. IEEE, 2013, 440–5.
21. Mizutani S, Pauwels E, Stoven V, *et al*. Relating drug–protein interaction network with drug side effects. *Bioinformatics* 2012;**28**:i522–8.
22. Yamanishi Y, Pauwels E, Kotera M. Drug side-effect prediction based on the integration of chemical and biological spaces. *J Chem Inf Model* 2012;**52**:3284–92.
23. Huang LC, Wu X, Chen JY. Predicting adverse drug reaction profiles by integrating protein interaction networks with drug structures. *Proteomics* 2013;**13**:313–24.
24. Liu M, Wu Y, Chen Y, *et al*. Large-scale prediction of adverse drug reactions using chemical, biological, and phenotypic properties of drugs. *J Am Med Inform Assoc* 2012;**19**:e28–35.
25. Zhang W, Liu F, Luo L, *et al*. Predicting drug side effects by multi-label learning and ensemble learning. *BMC Bioinformatics* 2015;**16**(1):365.
26. Zhang W, Zou H, Luo L, *et al*. Predicting potential side effects of drugs by recommender methods and ensemble learning. *Neurocomputing* 2016;**173**:979–87.
27. Muñoz E, Novacek V, Vandenbussche PY. Using drug similarities for discovery of possible adverse reactions. In: *AMIA 2016, American Medical Informatics Association Annual Symposium*. American Medical Informatics Association, 2016, 924–33. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5333276/
28. Zhang W, Chen Y, Tu S, *et al*. Drug side effect prediction through linear neighborhoods and multiple data source integration. In: *Proceedings of the* IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2016. IEEE, 2016, 427–434. https://doi.org/10.1109/BIBM.2016.7822555
29. Rahmani H, Weiss G, Méndez-Lucio O, *et al*. ARWAR: a network approach for predicting adverse drug reactions. *Comput Biol Med* 2016;**68**:101–8.
30. Kuhn M, Letunic I, Jensen LJ, *et al*. The SIDER database of drugs and side effects. *Nucleic Acids Res* 2016;**44**:D1075.
31. Kim S, Thiessen PA, Bolton EE, *et al*. PubChem substance and compound databases. *Nucleic Acids Res* 2016;**44**:D1202.
32. Law V, Knox C, Djoumbou Y, *et al*. DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Res* 2014;**42**:D1091–7.

33. Kanehisa M, Furumichi M, Tanabe M, *et al*. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* 2017;**45**:D353–61.
34. Belleau F, Nolin MA, Tourigny N, *et al*. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J Biomed Inform* 2008;**41**:706–16.
35. Banda JM, Evans L, Vanguri RS, *et al*. A curated and standardized adverse drug event resource to accelerate drug safety research. *Sci Data* 2016;**3**:160026.
36. Dumontier M, Callahan A, Cruz-Toledo J, *et al*. Bio2RDF release 3: a larger, more connected network of linked data for the life sciences. In: *Proceedings of the 2014 International Conference on Posters & Demonstrations Track*, Vol. 1272. CEUR-WS.org, 2014, 401–404.
37. Ritz A, Tegge AN, Kim H, *et al*. Signaling hypergraphs. *Trends Biotechnol* 2014;**32**:356–62.
38. Bisgin H, Liu Z, Fang H, *et al*. Mining FDA drug labels using an unsuperv learning technique—topic modeling. *BMC Bioinformatics* 2011;**12**:S11.
39. Tsoumakas G, Katakis I. Multi-label classification: an overview. *Int J Data Warehousing Min* 2006;**3**:1–13. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.9401
40. Zhang ML, Zhou ZH. A review on multi-label learning algorithms. *IEEE Trans Knowl Data Eng* 2014;**26**:1819–37.
41. Zhang ML, Zhou ZH. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans Knowl Data Eng* 2006;**18**:1338–51.
42. Zha ZJ, Mei T, Wang J, *et al*. Graph-based semi-supervised learning with multiple labels. *J Vis Commun Image Represent* 2009;**20**:97–103.
43. Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**:2825–30.
44. Choi S, Cha S, Tappert CC. A survey of binary similarity and distance measures. *J Syst Cybern Inf* 2010;**8**:43–8.
45. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In: *Proceedings of the 23rd International Conference on Machine Learning. ICML '06*. New York, NY: ACM, 2006, 233–40.
46. Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. In: O Maimon, L Rokach (eds). *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2010, 667–85.
47. Manning CD, Raghavan P, Schütze H. Chapter 8: Evaluation in information retrieval. In *Introduction To Information Retrieval*. Cambridge: Cambridge University Press, 2008, 151–75.
48. Polikar R. Ensemble based systems in decision making. *IEEE Circuits Syst Mag* 2006;**6**:21–45.
49. Yang R, Zhang C, Gao R, *et al*. An ensemble method with hybrid features to identify extracellular matrix proteins. *PLoS One* 2015;**10**(2):e0117804.
50. Schlimmer JC, Granger RH. Incremental learning from noisy data. *Mach Learn* 1986;**1**:317–54.
51. Rüping S. Incremental learning with support vector machines. In: Proceedings 2001 IEEE International Conference on Data Mining. IEEE Computer Society, 2001, 641–2. https://doi.org/10.1109/ICDM.2001.989589
52. Raway T, Schaffer DJ, Kurtz KJ, *et al*. Evolving data sets to highlight the performance differences between machine learning classifiers. In: Proceedings of the Annual Conference Companion on Genetic and Evolutionary Computation. New York, NY, USA: ACM, 2012, 657–8. https://doi.org/10.1145/2330784.2330907
53. Dai G, Yeung DY. Tensor embedding methods. In: *Proceedings of the 21st National Conference on Artificial Intelligence*, Vol. 1. AAAI Press, 2006, 330–335. https://dl.acm.org/citation.cfm?id=1597592
54. Barabási AL. *Network Science*. Cambridge University Press, 2016. http://dx.doi.org/10.1063/PT.3.3526
55. Liu TY. *Learning to Rank for Information Retrieval*. Springer Berlin Heidelberg: Springer Science & Business Media, 2011. http://dx.doi.org/10.1007/978-3-642-14267-3

Chapter 10

Integrated Biomedical Knowledge Graph

BioKG: A Knowledge Graph for Relational Learning On Biological Data

Brian Walsh
Data Science Institute, NUI Galway
Insight Centre for Data analytics
Galway, Ireland
brian.walsh@insight-centre.org

Sameh K. Mohamed
Data Science Institute, NUI Galway
Insight Centre for Data analytics
Galway, Ireland
sameh.kamal@insight-centre.org

Vít Nováček
Data Science Institute, NUI Galway
Insight Centre for Data analytics
Galway, Ireland
vit.novacek@insight-centre.org

ABSTRACT

Knowledge graphs became a popular means for modelling complex biological systems where they model the interactions between biological entities and their effects on the biological system. They also provide support for relational learning models which are known to provide highly scalable and accurate predictions of associations between biological entities. Despite the success of the combination of biological knowledge graph and relation learning models in biological predictive tasks, there is a lack of unified biological knowledge graph resources. This forced all current efforts and studies for applying a relational learning model on biological data to compile and build biological knowledge graphs from open biological databases. This process is often performed inconsistently across such efforts, especially in terms of choosing the original resources, aligning identifiers of the different databases and assessing the quality of included data. To make relational learning on biomedical data more standardised and reproducible, we propose a new biological knowledge graph which provides a compilation of curated relational data from open biological databases in a unified format with common, interlinked identifiers. We also provide a new module for mapping identifiers and labels from different databases which can be used to align our knowledge graph with biological data from other heterogeneous sources. Finally, to illustrate practical relevance of our work, we provide a set of benchmarks based on the presented data that can be used to train and assess the relational learning models in various tasks related to pathway and drug discovery.

CCS CONCEPTS

• **Applied computing** → **Biological networks**; **Bioinformatics**; • **Information systems** → **Extraction, transformation and loading**.

ACM Reference Format:
Brian Walsh, Sameh K. Mohamed, and Vít Nováček. 2020. BioKG: A Knowledge Graph for Relational Learning On Biological Data. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412776

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412776

1 INTRODUCTION

Knowledge graphs (KGs) are a popular means for modelling relational data in many systems and applications. They have currently become the backbone of many semantic web search engines and question answering systems in both academic and industrial settings [27]. This encouraged the development of many public knowledge graphs which model information from different domains such as general human knowledge [18], lexical information [20] and other domains. These knowledge graphs provide support for predictive models in different tasks and facilitate information retrieval on the original linked data.

In recent years, knowledge graphs have also become a favourable choice for modelling complex biological systems where they were used in different predictive tasks such as predicting drug protein targets [26], predicting polypharmacy side-effects [40] and the prediction of cellular functions of proteins at the tissue level [22]. In each of these tasks, KGs were used to model biological networks, and then relational learning models were used to provide new predictions. Despite the effectiveness of such approaches [24], there is a lack of open biological knowledge graphs to support them. Furthermore, current approaches rely on building customized knowledge graph by parsing and transforming open biological databases [24, 26, 29].

The effectiveness of knowledge graphs and the popularity of the RDF framework form modelling linked data have encouraged many open biological database to provide their contents in RDF format. For example, the UNIPROT [6, 7], Reactome [8] and Gene Ontology [5] databases provide an RDF version of their content which preserves both the interlinks and metadata of their contained biological entities. However, these RDF graphs only focus on a limited set of biological entity types covered by the corresponding original database. Moreover, they do not share any common entity coding system, which makes it hard to use them in concert. There is also a large body of biological data that has no RDF counterpart at all. This encouraged efforts such as the Bio2RDF project [3] to build and provide a network of linked biological data by transforming open biological databases to RDF graphs.

The Bio2RDF project consists of a set of web parsers for open biological data which consume, process and convert these database to RDF graphs. Despite the high coverage of its generated RDF graphs, they are not commonly used in the different predictive biological task by relational learning models [24]. One of the main reasons is the large volume of metadata information stored in these graphs which often decreases the predictive accuracy of relational models.

This is due to the models' tendency to over-represent the clearly-interpretable and uniform metadata links and under-represent the more subtle actual biological relationships.

The current studies of the applications of relational leaning models in biological settings commonly involve building customized biological graphs from open biological databases [24, 26, 29]. This process involves repeated procedures such as parsing the different database sources into intermediate formats then merging these format into knowledge graphs. It also involves mapping entity identifiers to a unified ID system as biological databases commonly employ different identifier systems. Such steps are frequently associated with many rather arbitrary decisions that complicate reproducibility and meaningful comparisons between the corresponding models. To address this problem, we provide a new open biological knowledge graph, BioKG, and tools for its transparent creation, updates and extensions. Contrary to existing resources like Bio2RDF, BioKG combines information from different open biological databases in a simple graph format which focuses on biological relationships while preserving basic important ontological information, and thus it allows for straightforward development and comparative evaluation of relation learning models.

We discuss related works in Section 2 and we discuss our main contributions in Sections 3,4 and 5 as follows:

(1) In Section 3, we propose a biological knowledge graph (BioKG) compiled from open source databases to support relational learning models in predictive tasks on biological data.
(2) In Section 4, we propose a software module (BioDBLinker) which provides name-id lookup and mapping of different id systems for biological entities.
(3) In Section 5, we propose a set of five benchmarking datasets for assessing the predictive accuracy of relational learning models in different tasks related to drug–protein, drug–drug and protein–protein interactions.

In Section 6, we discuss potential applications and possible issues related to the development and use of BioKG knowledge graph, and our intentions for future extensions of this work. Finally, in Section 7 we discuss our conclusions.

2 RELATED WORKS

In this section, we discuss studies and resources related to our newly proposed knowledge graphs.

2.1 Open Biological Databases

Open biological databases support research in both clinical and computational biology. They contain different types of structured and unstructured data related to different biological phenomenons. In this work, we focus on databases that provide biological data which is related to molecular and pharmacological activities, *e.g.* protein interactions, drug protein targets, etc.

In Table 1 we provide detailed statistics on a selected set of popular biological databases which are commonly used to train relational learning models in bioinformatics settings. We provide a comparison between these databases in terms of their data formats, specialities and the covered biological entities. In terms of the data format, the table shows that almost all the databases contain structured data while a subset of these databases contains both structured and unstructured data. These unstructured data are usually comments and annotations of describing pieces of the structured data as in the protein–protein interactions related comments in the UniProt database [6].

Table 1: A comparison between popular biological databases in terms of the coverage of different types of biological entities. The abbreviation S represent structured data, U represents unstructured data, DR represents drugs, PR represents proteins, GO represents gene ontology,PA represents pathways, GD represents genetic disorders, CL represents cell-lines and *CH* denotes chemicals.

	Properties		Entity coverage							
Database Name	**Format**	**Speciality**	**Proteins**	**Drugs**	**Indications**	**Diseases**	**Gene Ontology**	**Expressions**	**Pathways**	**In BioKG ?**
UniProt [7]	S/U	PR	✓	✓	✗	✓	✓	✓	✓	✓
REACTOME [8]	S	PA	✓	✗	✗	✗	✓	✗	✓	✓
KEGG [14]	S	PA	✓	✓	✗	✓	✗	✗	✓	✓
DrugBank [15]	S/U	DR	✓	✓	✗	✗	✗	✗	✓	✓
GO [5]	S	GO	✓	✗	✗	✗	✓	✗	✓	✓
CTD [19]	S/U	CH	✓	✓	✗	✗	✓	✗	✓	✓
SIDER [16]	S	DR	✗	✓	✓	✗	✗	✗	✗	✓
HPA [36]	S/U	PR	✓	✗	✗	✗	✓	✓	✗	✓
STRING [33]	S	PR	✓	✗	✗	✗	✗	✗	✗	✗
BIOGRID [32]	S	PR	✓	✗	✗	✗	✗	✗	✗	✗
IntAct [30]	S	PR	✓	✗	✗	✗	✗	✗	✗	✓
InterPro [21]	S	PR	✓	✗	✗	✗	✗	✗	✗	✓
PharmaGKB [10]	S	DR	✓	✓	✗	✗	✗	✗	✗	✗
TTD [17]	S	DR	✓	✓	✗	✗	✗	✗	✗	✗
Supertarget [9]	S	DR	✓	✓	✗	✗	✗	✗	✗	✗
Cellosaurus [2]	S/U	CL	✗	✗	✗	✗	✗	✓	✗	✓
MESH [1]	S/U	CL	✗	✗	✗	✓	✗	✗	✗	✓
OMIM [1]	S	GD	✓	✗	✗	✗	✗	✗	✗	✓

While the majority of the reviewed databases specialize in data focused on proteins, the UniProt database is the most popular source for protein related data as it has the highest coverage of expert-curated protein annotations [7]. The UniProt database consists of two parts SwissProt (expert-curated) and TrEMBL (lower confidence annotation). It also use a protein id naming system known as "UniProt Accessions". Other databases such as KEGG and CTD databases use "Gene Id Numbers" as ids for proteins where they define unique proteins based on their source genes. The HPA [36] and STRING [37] databases use yet another a different gene-based id system for proteins. Although all these databases have intersection between their reported protein annotations, they do not have a one-to-one mapping between their ids, therefore merging their annotations can be complicated. Similarly, databases that provide data on drugs such as the DrugBank [15], SIDER [16], CTD [19] and KEGG [14] databases also use different id systems for drugs which often does not have a one-to-one mapping for some of their common entities.

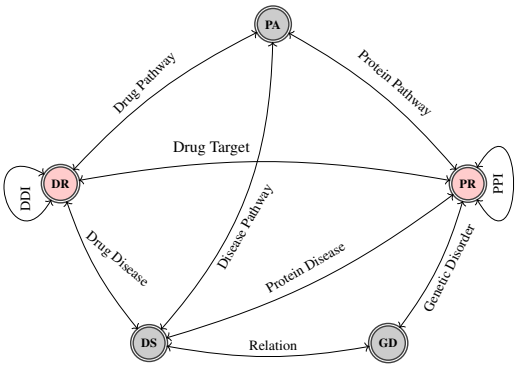


Figure 1: The schema of BioKG main biological entities and their connections. Abbreviations in this illustration are the same as in Table 1.

While resources like Bio2RDF and RDF versions of open biological databases aim to resolve these problems, their main objective was to integrate the biological databases with semantic web technologies. This led to the development of biological RDF graphs that have complex ontological information. These graphs, however, still have issues when used for training relational learning models due to their use of different id systems, variable quality and dense ontological data that is largely irrelevant to training predictive models, which the presented work attempts to remedy.

2.2 Relational Learning in Bioinformatics

In recent years, relational learning models (RLMs) became a popular method in many bioinformatics predictive tasks where they outperform other state-of-the-art approaches in various tasks [24]. They use knowledge graphs to model complex biological systems and they then learn feature representations of entities and relationships to provide accurate and scalable predictions. For example, Zitnik et. al. [40] have modelled drug–drug interactions and their associated side-effects as a knowledge graph and they applied a graph convolutional network model to predict the polypharmacy side-effects of drugs. This work has also shown that such an approach outperforms previous state-of-the-art methods in terms of the predictive accuracy. Furthermore, other studies have shown that modelling biological data with knowledge graphs and using knowledge graph embedding models *e.g.* TransE [4], ComplEx [35], TriVec [25], etc, to predict biological relationships is effective in tasks such as drug target interaction prediction [26] and tissue-specific protein function prediction [22].

In Fig. 1, we provide an basic graph schema of the mainly investigated relationships between biological entities and their related information at the molecular and pharmacological level. These relationships include the previously mentioned drug protein targets and drug side-effects along with other relationships such as disease associated genes, protein associated pathway, etc. In the context of predicting each of these relationships, relational learning models usually build a knowledge graph centred around the two end of the relationships where it usually include other relationships in graph schema. For example, in the prediction of relationships between drugs and proteins, RLMs are usually training on a knowledge graph that has information about drugs such as the ATC class, chemical structure groups, etc [26]. It also includes information about proteins such as protein–protein interactions, protein related pathway, gene ontology annotations, etc. All this information has to be fused in one knowledge graph centred around drug–protein relationships to enable RLMs to efficiently model and predict new drug–protein interactions. However, there is no existing, publicly available data set that would enable this with sufficient coverage, which is another gap the presented work covers.

3 BIOKG KNOWLEDGE GRAPH

In this section, we discuss the contents of BioKG knowledge graph and the details of the pipeline to build these contents. We also provide statistics of its covered entities and relations.

3.1 Processing the Original Data Sources

The BioKG knowledge graph is built through a two-phase procedure as shown in Fig. 2. This procedure includes parsing open biological database to intermediate structured formats, then integrating these formats to obtain the BioKG contents. In the following, we discuss this procedure where we describe materials and techniques used in each phase.

3.1.1 Parsing Sources. The BioKG consists of a set open biological databases (identified in Table 1) which contain different types of biological data. The criteria used for choosing these databases depend on three factors: entity coverage, data quality, and integration with other databases. In terms of coverage, the UniProt and KEGG databases are the most popular sources for protein data as they have the highest coverage of proteins/genes especially in humans. This can be shown by the wide adoption of UniProt ids as baseline references for proteins in multiple open biological databases such as Reactome, Phosphosite, etc. We also selected a wide range database to cover the different types of biological entities such as proteins, drugs, pathways, expressions and other entities as illustrated in Fig. 1. In terms of quality, we focus on databases that have expert-curated data and we exclude data generated by inference technique to ensure high quality data. We, therefore, only include the SwissProt part of UniProt which contain only reviewed protein entries and we exclude the inferred data parts of databases. We also selected databases that have better integration to ensure the connectivity of the different parts of BioKG knowledge graph.

For each of the selected databases, we parse the database contents into a structured tabular format. This format allows for more dynamic representations of the included data which is often modelled in higher dimensionality and/or more complex formats than knowledge graph triplets. For example, the protein tissue expression data parsed from the human protein atlas (HPA) is associated with different expression levels. This data is stored in an intermediate format in the form (<protein>, expressed_in, <tissue>, <expression_level>) where protein, tissue and tissue levels are variable depending on the different entries. This format is incompatible with knowledge graph triplets, so, in the final phase it is converted to triplets by excluding the expression level column data. We have developed automated parsers for each of the included databases which consume

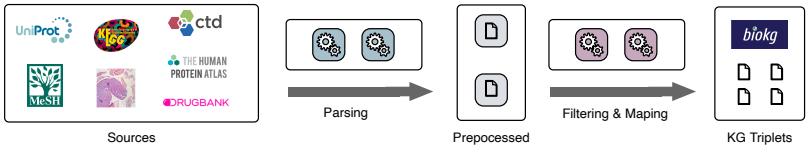


Figure 2: An Illustration of the processing pipeline to build the BioKG data.

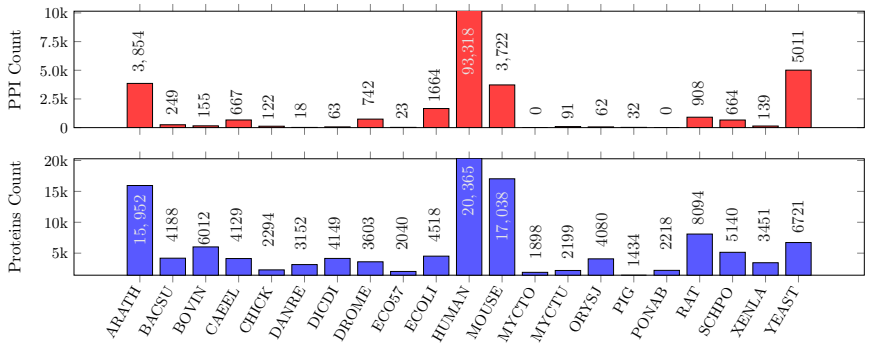


Figure 3: Statistics of proteins in the BioKG and their protein-protein interactions (PPIs) categorized by their species. The PPIs for each species are only considered where both proteins are from the same species.

and parse the contents of the database and output intermediate data files.

3.1.2 Compiling BioKG Contents. The BioKG knowledge graph contents is compiled from the intermediate formats generated by the biological databases' parsers. The compilation process mainly involves to three procedures: id , filtering and building triplets. The mapping of ids is the process of converting ids of entities of the same type to the same id system. We execute this process using the BioDBLinker module that discuss in details in Section 4. For example, all drugs in the BioKG use the DrugBank ids while all proteins use the UniProt ids. This ensures the connectivity of nodes coming from heterogeneous source in the BioKG knowledge graph.

Data filtering process in the building of BioKG contents is aimed to satisfy to objectives: high quality data and focus on drug discovery biological applications. The high quality data is obtained by filtering intermediate data formats to only include the expert curated parts of the included databases. The data is also filtered to only include biological entities and phenomenons related to drugs and their related protein targets' activities such as illustrated in Fig. 1.

3.2 Structure of the Knowledge Graph

The BioKG knowledge graph compiles biological data from different sources in a graph format with focus on data on proteins and chemical drugs. The contents of BioKG knowledge graph can be categorized into three categories: links, properties and metadata. Links represent the connections between the different biological entities, while properties represent the annotations associated to entities. Each biological entity type has its own set of links and properties that describe its activities in biological systems. Fig. 1 illustrates the main biological entities included in the BioKG and the relationships between them.

In the following, we discuss the contents included under each of the three categories (links, properties and metadata) in the BioKG knowledge graph.

3.2.1 BioKG Links. The links part of the BioKG data is the core part of BioKG which models the relationships between the biological entities as illustrated in Fig. 1. The number of instances of each of the relationships in BioKG is illustrated in Fig 4 and they are described as follows:

- **protein–protein interactions (PPIs)**. BioKG contains 113451 PPIs of 21 selected species as illustrated in Fig 3. These interactions are extracted from the respective protein entries from SwissProt and IntAct databases.
- **drug–protein interactions (DPIs)**. BioKG contains different types of DPIs such as drug–protein targets, carriers, enzymes and transporters. The DPIs in BioKG are focused on human proteins and they are extracted from the DrugBank and KEGG databases exclusively, and they are considered in two forms: unified relation and separate relation for each type (exclusively from DrugBank). The DPI relations (27781 instances) in the BioKG links are the union of all the separate instances of the drug-carries, drug-transporters, drug-targets, drug-enzymes relationships combined with other drug-protein interactions extracted from the KEGG database.

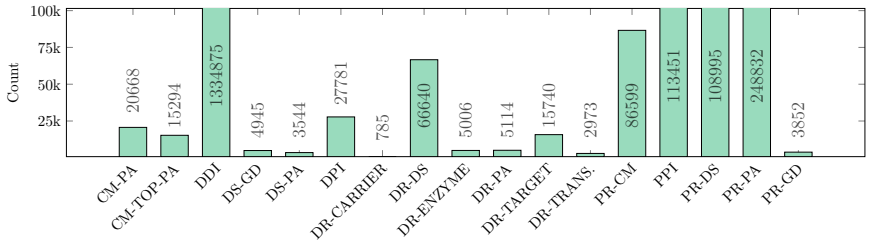


Figure 4: Statistics of the links part of the BioKG knowledge graph. DPI and PPI refer to drug-protein interactions and protein-protein interactions respectively, CM denotes complexes and other abbreviations follow definitions in Table 1.

- **drug–drug interactions (DDIs)**. drug–drug interactions represent the interactions between drugs which often happens because of the prescription of drug combinations *i.e.* polypharmacy. The data of DDIs in BioKG is collected exclusively from the DrugBank database where there are 1334875 instances of DDIs relationships in BioKG links.
- **protein relations to genetic disorders**. Proteins are the products of genes, and the protein–genetic disorder relations capture the associations between proteins and the disorders of their origin genes. There are 3852 associations between proteins and genetic disorders in BioKG which are extracted from the SwissProt database and their links to the OMIM genetic disorder database.
- **protein relations to diseases**. BioKG contains 108995 instances of relations between diseases and their associated proteins which is extracted from the KEGG database. All disease ids are set the Medical Subject Headings (MeSH) format and all protein ids are set to UniProt format to comply with the rest of BioKG.
- **protein–pathway associations**. The involvement of proteins in specific pathways is captured in protein–pathway relation in BioKG where there are 248832 instance of protein–pathway associations. These instances are collected from the UniProt, KEGG, Reactome and DrugBank databases.
- **disease–genetic disorder associations**. There are 4945 disease–genetic disorder relationships in BioKG which are extracted from the KEGG and MESH databases.
- **disease–pathway associations**. Associations between diseases and specific protein interaction chains (pathways) which describe the disease or describe another biological process related to it. BioKG contains 3544 disease–pathway associations exclusively extracted from the KEGG database.
- **drug–pathway associations**. There are 5114 associations between drugs and pathways in the BioKG which are exclusively extracted from the DrugBank database.
- **complex related associations**. Complexes are composites of proteins which represent a set of physically interacting proteins. BioKG contain data on complexes and their member proteins and associated pathways which is extracted exclusively from the Reactome database.

3.2.2 Properties. The properties part of BioKG contains the associations between the previously discussed biological entities and their different attributes as illustrated in Fig. 5. For example, protein attributes include their association to gene ontology entries and their sequence annotations. On the other hand, properties of drugs in BioKG are their associated side-effects, indications and ATC classification codes.

BioKG also contains other types of properties for pathways, disease and genetic disorders where these properties are often a categorization of these entity types into groups based on different type-specific criteria.

3.2.3 Metadata. The metadata part contains data about biological entities names, types, synonyms, etc. This part of the data is not meant to be used in the training of relational learning models, and it does not contain any attributes or important associations for biological entities. Our objective, however, in this part is to maximize the richness of metadata on each of the included biological entities to facilitate analysing their related insights and to allow for tracking history of changes of ids and synonyms of biological entities' databases entries.

4 BIODBLINKER

In this section, we discuss the motivation behind the BioDBLinker library and its implementation as well as its usage.

4.1 Motivation

Many biological knowledge bases contain overlapping or partially overlapping data. In order to extract the unique set of relations between a given set of entities it is therefore necessary to remove this duplication. This process is made more difficult by the fact that some data sources use different identifiers for the same entity. To overcome this issue it is necessary to parse entity ids into a unified id system. As we have found that this is a recurring step required when generating biological knowledge graphs we undertook to develop a library which could be reused for this purpose which we believe would be useful to others in the community when building biological knowledge graphs.

Current methods for mapping biological entities include online services which require manual data entry, or mapping files which require writing scripts to process mapping inputs. Our approach main objective is to tackle these issues by providing offline services for the mappings which can be used automatically/programmatically in various application.

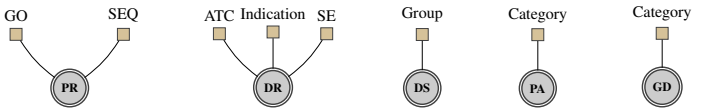


Figure 5: Illustration of BioKG main biological entities and their associated attributes in the properties part of the BioKG data.

4.2 Mapping Generation

BioDBLinker library provides a mapping generator class to generate the mappings required for the linker classes. The mapping generator parses the data in a number of formats from a number of different biological data sources to extract mappings from these data sources to other biological data sources in a unified manner. The source files for the mappings are downloaded from their respective biological data source at runtime allowing the mappings to be updated as new versions of the source files are released as required. For example, we use the UniProt mapping file to building mappings from/to UniProt ids and other 21 identifier systems. Similarly, we build mappings from the mapping files provided by the KEGG database to map its ids with other related databases.

4.3 Usage

BioDBLinker provides 3 main functions: (1) linking entity ids and names, (2) linking from entity ids to ids in other data sources and (3) retrieving ids for a given entity type in a specific database. The ids in a linker class can be accessed as properties of the class. When converting to names or other data sources a list of ids are passed to the function and a list of lists of names or mapped ids is returned, this allows for the case where an entity has multiple names or can be mapped to multiple entities in the target data sources.

4.4 Coverage

The BioDBLinker module covers 5 main data sources in relation to the BioKG knowledge graph, UniProt/SwissProt, Drugbank, KEGG, SIDER and Human Protein Atlas (HPA). Each of these data sources map to a number of other biological data sources, the BioDBLinker covers the mappings from/to each of the main mentioned databases to their respective associated databases.

5 BENCHMARKS

In this section, we discuss a set of four benchmarks that we provide with the BioKG. These benchmarks are focused on drug target discovery and drug-drug interactions related effects. In the following, we discuss the properties of each of our proposed benchmark datasets.

DDI-MINERAL Benchmark. The DDI-MINERAL benchmark consists of a set of drug-drug interactions and their associations to abnormalities of minerals levels in the human body where we focus on four elements: potassium, calcium, sodium and glucose. The benchmark contains 56017 drug-drug interactions of 922 drugs which is associated to an increased or decreased risk of an abnormal level of a mineral. For example, the interaction between the Canagliflozin and Miglitol drugs is associated with an increase of the risk of hypoglycemia (the condition in which your blood sugar (glucose) level is lower than normal).

The benchmark is formatted in triplets form where each entry represents a (drug, condition, drug) triplet. Each condition in the entries consist of two parts:risk modifier and risk type. The risk modifier is a basic increase or decrease flag while the risk type part denotes the risk type such as hyperkalemia, hyponatremia, etc.

This benchmark can be used in the assessment of relational models in the context of drug-drug interactions and their associated side-effects. The current popular polypharmacy side-effects benchmark provided by Zitnik et. al. [40] is based on a rather outdated TWOSIDES dataset [34] and it has a non-specific range of diverse side-effects. Our benchmark, on the other hand, is based on the DrugBank database (a recent, continually updated and more comprehensive resource) and focuses on a more specific set of DDI risks (e.g. the anomalies of minerals levels). This supports training more up to date and specific predictive models.

DDI-EFFICACY Benchmark. The DDI-EFFICACY benchmark is another drug-drug interaction based benchmark which is focused on the relation between these interaction and the therapeutic efficacy of the interacting drugs. The benchmark consists of 136127 drug-drug interactions of 3342 drugs and their effect (increase or decrease) on the therapeutic efficacy of the interacting drugs.

Similar to the DDI-MINERAL Benchmark benchmark, this benchmark provides a dataset which is focused on polypharmacy side-effects in relation to the drug efficacy. This benchmark can be used to assess the ability of relational models to predict such specific side-effects of interactions between drugs.

It is also worth noting that the types of polypharmacy side-effects included in both the DDI-EFFICACY and DDI-MINERAL benchmarks are not included at all in the current standard benchmarks such as Zitnik et. al. [40].

DPI-FDA Benchmark. The DPI-FDA consist of a set of drug target protein interactions of FDA approved drugs which is compiled from the KEGG and drug bank databases. This benchmark consist of 18928 drug protein interaction of 2277 drugs and 2654 proteins.

This benchmarks provides an extension to currently available benchmarks such as the DrugBank_FDA [38] and Yamanishi09 [39] benchmarks which have 9881 and 5127 DPIs respectively. Such an increase in the data size can enhance the training process of relational learning models and mitigate the generalization problems associated with the smaller benchmarks [26]. This extension of the number of DPIs provided in our benchmark is possible as we use the latest data releases of related biological databases unlike current benchmarks which we based on outdated versions (sometimes 10+ years old).

DPI-FDA-EXP Benchmark. The DPI-FDA-EXP is drug-protein association based benchmark which capture the effect of drugs on the expression levels of proteins in the living systems. The

benchmark contain 903429 statements on 1291 FDA approved drugs and their effects on the expression of 55196 proteins.

PPI-PHOSPHO Benchmark. Protein phosphorylation interactions is an enzymic protein–protein interaction which happens when a protein *i.e.* kinases, donates a phosphate group to another protein *i.e.* substrate. This type of PPIs is crucial for signalling in virtually any living cell. The PPI-PHOSPHO benchmark dataset is a kinase–substrate phosphorylation dataset which is based on the PhosphoSitePlus database [13] and the work of Hijazi et. al. [11]. The benchmark contains 25662 records in the format (<kinase>, PHOSPHORYLATES, <substrate>, <site>), where kinases and substrates are specific protein types represented using the proteins' UniProt ids and the site field represents the specific residue in substrate sequence which interacts with the kinase protein.

This benchmark provides a richer dataset for phosphorylation interactions which extends the currently used benchmarks [12, 28, 31] that suffer from limited coverage of kinases and substrates, and have fewer records due to dependence on older version of phosphorylation databases.

All the benchmarks can be downloaded from the biokg github repository at https://github.com/dsi-bdi/biokg/releases/download/v1.0.0/benchmarks.zip.

6 DISCUSSION

In this section, we discuss issues, lessons learned and other observations regarding the development and the use of BioKG knowledge graph in building relational learning models for biological applications.

6.1 Data Quality

In the process of building the BioKG knowledge graph, we tried to ensure the highest quality of all the data included by extracting data from curated sources exclusively. However, one of our included sources, the InterPro database for protein sequence annotations [21], is based on predicted sequence patterns using predefined rules and Markov models. We included this database as it is well-integrated with expert-curated SwissProt database and it compiles the most accurate parts of open sources for sequence annotations [21].

6.2 Availability

The implementation of the pipeline to build the BioKG contents is available at https://github.com/dsi-bdi/biokg. Downloadable BioKG contents (links, properties and metadata) are also available in the releasees section of the BioKG repository.

The BioDBLinker module is available as a Python module called *biodblinker* which can be installed using the standard *pip* process. The source code of the BioDBLinker module is also available on GitHub at https://github.com/dsi-bdi/biodblinker.

6.3 Limitations and Potential Issues

Despite the high coverage of biological entries in the BioKG, it still suffers from sparsity of data due to the unbalanced representation biological entities in open biological databases [24]. This unbalance is a result of the unbalanced research focus on specific entities, where some biological entities which are related to popular phenomenons are heavily studied, therefore, have richer database entries and annotations. This unbalance has a negative effect on relational learning models where they learn less efficient representations for the under-represented entities [24].

The use of BioKG and any other form of biological knowledge graphs can often lead to train-test data leakage when used without careful review of the relation between investigated phenomenons and the data in the knowledge graph. For example, the data on drug–drug combinations interactions (polypharmacy) is related to drug–protein interactions where two interacting drugs are often detected when they have the same protein target. The TWOSIDES database for instance uses DPIs to extract DDIs [34], therefore, using DPIs as extra data to support relational models in predicting DDIs can introduce indirect data leakage in such settings. We, therefore, suggest careful review of the relation between training knowledge graphs and investigated phenomenons in the testing data in biological predictive tasks to avoid such an issue.

Knowledge graphs and their embedding models are also biased towards well-studied biological entities which have better representation within the graph. Hence, the performance of models relying on biological KGs can suffer from low accuracy when executed on understudied or new biological entities which is absent from the graph. We, therefore suggest incorporating other forms of biological data such as protein sequences, protein structures and structures of chemical compounds into the predictive models to enhance their representations of the understudied entities.

6.4 Future Directions

We aim to provide updates to the BioKG in future works to keep it updated with the latest releases and changes of the source biological databases. We have recently investigated the principles and results presented here in the development of several state-of-the-art relational learning models [22–24, 26], and we aim to continue this line of work where we intend to assess the predictive accuracy and scalability of popular relational models on various other benchmarks based on the data introduced in this paper.

7 CONCLUSIONS

In this work, we proposed a new knowledge graph, BioKG, which covers a broad range of primary sources of biological data with the objective of supporting relational learning models on biological predictive tasks. The BioKG creation pipeline extracts data from open biological databases and provides them in a form of a graph of biological entities and their connections to each other along with their attributes and other related metadata. The contents of BioKG is compiled from expert-curated and popular sources to ensure high data quality and high level of integration.

We also provided a module for linking biological entities from different databases, the BioDBLinker which is based on open biological databases mappings. The module provides offline services for mapping between the different id systems for biological entities along with bidirectional name-id lookup services. The range and depth of resources covered as well as the flexibility in adding new sources arguably complements and extends currently available solutions, such as the Bio2RDF suite.

Furthermore, we have proposed a set of benchmarking datasets which can be are built from the drug–drug and drug–protein interactions data in the BioKG. These benchmarks cover different aspects related to both types of interactions and can be useful means for assessing the predictive accuracy of relational learning models in corresponding discovery tasks.

8 ACKNOWLEDGEMENTS

The work presented in this paper was supported by the CLARIFY project funded by European Commission under the grant number 875160, and by the Insight Centre for Data Analytics at the National University of Ireland Galway, Ireland (supported by the Science Foundation Ireland grant (12/RC/2289_P2).

REFERENCES

[1] Joanna S. Amberger, Carol A. Bocchini, François Schiettecatte, Alan F. Scott, and Ada Hamosh. 2015. OMIM.org: Online Mendelian Inheritance in Man (OMIM®), an online catalog of human genes and genetic disorders. *Nucleic Acids Research* 43 (2015), D789 – D798.

[2] Amos Bairoch. 2018. The Cellosaurus, a Cell-Line Knowledge Resource. *Journal of biomolecular techniques : JBT* 29 2 (2018), 25–38.

[3] François Belleau, Marc-Alexandre Nolin, Nicole Tourigny, Philippe Rigault, and Jean Morissette. 2008. Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics* 41 5 (2008), 706–16.

[4] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*. 2787–2795.

[5] Gene Ontology Consortium. 2005. The Gene Ontology (GO) project in 2006. *Nucleic Acids Research* 34 (2005), D322 – D326.

[6] The UniProt Consortium. 2010. The Universal Protein Resource (UniProt) in 2010. *Nucleic Acids Research* 38 (2010), D142 – D148.

[7] The UniProt Consortium. 2019. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research* 47 (2019), D506 – D515.

[8] David Croft and Gavin O'Kelly et. al. 2011. Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Research* 39 (2011), D691 – D697.

[9] Nikolai Hecker, Jessica Ahmed, Joachim von Eichborn, Mathias Dunkel, Karel Macha, Andreas Eckert, Michael K. Gilson, Philip E. Bourne, and Robert Preissner. 2012. SuperTarget goes quantitative: update on drug–target interactions. *Nucleic Acids Research* 40 (2012), D1113 – D1117.

[10] Micheal Hewett, Diane E. Oliver, Daniel L. Rubin, Katrina L. Easton, Joshua M. Stuart, Russ B. Altman, and Teri E. Klein. 2002. PharmGKB: the Pharmacogenetics Knowledge Base. *Nucleic acids research* 30 1 (2002), 163–5.

[11] Maruan Hijazi, Ryan Smith, Vinothini Rajeeve, Conrad Bessant, and Pedro R. Cutillas. 2020. Reconstructing kinase network topologies from phosphoproteomics data reveals cancer-associated rewiring. *Nature Biotechnology* 38 (2020), 493 – 502.

[12] Heiko Horn, Erwin Schoof, Jinho Kim, Xavier Robin, Martin L. Miller, Francesca Diella, Anita Palma, Gianni Cesareni, Lars Juhl Jensen, and Rune Linding. 2014. KinomeXplorer: an integrated platform for kinome biology studies. *Nature Methods* 11 (2014), 603–604.

[13] Peter V. Hornbeck, Jon M. Kornhauser, Sasha Tkachev, Bin Zhang, Elzbieta Skrzypek, Beth Murray, Vaughan Latham, and Michael Sullivan. 2012. PhosphoSitePlus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. *Nucleic Acids Research* 40 (2012), D261 – D270.

[14] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. 2016. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research* 44 (2016), D457 – D462.

[15] Craig Knox, Vivian Law, Timothy Jewison, Philip Liu, Son Ly, Alex Frolkis, Allison Pon, Kelly Banco, Christine Mak, Vanessa Neveu, Yannick Djoumbou, Roman Eisner, Anchi Guo, and David Scott Wishart. 2011. DrugBank 3.0: a comprehensive resource for 'Omics' research on drugs. *Nucleic Acids Research* 39 (2011), D1035 – D1041.

[16] Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. 2016. The SIDER database of drugs and side effects. *Nucleic Acids Research* 44 (2016), D1075 – D1079.

[17] Xin Liu, Feng Zhu, Xiaohua Ma, Lin Tao, Jingxian Zhang, Shengyong Yang, Yuquan Wei, and Y. Z. Chen. 2011. The Therapeutic Target Database: an internet resource for the primary targets of approved, clinical trial and experimental drugs. *Expert opinion on therapeutic targets* 15 8 (2011), 903–12.

[18] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *CIDR*. www.cidrdb.org.

[19] Carolyn J. Mattingly, Glenn T. Colby, John N. Forrest, and James L. Boyer. 2003. The Comparative Toxicogenomics Database (CTD). *Environmental Health Perspectives* 111 (2003), 793 – 795.

[20] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[21] Alex L. Mitchell and Terri K. Attwood et. al. 2019. InterPro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Research* 47 (2019), D351 – D360.

[22] Sameh K. Mohamed. 2020. Predicting tissue-specific protein functions using multi-part tensor decomposition. *Information Sciences* 508 (2020), 343–357.

[23] Sameh K Mohamed and Aayah Nounu. 2020. Predicting The Effects of Chemical-Protein Interactions On Proteins Using Tensor Factorisation. *AMIA Summits on Translational Science Proceedings* 2020 (2020), 430.

[24] Sameh K Mohamed, Aayah Nounu, and Vít Novácek. 2020. Biological applications of knowledge graph embedding models. *Briefings in Bioinformatics* (02 2020). https://doi.org/10.1093/bib/bbaa012 bbaa012.

[25] Sameh K. Mohamed and Vít Novácek. 2019. Link Prediction Using Multi Part Embeddings. In *ESWC (Lecture Notes in Computer Science, Vol. 11503)*. Springer, 240–254.

[26] Sameh K. Mohamed, Vít Novácek, and Aayah Nounu. 2020. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics* 36, 2 (2020), 603–610.

[27] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104 (2016), 11–33.

[28] John C. Obenauer, Lewis C. Cantley, and Michael B. Yaffe. 2003. Scansite 2.0: proteome-wide prediction of cell signaling interactions using short sequence motifs. *Nucleic acids research* 31 13 (2003), 3635–41.

[29] Rawan S. Olayan, Haitham Ashoor, and Vladimir B. Bajic. 2018. DDR: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics* 34 (2018), 1164 – 1173.

[30] Sandra E. Orchard, Mais G. Ammari, and Bruno Aranda et. al. 2014. The MIntAct project—IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research* 42 (2014), D358 – D363.

[31] Jiangning Song, Huilin Wang, Jiawei Wang, André Leier, Tatiana T. Marquez-Lago, Bingjiao Yang, Ziding Zhang, Tatsuya Akutsu, Geoffrey I. Webb, and Roger J. Daly. 2017. PhosphoPredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection. *Scientific Reports* 7 (2017).

[32] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. 2006. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research* 34 (2006), D535 – D539.

[33] Damian Szklarczyk, Andrea Franceschini, Michael Kuhn, Milan Simonovic, Alexander Roth, Pablo Mínguez, Tobias Doerks, Manuel Stark, Jean Muller, Peer Bork, Lars Juhl Jensen, and Christian von Mering. 2011. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research* 39 (2011), D561 – D568.

[34] Nicholas P. Tatonetti, Patrick Ye, Roxana Daneshjou, and Russ B. Altman. 2012. Data-driven prediction of drug effects and interactions. *Science translational medicine* 4 125 (2012), 125ra31.

[35] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *ICML (JMLR Workshop and Conference Proceedings, Vol. 48)*. JMLR.org, 2071–2080.

[36] Mathias Uhlén, Per Oksvold, Linn Fagerberg, Emma Lundberg, Kalle Jonasson, Mattias Forsberg, Martin Zwahlen, Caroline Kampf, Kenneth Wester, Sophia Hober, Henrik Wernérus, Lisa Björling, and Frederik Pontén. 2010. Towards a knowledge-based Human Protein Atlas. *Nature Biotechnology* 28 (2010), 1248–1250.

[37] Christian von Mering, Martijn A. Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. 2003. STRING: a database of predicted functional associations between proteins. *Nucleic acids research* 31 1 (2003), 258–61.

[38] David S. Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. 2008. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research* 36 (2008), D901–D906.

[39] Yoshihiro Yamanishi, Michihiro Araki, Alex Gutteridge, Wataru Honda, and Minoru Kanehisa. 2008. Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 24 (2008), i232 – i240.

[40] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34 (2018), i457 – i466.

Chapter 11

Discovering Protein Drug Targets

Bioinformatics, 36(2), 2020, 603–610
doi: 10.1093/bioinformatics/btz600
Advance Access Publication Date: 1 August 2019
Original Paper

Systems biology

Discovering protein drug targets using knowledge graph embeddings

Sameh K. Mohamed [1,2,*], Vít Nováček[1,2] and Aayah Nounu[3]

[1]Data Science Institute, College of Engineering and Informatics, [2]Insight Centre for Data Analytics, NUI Galway, Galway, Ireland and [3]MRC Integrative Epidemiology Unit, Bristol Medical School, University of Bristol, Bristol, UK

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on February 11, 2019; revised on July 20, 2019; editorial decision on July 25, 2019; accepted on July 27, 2019

Abstract

Motivation: Computational approaches for predicting drug–target interactions (DTIs) can provide valuable insights into the drug mechanism of action. DTI predictions can help to quickly identify new promising (on-target) or unintended (off-target) effects of drugs. However, existing models face several challenges. Many can only process a limited number of drugs and/or have poor proteome coverage. The current approaches also often suffer from high false positive prediction rates.

Results: We propose a novel computational approach for predicting drug target proteins. The approach is based on formulating the problem as a link prediction in knowledge graphs (robust, machine-readable representations of networked knowledge). We use biomedical knowledge bases to create a knowledge graph of entities connected to both drugs and their potential targets. We propose a specific knowledge graph embedding model, TriModel, to learn vector representations (i.e. embeddings) for all drugs and targets in the created knowledge graph. These representations are consequently used to infer candidate drug target interactions based on their scores computed by the trained TriModel model. We have experimentally evaluated our method using computer simulations and compared it to five existing models. This has shown that our approach outperforms all previous ones in terms of both area under ROC and precision–recall curves in standard benchmark tests.

Availability and implementation: The data, predictions and models are available at: drugtargets.insight-centre.org.

Contact: sameh.kamal@insight-centre.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The development of drugs has a long history (Drews, 2000). Until quite recently, pharmacological effects were often discovered using primitive trial and error procedures, such as applying plant extracts on living systems and observing the outcomes. Later, the drug development process evolved to elucidating mechanisms of action of drug substances and their effects on phenotype. The ability to isolate pharmacologically active substances was a key step towards modern drug discovery (Sneader, 2005; Terstappen *et al.*, 2007). More recently, advances in molecular biology and biochemistry allowed for more complex analyses of drugs, their targets and their mechanisms of action. The study of drug targets has become very popular with the objective of explaining mechanisms of actions of current drugs and their possible unknown off-target activities. Knowing targets of potential clinical significance also plays a crucial role in the process of rational drug development. With such knowledge, one can design candidate compounds targeting specific proteins to achieve intended therapeutic effects.

However, a drug rarely binds only to the intended targets, and off-target effects are common (Xie *et al.*, 2012). This may lead to unwanted adverse effects (Bowes *et al.*, 2012), but also to successful drug re-purposing, i.e. use of approved drugs for new diseases (Corbett *et al.*, 2012). To illustrate the impact off-target effects can

have in new therapy development, let us consider *aspirin* that is currently being considered for use as a chemopreventive agent (Rothwell *et al.*, 2010). However, such a therapy would be hampered by known adverse side-effects caused by long-term use of the drug, such as bleeding of upper gastrointestinal tract (Li *et al.*, 2017). After identifying the exact protein targets of *aspirin* that cause these adverse effects, the proteins can be targeted by newly developed and/or re-purposed drugs to avoid the unwanted side-effects of the proposed treatment.

Large-scale and reliable prediction of drug–target interactions (DTIs) can substantially facilitate development of such new treatments. Various DTI prediction methods have been proposed to date. Examples include chemical genetic (Terstappen *et al.*, 2007) and proteomic methods (Sleno and Emili, 2008) such as affinity chromatography and expression cloning approaches. These, however, can only process a limited number of possible drugs and targets due to the dependency on laboratory experiments and available physical resources. Computational prediction approaches have therefore received a lot of attention lately as they can lead to much faster assessments of possible drug–target interactions (Mei *et al.*, 2013; Yamanishi *et al.*, 2008).

The work of Yamanishi *et al.* (2008) was one of the first approaches to predict drug targets computationally. Their approach utilized a statistical model that infers drug targets based on a bipartite graph of both chemical and genomic information. The BLM-NII (Mei *et al.*, 2013) model was developed to improve the previous approach by using neighbour-based interaction-profile inference for both drugs and targets. More recently, (Cheng *et al.*, 2012a, b) proposed a new way for predicting DTIs, where they have used a combination of drug similarity, target similarity and network-based inference. The COSINE (Rosdah *et al.*, 2016) and NRLMF (Liu *et al.*, 2015) models introduced the exclusive use of drug–drug and target–target similarity measures to infer possible drug targets. This has an advantage of being able to compute predictions even for drugs and targets with limited information about their interaction data. However, these methods only utilized a single measure to model components similarity. Other approaches such as the KronRLS-MKL (Nascimento *et al.*, 2016) model used a linear combinations of multiple similarity measures to model the overall similarity between drugs and targets. Non-linear combinations were also explored in (Mei *et al.*, 2013) and shown to provide better predictions.

Recently, Hao *et al.* (2017) proposed a model called DNILMF that uses matrix factorization to predict drug targets over drug information networks. This approach showed significant improvements over other methods on standard benchmarking datasets (Hao *et al.*, 2017; Yamanishi *et al.*, 2008). All the previously discussed works were designed to operate on generic similarities of drug structure and protein sequence, therefore they can provide efficient predictions on new chemicals. More recently, approaches that incorporate prior knowledge about drugs and targets were proposed to enhance predictive accuracy on well-studied chemicals and targets. Such models may not be best suited to de novo drug discovery. However, they may provide valuable new insights in the context of drug repurposing and understanding the general mechanisms of drug action. The current state-of-the-art work in this context is arguably the DDR model (Olayan *et al.*, 2018), which uses a multi-phase procedure to predict drug targets from relevant heterogeneous graphs. The gist of the approach is to combine various similarity indices and random walk features gained from the input graphs by means of non-linear fusion. Similarly, the NeoDTI model (Wan *et al.*, 2019) predicts DTIs using supporting information about drugs and targets and a non-linear learning model over heterogeneous network data.

Despite continuous advances of similarity based approaches like DDR, these models depended on time-consuming training and prediction procedures as they need to compute the similarity features for each drug and target pair during both training and prediction. Also, the models still have a high false positive rate, especially when using large drug target interaction datasets like DrugBank_FDA (Olayan *et al.*, 2018).

Here, we propose a method utilizing prior knowledge about drugs and targets, similarly to the DDR and NeoDTI model. Our method overcomes the afore-mentioned limitations by approaching the problem as link prediction in knowledge graphs. Knowledge graphs are a data representation model that represents relational information as a graph, where the graph nodes represent entities and edges represent relations between them. Facts are modelled as (subject, predicate, object) (SPO) triples, e.g. (*Aspirin*, *Drug–Target*, *COX-1*), where a subject entity (drug) is connected to an object entity (target protein) through a predicate relation (Drug–Target). In recent years, knowledge graphs have been successfully used for knowledge representation and discovery in many different domains, including life sciences (Dumontier *et al.*, 2014; Lehmann *et al.*, 2014; Muñoz *et al.*, 2019).

Our work utilizes the fact that the current drug target knowledge bases like DrugBank (Wishart *et al.*, 2006) and KEGG (Kanehisa *et al.*, 2017) are largely structured as networks representing information about drugs in relationship with target proteins (or their genes), action pathways and targeted diseases. Such data can naturally be interpreted as a knowledge graph. The task of finding new associations between drugs and their targets can then be formulated as a link prediction problem based on knowledge graph embeddings (Nickel *et al.*, 2016).

We have proposed a new knowledge graph embedding based approach, TriModel, for predicting drug target interactions in a multi-phase procedure. We first used the currently available knowledge bases to generate a knowledge graph of biological entities related to both drugs and targets. We then trained our model to learn efficient vector representations (i.e. embeddings) of drugs and target in the knowledge graph. These representations were then used to score possible drug target pairs using a scalable procedure that has a linear time and space complexity. We compared our method to other state-of-the-art models using experimental evaluation on standard benchmarks. Our results show that the TriModel model outperforms all other approaches in areas under ROC and precision recall curve, metrics that are well suited to assessing general predictive power of ranking models (Davis and Goadrich, 2006).

2 Materials

In this section we discuss the datasets that we used to train and evaluate our model. We present the standard benchmarking datasets: Yamanishi_08 (Yamanishi *et al.*, 2008) and DrugBank_FDA (Wishart *et al.*, 2008), and we present statistics for elements in both datasets. We also discuss some flaws in the Yamanishi_08 dataset, and we present a new KEGG based drug targets dataset that addresses these flaws.

2.1 Standard benchmarks

The Yamanishi_08 (Yamanishi *et al.*, 2008) and DrugBank_FDA (Wishart *et al.*, 2008) datasets represent the most frequently used gold standard datasets in the previous state-of-the-art models for

Table 1. Statistics of elements in the benchmarking datasets used in this work

Dataset	Group	Drugs	Proteins	DTIs	Corruptions	P2N
Yamanishi_08	E	445	664	2926	≈300K	1.00%
	IC	210	204	1476	≈41K	3.57%
	GPCR	223	95	635	≈21K	3.03%
	NR	54	26	90	1314	6.67%
	All	791	989	5127	≈777K	0.66%
DrugBank_FDA	—	1482	1408	9881	≈2.1M	0.48%
KEGG_MED	—	4284	945	12 112	≈4M	0.30%

Note: The DTIs column represent the number of known drug target interactions, the Corruptions column represent the number of all possible combinations of drugs and targets that are not in the known drug target interactions which is used as negative in model training and evaluation, and the P2N column represents the ratio of positive to negative instances.

predicting drug targets (Olayan *et al.*, 2018). The DrugBank_FDA (Wishart *et al.*, 2008) dataset consists of a collection of DTIs of FDA approved drugs that are gathered from DrugBank Database (https://www.drugbank.ca). The Yamanishi_08 dataset is a collection of known drug target interactions gathered from different sources like KEGG BRITE (Kanehisa *et al.*, 2006), BRENDA (Schomburg *et al.*, 2004), SuperTarget (Günther *et al.*, 2007) and DrugBank (Wishart *et al.*, 2008). It consists of four groups of drug target interactions corresponding to four different target protein classes: (i) enzymes (E), (ii) ion-channels (IC), (iii) G-protein-coupled receptors (GPCR) and (iv) nuclear receptors (NR). The data in these groups vary in terms of size and positive to negative ratios as shown in Table 1, ranging from 90 known DTIs with 1:15 as in the NR group to 2926 DTIs with 1:100 in the E group. These properties of the datasets affect the effectiveness of both training and evaluating models that use them. For example, the NR DTIs group has the largest positive to negative ratio among all the groups in the Yamanishi_08 dataset and therefore they are the easiest for predictive models in terms of evaluation. Contrary to that, the state-of-the-art models show the worst evaluation results on the NR group compared to other groups. This happens due to the low number of available DTIs training instances, which affects the models' generalization on the training data.

2.2 New KEGG based benchmarking dataset

The Yamanishi_8 benchmarking dataset was published in 2008, and it contained drug target interactions from various sources including the KEGG BRITE, BRENDA and SuperTarget databases (Yamanishi *et al.*, 2008). In recent years, these sources have witnessed multiple developments [modifications, deletions, additions of many brand new records to their data (Hecker *et al.*, 2012; Placzek *et al.*, 2017)]. These modification have directly affected the Yamanishi_08 dataset, where a subset of the identifiers of both its drugs and targets has been modified through these developments. This affects the ability to link these drugs and targets to their corresponding properties, e.g. associated pathways, diseases, or other biological entities in the recent versions of biological knowledge bases. These modifications have also included various newly discovered drug target interactions that are not included in the Yamanishi_08 dataset. For example, the KEGG database alone contains 12 112 drug target interactions, while the total number of drug target interactions in the Yamanishi_08 dataset is only 5127.

To overcome these limitations, we propose a new drug target interaction benchmarking dataset that depends on recent versions of biological knowledge bases and includes a larger set of drug target interactions than the Yamanishi_08 dataset. We propose KEGG_MED, a dataset which is collected by extracting all the drug target interactions from the KEGG medicus database (https://www.genome.jp/kegg/medicus.html). The KEGG_MED dataset contains 4284 drugs and 945 targets which are connected with 12 112 drug target interactions. Table 1 shows a summary of statistics of the content on the dataset. Later in this paper, we report our results on this new suggested benchmark (in addition to the comparative validation on DrugBank_FDA) so that future approaches can be compared to our model.

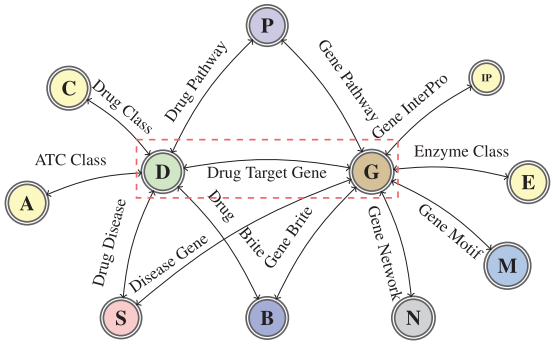


Fig. 1. A graph schema for a knowledge graph about drugs, their target genes, pathways, diseases and gene networks extracted from KEGG and UniProt databases

2.3 Supporting knowledge graphs

Link prediction with knowledge graph embedding models require data to be modelled in a graph form, where the objective is to predict new links between graph entities. In the case of drug target discovery, we use supporting data from biomedical knowledge bases to generate informative graphs around drug target interactions. We generate a knowledge graph for each dataset to provide descriptive features for both drugs and targets. These knowledge graphs are extracted from different sources like KEGG (Kanehisa *et al.*, 2017), DrugBank (Wishart *et al.*, 2006), InterPro (Mitchell *et al.*, 2019) and UniProt (Consortium, 2017). In our study we use a customized set of knowledge assertions about both drugs and targets. Supplementary Appendix S1 and Supplementary Table S1 contain more information about the relation types present in each knowledge graph, and about their construction. For further information about the construction of such knowledge bases we refer to the work of Himmelstein *et al.* (2017) that provides a study of systematic integration of biological knowledge for learning drug–target interactions.

We generate a group-specific knowledge graph of information extracted from KEGG and UniProt for each DTI groups in the Yamanishi_8 dataset, while we use the DrugBank with UniProt knowledge bases to model information about DTIs of the DrugBank_FDA dataset. The information extracted in both cases is modelled as a graph of interconnected biological entities (schema shown in Fig. 1).

3 Methods

The knowledge graph embedding models we use follow a generative approach to learn low-rank embedding vectors for knowledge

entities and relations. For learning the embeddings, multiple techniques can be used, such as tensor factorization [c.f. the DistMult model (Bordes *et al.*, 2013)] or latent distance similarity [c.f. the TransE model (Yang *et al.*, 2015)]. The goal of all these techniques is to model possible interactions between graph embeddings and to provide scores for possible graph links. In the following, we provide details on the knowledge graph embedding procedure and the design of our proposed model, TriModel.

3.1 Knowledge graph embedding

Knowledge graph embedding (KGE) models learn a low rank vector representation of knowledge entities and relations that can be used to rank knowledge assertions according to their factuality. They are trained in a multi-phase procedure. First, a KGE model initializes all embedding vectors using random noise values. It then uses these embeddings to score the set of true and false training facts using a model-dependent scoring function. The output scores are then passed to the training loss function to compute training error. These errors are used by optimizers like AMSGrad (Reddi *et al.*, 2018) to generate gradients and update the initial embeddings, where the updated embeddings give higher scores for true facts and lower scores for false facts. This procedure is performed iteratively for a set of iterations, i.e. epochs in order to reach a state where the learnt embeddings provide best possible scoring for both true and false possible facts.

In the rest of this paper, we use $\mathbb{E}$ and $\mathbb{R}$ to denote the set all entities and relations in a knowledge graph respectively, where N_e and N_r represent the number of instances in $\mathbb{E}$ and $\mathbb{R}$ respectively. We also use Θ_E and Θ_R which denote the embeddings of entities and relations respectively, where $\Theta_E(i)$ is the embedding of entity i, $\Theta_R(j)$ is the embedding of relation j, and $f_m(s, r, o, \Theta)$ denotes the score of the fact that a subject entity s is connected to an object entity o with a relation r based on the embedding values Θ of the model m.

3.2 Embeddings representation

TriModel is a knowledge graph embedding model based on tensor factorization that extends the DistMult (Yang *et al.*, 2015) and ComplEx (Trouillon *et al.*, 2016) models. It represents each entity and relation using three embedding vectors such that the embedding of entity i is $\Theta_E(i) = \{e_i^1, e_i^2, e_i^3\}$ where all embedding vectors have the same size K (a user-defined embeddings size). Similarly, the embedding of relation j is $\Theta_R(j) = \{w_j^1, w_j^2, w_j^3\}$. e^m and w^m denote the m part of the embeddings of the entity or the relation, and $m \in \{1, 2, 3\}$ represents the three embeddings parts. The embeddings in the TriModel model are initially with random values generated by the Glorot uniform random generator (Glorot and Bengio, 2010). The embedding vectors are then updated during the training procedure to provide optimized scores for the knowledge graph facts.

3.3 Training procedure

The TriModel is a knowledge graph embedding model that follows the multi-phase procedure discussed in Section 3.1 to effectively learn a vector representation for entities and relation of a knowledge graph. First, the model initializes its embeddings with random noise. It then updates them by iterative learning on the training data. In each training iteration i.e. epoch, the model splits the training data into mini-batches and executes its learning pipeline over each batch. The learning pipeline of the model learns the embeddings of entities and relations by minimizing a negative softmax log-loss that maximizes the scores of true facts and minimizes the scores of unknown facts (assumed false during training). This loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{spo}^{\text{TriModel}} &= -\phi_{spo} + \log\left(\textstyle\sum_{o'} \exp\left(\phi_{spo'}\right)\right) \\ &\quad -\phi_{spo} + \log\left(\textstyle\sum_{s'} \exp\left(\phi_{s'po}\right)\right) \\ &+ \frac{\lambda}{3}\sum_{k=1}^{K}\sum_{m=1}^{3}\left(|e_s^m|^3 + |w_p^m|^3 + |e_o^m|^3\right)\end{aligned} \tag{1}$$

where x' represents an entity $e : e \neq x, e \in \mathbb{E}$, e_i^m is the embedding part m of the entity embedding $\Theta_E(i)$, w_i^m is the embedding part m of the relation embedding $\Theta_R(i)$, ϕ_{spo} denotes the score of the triple (s, p, o), m denotes the embedding part index, λ denotes a configurable regularization weight parameter and $|x|$ is the absolute of x. The term $\frac{\lambda}{3}\sum_{k=1}^{K}\sum_{m=1}^{3}(|e_s^m|^3 + |w_p^m|^3 + |e_o^m|^3)$ is the nuclear 3-norm, which is a regularization term (Lacroix *et al.*, 2018) that enhances model generalization over datasets with large entity vocabularies.

The scores of the TriModel model are computed using an embeddings interaction function (scoring function) that is defined as follows:

$$f_{\text{TriModel}}(s, r, o, \Theta) = \sum_{}^{K} e_s^1 w_r^1 e_o^3 + e_s^2 w_r^2 e_o^2 + e_s^3 w_r^3 e_o^1. \tag{2}$$

It uses a set of three interactions: one symmetric interaction: $(e_s^2 w_p^2 e_o^2)$ and two asymmetric interactions: $(e_s^1 w_p^1 e_o^3)$ and $(e_s^3 w_p^3 e_o^1)$ for a convenient graphical explanation of the interaction (see Supplementary Fig. S2). This approach models both symmetry and asymmetry in simple form similar to the DistMult (Yang *et al.*, 2015) model where the DisMult model can be seen as a special case of the TriModel model if the first and third embeddings parts are equivalent $(e^1 = e^3)$. We include more details about the training procedure in Supplementary Appendix S2.

4 Results

In this section we describe the configuration of the data used in the experimentation, the evaluation protocol, the setup of our experiments and the results and findings of our experiments. We also compare the predictive accuracy of our model to selected existing approaches, including the state-of-the-art one.

4.1 Evaluation protocol

In order to facilitate comparison with the state-of-the-art models, we use a 10-fold cross validation (CV) to evaluate our model on the Yamanishi_08 and DrugBank_FDA datasets. First, we split the drug target interaction data into 10 splits i.e. folds. We then evaluate the model 10 times on each split, where the model is trained on the other 9 splits. This procedure is repeated 5 times and average results across these runs are reported. This is to further minimize the impact of data variability on the result stability.

In each training configuration we use the known drug target interactions as positives, and all other possible combinations between the investigated dataset drugs and protein targets as negatives. This yields different positive to negative ratios since the datasets have different number of drugs, targets and drug target interactions (see Table 1 for exact statistics of the ratios for each dataset).

We use the area under the ROC and precision recall curves (AUC-ROC and AUC-PR respectively) as an indication of the predictive accuracy of our model. We compute both metrics on the testing data (DTIs), where we divide the testing data into three groups: (i) S_p, containing testing drug target interactions where both the drug and the target are involved in known drug target interactions in the training data, (ii) S_d, containing testing drug target

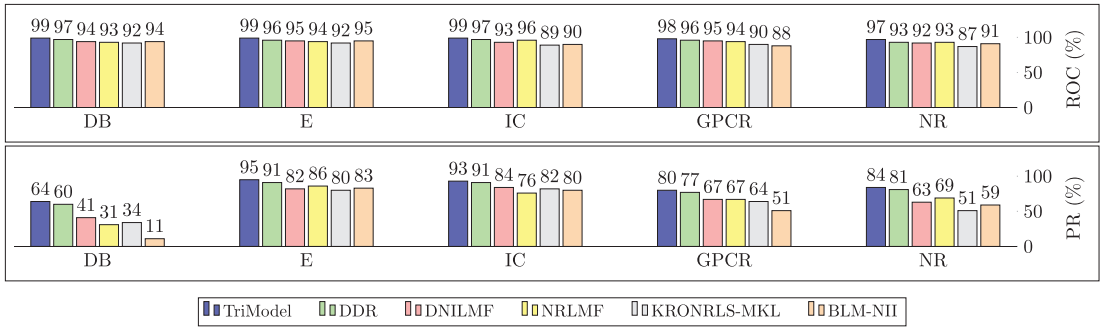


Fig. 2. Bar chart for the values of the area under the roc curve (AUC-ROC) and area under the precision recall curve (AUC-PR) for the TriModel compared to other state-of-the-art models on standard benchmarking datasets. All values are rounded to two digits and multiplied by 100 to represent a percentage (%). DB represents the DrugBank_FDA dataset

interactions which contain drugs that have no known drug target interactions in the training data, (iii) S_t, containing testing data of targets that has not involved in any known drug target interactions in the training data. The main reason for splitting the data this way was that one of the methods could not be compared with the others on the S_t, S_p data. The largest S_p group, however, generally exhibits least fluctuations across particular cross-validation runs, and therefore it is arguably most representative in terms of the comparative validation.

We also compute aggregated weighted AU-ROC, AU-PR scores for comparing the different models regardless the data group. These scores are defined as follows:

$$M = \sum_{g} \omega_g \cdot M_g, \tag{3}$$

where $g \in \{S_p, S_d, S_t\}$, M represents the aggregated score (AUC-ROC or AUC-PR), M_g is the specific score value for the group g, and ω_g is the weight of the particular data group computed by dividing the number of instances in g by the total number of instances in $S_p \cup S_d \cup S_t$.

4.2 Experimental setup

We use the supporting knowledge graph to perform a grid search to learn the model's best hyperparameters. In all of our experiments we initialize our model embeddings using the Glorot uniform random generator (Glorot and Bengio, 2010) and we optimize the training loss using the AMSGrad optimizer (Reddi *et al.*, 2018), where the learning rate (lr) $\in \{0.01, 0.02, 0.03\}$, embeddings size $(K) \in \{50, 100, 150, 200\}$ and batch size $(b) \in \{128, 256, 512, 1024, 4000\}$. The rest of the grid search hyper parameters are defined as follows: the regularization weight $(\lambda) \in \{0.1, 0.3, 0.35, 0.01, 0.03, 0.035\}$, dropout $(d) \in \{0.0, 0.1, 0.2, 0.01, 0.02\}$. The number of training epochs is fixed to 1000. The outcome best parameter for this grid search is included in Supplementary Table S2.

We use Tensorflow framework (GPU) along with Python 3.5 to perform our experiments. All experiments were executed on a Linux machine with processor Intel(R) Core(TM) i70.4790K CPU @ 4.00 GHz, 32 GB RAM, and an nVidia Titan Xp GPU. We include the training runtime of the TriModel model for each cross-validation iteration for all the investigated benchmarks in Supplementary Figure S1.

4.3 Comparison with state-of-the-art models

We evaluate our model on the Yamanishi_08 and DrugBank_FDA datasets, and we compare our results to the following state-of-the-art models: DDR (Olayan *et al.*, 2018), NRLMF (Hao *et al.*, 2017), NRLMF (Liu *et al.*, 2015), KRONRLS-MKL (Nascimento *et al.*, 2016), COSINE (Lim *et al.*, 2016) and BLM-NII (Mei *et al.*, 2013). The comparison is made using the metrics of area-under-the-ROC (AUC-ROC) and precision–recall (AUC-PR) curves.

Figure 2 presents overall results in terms of the AUC-ROC and AUC-PR scores for all compared models. The overall scores are combined across all testing configurations (S_p, S_d, S_t) for each dataset, where each specific score is computed as described in Eq. 3.

The results show that the TriModel model outperforms all other models in terms of AUC-ROC and AUC-PR on every benchmarking dataset. The TriModel model achieves a better AUC-PR score with a margin of 4%, 2%, 3%, 3%, 4% on E, IC, GPCR, NR and DrugBank_FDA datasets respectively. It should be noted that we did not include the COSINE method in Figure 2 as it is specifically designed to predict new drugs that do not have DTIs in the training phase. As such, the description of the method only reports accuracy on the new drug configuration (S_d), while the presented combined scores require values of all three evaluation configurations.

Table 2 shows a detailed comparison of the TriModel model and state-of-the-art models on all the standard benchmarking datasets for the different evaluation settings S_p, S_d and S_t. It also shows the relative number (in per cent) of drug–target statements available for each of the three validation settings.

The results in Table 2 show that the TriModel model outperforms other state-of-the-art models on 13 out of 15 different AUC-ROC experimentation configurations. In case of AU-PR, our model is better 14 out of 15 configurations. The results also show that the experimental configurations where our model is not the best represent a small portion of the total number of DTIs, while the TriModel model provides consistently better results for the largest S_p partition of the validation data.

Table 2 also show the results of the TriModel model on our proposed KEGG_MEDD dataset, where the model's AUC-PR scores are 0.18, 0.18 and 0.94 and its AUC-ROC scores are 0.81, 0.58 and 0.99 on the configurations S_d, S_t and S_p respectively. No comparison with existing tools has been performed as their published versions cannot be directly applied to this dataset.

4.4 Limitations

Despite the very promising results achieved by the prior knowledge-based models like DDR and TriModel, their predictive capabilities are best suited to finding new associations between well-studied drugs and targets (useful for instance in the drug repurposing context). If one needs predictions for de novo drug discovery, the models that utilize drug structure and target sequence similarities (e.g.

Table 2. A comparison with state-of-the-art models on standard datasets using multiple configurations (S_p, S_d, S_t)

M.	Model	Ft.	E			IC			GPCR			NR			DB			KM		
	Config.		S_d	S_t	S_p	S_d	S_t	S_p	S_d	S_t	S_p	S_d	S_t	S_p	S_d	S_t	S_p	S_d	S_t	S_p
	Count (%)		4%	5%	91%	4%	1%	95%	5%	4%	91%	10%	4%	86%	4%	11%	85%	5%	3%	92%
AUC-ROC	BLM-NII	Structure	0.73	0.89	0.96	0.83	0.89	0.91	0.85	0.87	0.88	0.88	0.85	0.91	0.71	0.75	0.90	-	-	-
	COSINE		0.80	-	-	0.82	-	-	0.88	-	-	0.89	-	-	0.77	-	-	-	-	-
	KRONRLS-MKL		0.71	0.88	0.93	0.77	0.86	0.90	0.81	0.84	0.91	0.79	0.76	0.87	0.79	0.81	0.88	-	-	-
	NRLMF		0.75	0.90	0.95	0.80	0.93	0.98	0.87	0.92	0.95	0.88	0.83	0.93	0.89	0.80	0.93	-	-	-
	DNILMF		0.81	0.92	0.96	0.81	0.92	0.94	0.86	0.92	0.96	0.83	0.83	0.92	0.90	0.82	0.95	-	-	-
	DDR	Ext.	0.84	0.92	0.97	**0.94**	0.97	0.98	0.91	**0.93**	0.96	**0.90**	**0.88**	0.92	0.91	0.86	0.96	-	-	-
	TriModel		**0.95**	**0.96**	**0.99**	0.93	**0.98**	**0.99**	**0.92**	0.86	**0.99**	0.89	0.85	**0.99**	**0.94**	**0.94**	**0.99**	**0.81**	**0.58**	**0.99**
AUC-PR	BLM-NII	Structure	0.22	0.73	0.86	0.37	0.61	0.83	0.35	0.37	0.53	0.35	0.41	0.62	0.03	0.05	0.12	-	-	-
	COSINE		0.35	-	-	0.36	-	-	0.40	-	-	0.56	-	-	0.30	-	-	-	-	-
	KRONRLS-MKL		0.07	0.07	0.87	0.23	0.23	0.86	0.31	0.37	0.67	0.49	0.46	0.51	0.22	0.18	0.35	-	-	-
	NRLMF		0.28	0.76	0.89	0.30	0.61	0.79	0.36	0.55	0.69	0.49	0.45	0.72	0.28	0.23	0.32	-	-	-
	DNILMF		0.30	0.76	0.85	0.30	0.61	0.87	0.31	0.56	0.70	0.41	0.52	0.66	0.24	0.21	0.42	-	-	-
	DDR	Ext.	0.73	0.82	0.92	0.69	0.80	0.92	0.63	0.61	0.79	0.71	0.64	0.83	0.44	0.39	0.61	-	-	-
	TriModel		**0.78**	**0.83**	**0.96**	**0.76**	**0.87**	**0.95**	**0.81**	**0.73**	**0.80**	**0.87**	**0.77**	**0.84**	**0.59**	**0.62**	**0.64**	**0.18**	**0.18**	**0.94**

Note: The state-of-the-art results were obtained from (Olayan *et al.*, 2018). The count (%) represents the percentage of the configuration instances, and the DB and KM columns represent DrugBank_FDA and KEGG_MED respectively. All the experimental configurations on all the datasets are evaluated using a 10-fold cross validation which is repeated 5 times. The **M.** column represents metrics. The **Ft.** column represents model's feature type. The *structure* feature type represents protein and drug structure based features and *Ext.* denotes extensive prior knowledge features. Underlined scores represent the best scores in their feature category while the overall best results are in bold and highlighted with green colour. (Color version of this table is available at *Bioinformatics* online.)

BLM-NII, COSINE, KRONRLS-MKL, NRLMF or NRLMF) will likely deliver better results.

4.5 Web application for exploring the TriModel predictions

To let users explore our results, we have designed a web application (Hosted at: http://drugtargets.insight-centre.org). The application allows for searching the predictions of the TriModel model. One can look for predictions using either drugs or targets as queries. Queries concerning multiple entities are possible simply by appending new terms to the search query. The results are presented as a table of the TriModel model scores of all the possible drug–target associations of the searched term.

The predictions provided by the web application are learnt by training the TriModel model on all the Yamanishi_08 dataset. The prediction scores are then computed for all possible drug–target combinations induced by the dataset. The scores of known drug interactions in the Yamanishi_08 dataset are set to 1, while the scores of all other drug target interactions are the normalized outcome of the TriModel predictions. The table of predictions in the application indicates the origin of each score, where a unique label '*Experimental Evidence*' is given to known DTIs and another label '*Model Prediction*' is assigned to the predicted scores.

5 Discussion

In the following we discuss possible reasons for the improved performance of our approach when compared to existing methods. We also review the limitations of the current DTI prediction benchmarks and discuss impact of data stratification on the predictive power of the models. Last but not least, we present tentative results in expert-based validation of predictions of our model that are not covered by the benchmark datasets. These results show high promise in terms of actual new discoveries predicted by our model.

5.1 Distinctive features of the presented approach

The relative success of the TriModel model can be attributed to two distinctive features not present in the state-of-the-art models. Firstly, we model input for the training as knowledge graphs. This allows for encoding multiple types of associations within the same graph and thus utilizing more complex patterns. Other models that use graph-based data are limited in this respect as they only employ networks with single relation type. Secondly, the TriModel model uses a generative approach to learn efficient representations for both drugs and their targets. This approach enables scalable predictions of large volumes of drug–target interactions as it uses linear training time (Nickel *et al.*, 2016) and constant prediction time, which is not the case of the existing works. Furthermore, the TriModel model is able to predict other biological associations within the training data (e.g. drug and target pathways) with no extra computational effort. This shows substantial promise for further development of this technique.

5.2 Impact of data stratification on the predictive power

The Yamanishi_08 dataset is divided into four groups of DTIs according to the functionality of the target proteins. The groups are enzymes (E), ion-channels (IC) G-protein-coupled receptors (GPCR) and nuclear receptors (NR). The objective of this categorization is to distinguish between models specifically tailored to predicting targets associated with a particular drug class (Yamanishi *et al.*, 2008). Olayan *et al.* (2018) confirmed that organizing the drug target interactions into groups according to the target's biological functionality enhances the predictive accuracy of models trained on such stratified data.

Based on our observations, we suggest a different explanation. The differences in performance appear to correlate with the relative numbers of negative examples in the grouped and full dataset configuration. Table 1 shows that the full Yamanishi_08 dataset configuration has a 0.66% positive to negative ratio, while the groups E, IC, GCPR and NR have 1, 3.57, 3.03 and 6.67% respectively. These differences can explain the variability of model performance quite well, since predicting positive instances is generally harder with more negatives present in the data (Liu *et al.*, 2007). In addition, dividing the DTI information gives rise to groups like the GPCR and NR groups. These contain only a small number of true DTIs (635 and 90 DTIs respectively), which further hampers the ability of models to generalize well (as we show in Section 2).

5.3 Validating the discovery potential of TriModel

Good performance of a model in benchmark tests is no doubt important. For various reasons like overfitting or training data

imbalances, however, good benchmark results may not necessarily mean that the model can effectively support new discoveries.

Laboratory validation can ultimately confirm the model predictions as actual discoveries, but this is costly and time-consuming to be done at large scale. One can, however, perform alternative validations of the predictions using data that was not used for training the model. Such complementary validation can provide stronger foundations for claiming a model has high generalization power.

We have performed a complementary validation of the TriModel's predictions by manual analysis of top-10 drug–target associations per each of the examined benchmarking datasets. To decide whether or not the associations are true positives, we reviewed available literature. We only validated the predictions that were not part of the training data. The validation outcome shows that the TriModel model achieves 7 out of 10, 7 out of 10, 8 out of 10, 7 out of 10 and 6 out of 10 true predictions on the E, IC, GPCR, NR, DB datasets respectively. A detailed version of the validated predictions is included in Supplementary Table S3.

One can easily see that our model puts actual drug–target introductions (some of which were only recently discovered) high up in the result list. This is very promising for further development of the model and its deployment in clinical application scenarios.

6 Conclusions and future work

In this work, we have approached the problem of predicting new drug targets as a link prediction task in biomedical knowledge graphs. We have presented the TriModel model, a knowledge graph embedding model that can efficiently predict new drug target interactions. We have generated knowledge graphs of biological entities related to drugs and targets using available biological knowledge bases like KEGG, UniProt and DrugBank. We have then used these knowledge graphs to train the TriModel model to learn efficient vector representation for both drugs and targets. In experiments using a standard benchmark data, we have demonstrated that the TriModel model outperforms state-of-the-art models in terms of both the area under ROC and precision recall curves.

Our study has also led to several secondary findings and contributions. We have shown that dividing datasets of drug target interactions into groups based on target properties does not positively affect the predictive accuracy of computation models. It can result in groups with very few drug target interactions, which negatively affects the accuracy of learnt models. Last but not least, we have developed a new KEGG based drug target interactions dataset that tackles the issues in the Yamanishi_08 dataset, and provides a richer set of up-to-date drug target interactions.

In future, we intend to explore how incorporation of more context data relevant to the target prediction problem can further improve the accuracy of our model. Last but not least, we will validate selected predictions of our model in laboratory experiments to demonstrate the clinical relevance of our results.

Funding

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

Conflict of Interest: none declared.

References

Bordes,A. *et al.* (2013) Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795.

Bowes,J. *et al.* (2012) Reducing safety-related drug attrition: the use of in vitro pharmacological profiling. *Nat. Rev. Drug Discov.*, **11**, 909.

Cheng,F. *et al.* (2012a) Prediction of chemical–protein interactions network with weighted network-based inference method. *PLoS One*, **7**, 1–13.

Cheng,F. *et al.* (2012b) Prediction of drug–target interactions and drug repositioning via network-based inference. *PLoS Comput. Biol.*, **8**, e1002503.

Consortium,T.U. (2017) Uniprot: the universal protein knowledgebase. *Nucleic Acids Res.*, **45**, D158–D169

Corbett,A. *et al.* (2012) Drug repositioning for Alzheimer's disease. *Nat. Rev. Drug Discov.*, **11**, 833.

Davis,J. and Goadrich,M. (2006) The relationship between precision–recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240. ACM.

Drews,J. (2000) Drug discovery: a historical perspective. *Science*, **287**, 1960–1964.

Dumontier,M. *et al.* (2014) Bio2rdf release 3: a larger, more connected network of linked data for the life sciences. In: *Proceedings of the ISWC* 2014, pp. 401–404.

Glorot,X. and Bengio,Y. (2010) Understanding the difficulty of training deep feedforward neural networks. In: *AISTATS, Volume 9 of JMLR Proceedings*, pp. 249–256. JMLR.org.

Günther,S. *et al.* (2007) Supertarget and matador: resources for exploring drug–target relationships. *Nucleic Acids Res.*, **36**, D919–D922.

Hao,M. *et al.* (2017) Predicting drug–target interactions by dual-network integrated logistic matrix factorization. *Sci. Rep.*, **7**, 40376.

Hecker,N. *et al.* (2012) Supertarget goes quantitative: update on drug–target interactions. *Nucleic Acids Res.*, **40**, D1113.

Himmelstein,D.S. *et al.* (2017) Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, **6**, e26726.

Kanehisa,M. *et al.* (2006) From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.*, **34**, D354–D357.

Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.

Lacroix,T. *et al.* (2018) Canonical tensor decomposition for knowledge base completion. In: *ICML, Volume 80 of JMLR Workshop and Conference Proceedings*, pp. 2869–2878. JMLR.org.

Lehmann,J. *et al.* (2014) DBpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web J*, **6**, 167–195.

Li,L. *et al.* (2017) Age-specific risks, severity, time course, and outcome of bleeding on long-term antiplatelet treatment after vascular events: a population-based cohort study. *Lancet*, **390**, 490–499.

Lim,H. *et al.* (2016) Improved genome-scale multi-target virtual screening via a novel collaborative filtering approach to cold-start problem. *Sci. Rep.*, **6**, 38860.

Liu,H. *et al.* (2015) Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics*, **31**, i221–i229.

Liu,T.-Y. *et al.* (2007) Learning to rank for information retrieval. *Found. Trends Inf. Retrieval*, **3**, 225–331.

Mei,J.-P. *et al.* (2013) Drug–target interaction prediction by learning from local information and neighbors. *Bioinformatics*, **29**, 238–245.

Mitchell,A.L. *et al.* (2019) Interpro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Res.*, **47**, D351–D360.

Muñoz,E. *et al.* (2019) Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. *Brief. Bioinf.*, **20**, 190–202.

Nascimento,A.C. *et al.* (2016) A multiple kernel learning algorithm for drug—target interaction prediction. *BMC Bioinformatics*, **17**, 46.

Nickel,M. *et al.* (2016) A review of relational machine learning for knowledge graphs. *Proc. IEEE*, **104**, 11–33.

Olayan,R.S. *et al.* (2018) DDR: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics*, **34**, 1164–1173.

Placzek,S. *et al.* (2017) Brenda in 2017: new perspectives and new tools in Brenda. *Nucleic Acids Res.*, **45**, D380.

Reddi,S. *et al.* (2018) On the convergence of Adam and beyond. In: ICLR.

Rosdah,A.A. *et al.* (2016) Mitochondrial fission—a drug target for cytoprotection or cytodestruction? *Pharmacol. Res. Perspect.*, **4**, e00235.

Rothwell,P.M. *et al.* (2010) Long-term effect of aspirin on colorectal cancer incidence and mortality: 20-year follow-up of five randomised trials. *Lancet*, **376**, 1741–1750.

Schomburg,I. *et al*. (2004) Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Res*., **32**, 431D–D433.

Sleno,L. and Emili,A. (2008) Proteomic methods for drug target discovery. *Curr. Opin. Chem. Biol*., **12**, 46–54.

Sneader,W. (2005) *Drug Discovery: A History*. John Wiley & Sons.

Terstappen,G.C. *et al*. (2007) Target deconvolution strategies in drug discovery. *Nat. Rev. Drug Discov*., **6**, 891.

Trouillon,T. *et al*. (2016) Complex embeddings for simple link prediction. In: *ICML, Volume 48 of JMLR Workshop and Conference Proceedings*, pp. 2071–2080. JMLR.org.

Wan,F. *et al*. (2019) NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. *Bioinformatics*, **35**, 104–111.

Wishart,D.S. *et al*. (2006) Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res*., **34**, D668–D672.

Wishart,D.S. *et al*. (2008) Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res*., **36**, D901–D906.

Xie,L. *et al*. (2012) Novel computational approaches to polypharmacology as a means to define responses to individual drugs. *Annu. Rev. Pharmacol. Toxicol*., **52**, 361–379.

Yamanishi,Y. *et al*. (2008) Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, **24**, i232–i240.

Yang,B. *et al*. (2015) Embedding entities and relations for learning and inference in knowledge bases. In: ICLR.

Chapter 12

Prediction of Kinase-Substrate Networks

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Accurate prediction of kinase-substrate networks using knowledge graphs

Vít Nováček[1,7†]*, Gavin McGauran[3], David Matallanas[3], Adrián Vallejo Blanco[3,4], Piero Conca[2], Emir Muñoz[1,2], Luca Costabello[2], Kamalesh Kanakaraj[1], Zeeshan Nawaz[1], Brian Walsh[1], Sameh K. Mohamed[1], Pierre-Yves Vandenbussche[2], Colm J. Ryan[3], Walter Kolch[3,5,6], Dirk Fey[3,6]*

1 Data Science Institute, National University of Ireland Galway, Ireland, **2** Fujitsu Ireland Ltd., Co. Dublin, Ireland, **3** Systems Biology Ireland, University College Dublin, Belfield, Dublin 4, Ireland, **4** Department of Oncology, Universidad de Navarra, Pamplona, Spain, **5** Conway Institute of Biomolecular & Biomedical Research, University College Dublin, Belfield, Dublin 4, Ireland, **6** School of Medicine, University College Dublin, Belfield, Dublin 4, Ireland, **7** Faculty of Informatics, Masaryk University, Brno, Czech Republic

†Lead author.
* novacek@fi.muni.cz (VN); dirk.fey@ucd.ie (DF)

Citation: Nováček V, McGauran G, Matallanas D, Vallejo Blanco A, Conca P, Muñoz E, et al. (2020) Accurate prediction of kinase-substrate networks using knowledge graphs. PLoS Comput Biol 16(12): e1007578. https://doi.org/10.1371/journal.pcbi.1007578

Editor: Anand R. Asthagiri, Northeastern University, UNITED STATES

Received: November 29, 2019

Accepted: August 10, 2020

Published: December 3, 2020

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1007578

Data Availability Statement: The training/testing splits for reproducing the computational experiments are available at https://doi.org/10.6084/m9.figshare.12179925.v1. In case of queries

Abstract

Phosphorylation of specific substrates by protein kinases is a key control mechanism for vital cell-fate decisions and other cellular processes. However, discovering specific kinase-substrate relationships is time-consuming and often rather serendipitous. Computational predictions alleviate these challenges, but the current approaches suffer from limitations like restricted kinome coverage and inaccuracy. They also typically utilise only local features without reflecting broader interaction context. To address these limitations, we have developed an alternative predictive model. It uses statistical relational learning on top of phosphorylation networks interpreted as knowledge graphs, a simple yet robust model for representing networked knowledge. Compared to a representative selection of six existing systems, our model has the highest kinome coverage and produces biologically valid high-confidence predictions not possible with the other tools. Specifically, we have experimentally validated predictions of previously unknown phosphorylations by the LATS1, AKT1, PKA and MST2 kinases in human. Thus, our tool is useful for focusing phosphoproteomic experiments, and facilitates the discovery of new phosphorylation reactions. Our model can be accessed publicly via an easy-to-use web interface (LinkPhinder).

Author summary

LinkPhinder is a new approach to prediction of protein signalling networks based on kinase-substrate relationships that outperforms existing approaches. Phosphorylation networks govern virtually all fundamental biochemical processes in cells, and thus have moved into the centre of interest in biology, medicine and drug development. Fundamentally different from current approaches, LinkPhinder is inherently network-based and makes use of the most recent AI developments. We represent existing phosphorylation data as knowledge graphs, a format for large-scale and robust knowledge representation.

related to reagent and resource sharing, the point of contact is Systems Biology Ireland, University College Dublin (sbiadmin@ucd.ie).

Funding: This work was supported by the CLARIFY project funded by European Commission under the grant number 875160, the TOMOE project funded by Fujitsu Laboratories Ltd., Japan and Insight Centre for Data Analytics at National University of Ireland Galway (supported by the Science Foundation Ireland grant 12/RC/2289) and Science Foundation Ireland grants 14/IA/2395 and 15/CDA/3495 to Walter Kolch and David Matallanas, respectively. The funders had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Training a link prediction model on such a structure leads to novel, biologically valid phosphorylation network predictions that cannot be made with competing tools. Thus our new conceptual approach can lead to establishing a new niche of AI applications in computational biology.

Introduction

Nearly all aspects of cell behaviour are controlled by phosphorylation events and intricate networks of kinases-substrate relationships mediating these phosphorylations [1]. Depending on the phosphorylation site, the attachment of a phosphate group can alter the activity of a substrate, its interaction with other proteins or its subcellular localization. This diversity of phosphorylation mediated processes control important cellular functions such as signal transduction, differentiation, migration, cell division and apoptosis. Dysregulation of these kinase-substrate relationships can have devastating consequences and are regularly observed in prevalent diseases, such as cancers or immune diseases. Therefore, kinases have emerged as attractive drug targets and have become the mainstay of targeted therapies with nearly fourty kinase inhibitors approved for clinical use as of 2018 [2] and over 150 in clinical trials since 2012 [3, 4].

In order to improve the design of kinase inhibitors, understand their mode of action and potential side effects, a better understanding of kinase-substrate relationships and the networks they form is necessary. With the advent of modern high-throughput mass-spectrometry based phosphoproteomics, many thousands of phosphorylation sites in substrate proteins can be identified [5]. However, large scale and reliable prediction of which kinase can phosphorylate which substrates at which sites remains challenging. High-throughput experiments are not informative in this case, because they cannot establish these detailed functional relationships, and addressing this issue in a one-by-one fashion is prohibitively expensive and time-consuming due to the large number of candidate interactions to be tested [6].

Reliable automated prediction of phosphorylation candidates is therefore much desired, because it can substantially reduce the number of possibilities that have to be tested experimentally. During the last decade, several tools for predicting phosphorylations have become available. The most widely used and recently described include: Scansite [7], GPS [8], NetPhos [9], NetPhorest [10], NetworKin [6, 10], PhosphoPredict [11]. Each of these tools, however, covers only a limited fraction of over 500 known human kinases [12], with 33, 217, 17, 178, and 6 kinases covered, accordingly. In addition to the limited coverage, existing approaches also suffer from an important conceptual limitation. Only intrinsic features of proteins (such as sequence, structure or functional annotations) are primarily used in training the predictive models. Phosphorylations, however, are inherent parts of complex interaction networks, and this type of information is largely neglected by current models.

Here, we show that predicting kinase-substrate relationships can be formulated as finding missing links in a knowledge graph (i.e. a relational, machine-readable knowledge base constructed from known phosphorylation networks). Knowledge graphs are a powerful way to organise descriptions of properties of objects and their connections [13]. However, they have not been widely used yet to analyse biological relationships. We show that using such a relational representation enables models that have superior generalisation power and precision when compared to existing approaches, lead to increased phosphoproteome coverage and produce biologically valid predictions. This can be explained by the fact that our approach fully utilises latent patterns in phosphorylation networks that are neglected by existing approaches

(e.g. long-range relational dependencies and implicit hierarchical structure). Moreover, the relational representation is not critically dependent on local features, which means our approach can make predictions even for under-researched proteins where existing approaches fail to provide results.

To test this concept, we have built a predictive model based the known phosphorylation network in PhosphoSitePlus [14] interpreted as knowledge graph. This model uses statistical relational learning to address the kinase-substrate prediction problem. We show that our model has superior predictive power based on a comparative validation trial following standard machine learning evaluatuion protocols. The model also outperforms existing tools in the total number of human kinases covered (327, nearly twice as many as the next best tool), which substantially increases the number of potential discoveries that can be made using our tool. The biological relevance of our approach is evidenced by the discovery and experimental validation of previously unknown kinase-substrate relationships for the AKT1, LATS1, PKA and MST2 kinases.

Results

The concept of our approach in comparison with related existing techniques is illustrated in Fig 1 and details are given in the Materials and Methods section. Where existing tools use

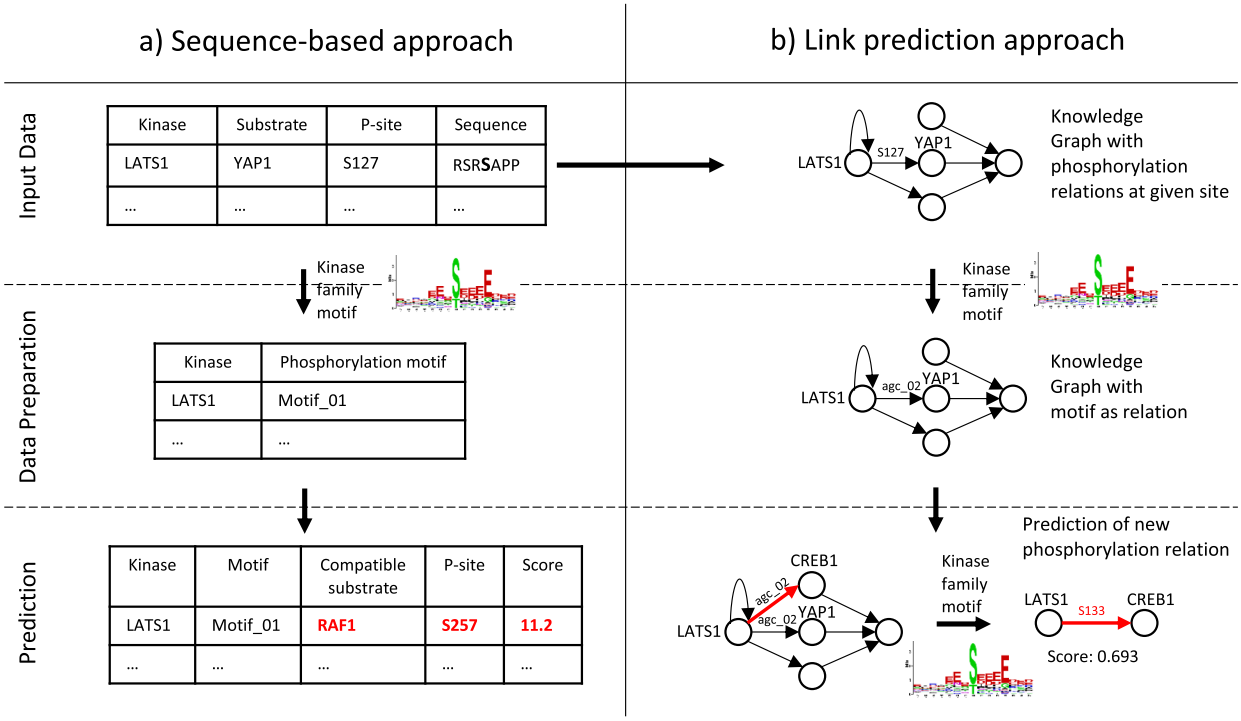


Fig 1. a) Sequence-based approaches aim to identify linear amino acid motifs that are phosphorylated by certain kinases. This is done based on known motif preferences of kinases, their groups or families. Each site and substrate is examined in isolation. Only limited numbers of well-studied kinases can typically be associated with substrates this way, and network context is largely ignored in such predictions. b) The LinkPhinder approach aims at learning regular patterns in a knowledge graph that represents the known kinase-substrate links as motif-based abstractions of the associated consensus sites. Based on the global, latent properties of the knowledge graph, the system can predict unknown, site-specific interactions between any kinase and substrate present in the input data.

https://doi.org/10.1371/journal.pcbi.1007578.g001

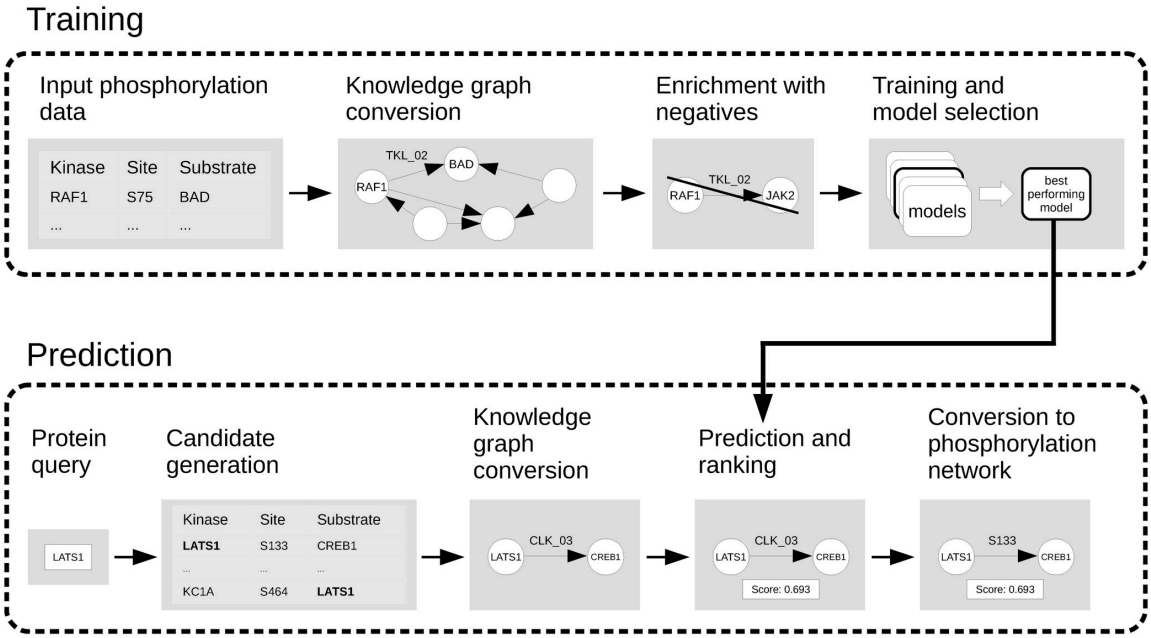


Fig 2. The model is first trained on phosphorylation network data that has been converted to a knowledge graph representation. Such a representation can be readily processed by link prediction algorithms (contrary to the original phosphorylation data). In the training stage, an optimal combination of model parameters is found and computationally validated. The optimal model is then trained on full phosphorylation network data and used for providing probabilistic ranking scores for all possible predictions that can be made using the input. Finally, reverse conversion technique is applied to the computed predictions to present them to users as residue-specific kinase-substrate relationships.

https://doi.org/10.1371/journal.pcbi.1007578.g002

primarily local features based on sequence of particular proteins (left-hand side), our approach also considers the network information in training the model. Our predicitons are effectively based on explicit and implicit functional links between kinases and substrates represented as knowledge graphs. Briefly, we used a PhosphoSitePlus, a highly curated database of experimentally confirmed phosphorylation sites [14], to construct a knowledge graph where links between kinases and substrate corresponded to shared characteristics of kinase consensus sites. This knowledge graph represented a training set of known kinase-substrate relationships that was used for learning our predictive model (effectively, a multi-variate probability distribution function fitted to the input data). This model can consequently be used for predicting unknown kinase-substrate relationships with high coverage and precision.

The workflow of our methodology is illustrated in Fig 2. Details on training the computational model and the data used are provided in the Materials and Methods section. The main steps of constructing the LinkPhinder model are: (i) Generation of a phosphorylation network based on kinase-substrate pairs reported in PhosphoSitePlus (albeit any other database could be used). (ii) Inference of phosphorylation site motifs for kinase families based on quantifying the contribution of each amino acid in a set of consensus sequences to the likelihood that this sequence is phosphorylated. (iii) Conversion of the phosphorylation network into a knowledge graph using the phosphorylation motifs as generalised links to connect compatible kinase-substrate pairs while preserving the site information. (iv) Learning of new links based on both explicit and latent relationships in the input network data. The learning process is supervised and thus requires negative kinase-substrate relationships. These were generated using random

perturbations of the positive examples. (v) Selection of the best performing model. (vi) Generation of all possible kinase-substrate combinations using the input data and using our trained model for computing ranking scores for each kinase-substrate link. This ranking effectively allows to select most likely, previously unknown phosphorylations a kinase or substrate of interest can be involved in. (vii) Conversion of kinase-substrate links back to phosphorylation site sequences that provide the user exact information about the amino acid sequence phosphorylated by the given kinase in the substrate.

In the following, we present the performance of the LinkPhinder model. First, we benchmark LinkPhinder against six commonly used existing tools. Then, we present results of biological validation experiments focused on selected kinases of clinical relevance and their substrates. Finally, we introduce a web interface that allows the scientific community convenient access to LinkPhinder.

Computational validation of LinkPhinder shows superior precision and kinome coverage

While the LinkPhinder model learns its parameters from the input data automatically, the optimal model configurations (also called hyperparameters) cannot be inferred that way and need to be determined empirically. In order to find these hyperparameters that optimise the performance of our model, we used the knowledge graph generated from PhosphoSitePlus [14] and evaluated several link prediction techniques across a range of their possible settings as described in the Materials and Methods section. The best method was ComplEx [15], which can handle large networks and generalises well for anti-symmetric relationships (of which the directed kinase-substrate links are an example). The optimal hyperparameters were identified by a grid search [16] and the best performing model was selected for the experiments described in this section. This model was trained on the entire network of phosphorylations contained in PhosphoSitePlus to produce unknown phosphorylation candidates for laboratory validation experiments described in the following sections.

The trained model can predict the likelihood of phosphorylation reactions that exist in the training dataset but have not been observed yet. In principle, any phosphorylation dataset can be used, but we chose PhosphoSitePlus because it is widely considered the most comprehensive and accurate dataset on known phosphorylations in many different organisms including human [17].

The computational validation experiments compared our approach to a selection of six existing and commonly used phoshorylation prediction techniques: Scansite [7], GPS [8], NetPhos [9], NetPhorest [10], NetworKin [6, 10] and PhosphoPredict [11]. For running this benchmarking trial, we generated 100 random train/test splits (90% train, 10% test) of true positives from the subset of PhosphoSitePlus human phosphorylations (i.e., *kinase-phosphorylation site-substrate* triples). A pool of negative statements was generated by random associations between all human kinases and *(phosphorylation site*, *substrate)* pairs available in PhosphoSitePlus. This pool was used for sampling as many negatives as there were positives in each train/test split. For each of the 100 splits, we trained our model on the 90% of the data and validated it on the unseen 10%. For the existing techniques, we generated all their predictions relevant to the proteins in the PhosphoSitePlus dataset and assessed them using the test splits.

Note that to make sure the presented relative differences between the methods are not merely due to the specific way we prepared the benchmarking data, we have also experimented with different train-test split and positive-negative ratios. The relative performances of the compared methods have not, however, changed from what is presented here. More

Table 1. Comparative validation results. AU-PR, AU-ROC refer to the area under the precision-recall and ROC curve, respectively. These metrics are widely used for validating predictive models based on ranking across their whole operating range [18]. P@K refers to the precision at K metric that gives the ratio of true positive statements ranked among top K results (e.g., P@10 refers to precision at 10; precision at 10 equal to 0.9 would mean that the corresponding tool typically returns 9 true positives among the top 10 results).

Model	AU-PR	AU-ROC	P@10	P@50
GPS	0.741±0.011	0.731±0.011	0.862±0.108	0.857±0.049
NetworKin	0.688±0.010	0.619±0.011	0.981±0.046	0.961±0.027
NetPhorest	0.650±0.012	0.598±0.011	0.905±0.091	0.905±0.041
Scansite	0.605±0.012	0.573±0.013	0.727±0.143	0.777±0.059
Phosphopredict	0.504±0.011	0.503±0.168	0.539±0.168	0.523±0.081
Netphos	0.612±0.012	0.563±0.013	0.865±0.105	0.863±0.048
LinkPhinder	**0.973±0.004**	**0.968±0.004**	**0.994±0.024**	**0.993±0.012**

https://doi.org/10.1371/journal.pcbi.1007578.t001

information on the benchmarking methodology and results corresponding to the different ratios can be found in the Materials and Methods section.

The results are summarised in Table 1. The corresponding charts with the PR and ROC curves are given in Fig 3. The table presents means and standard deviations for each of the performance metrics computed across the 100 experimental runs with random train/test splits. Our model outperforms the existing techniques in all validated metrics, often by rather large margins. The narrower confidence margins of LinkPhinder results (about 1.8-42 times less than for the related works) mean that even if the experiment was done just once, it is still very likely the relative performance differences between the tools would be the same as presented in the table.

To gain additional insights into the presented results, we analysed to what extent each tool covered the phosphorylations in the test splits. The coverage is an important factor influencing the results since we assign zero scores to phosphorylations which the systems are not able to process (i.e., those for which no ranking scores can be produced). Therefore, a tool that does not produce scores for negative examples will have these annotated with zeros automatically and thus they will be at the bottom of their ranking lists. This is a possible advantage over tools that do produce scores for such negative phosphorylations, as any positive scores can only move the negatives up in the ranking, resulting in more false positive assignments.

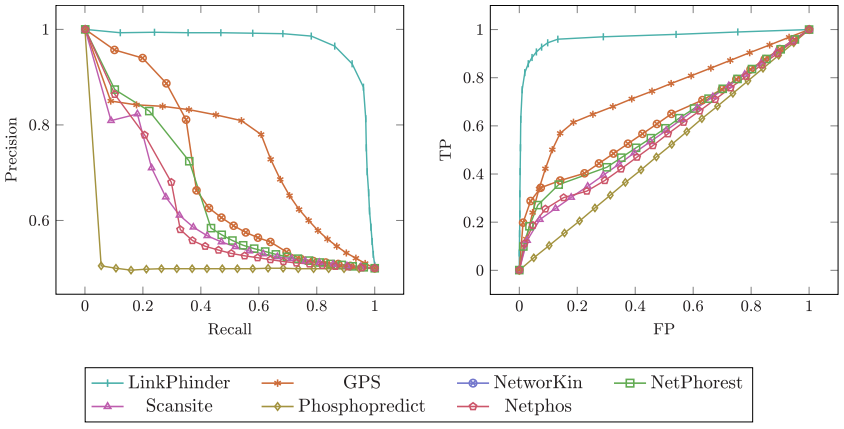


Fig 3. The average precision-recall and ROC curves as per the experimental results reported in Table 1 (left and right part of the figure, respectively).

https://doi.org/10.1371/journal.pcbi.1007578.g003

Table 2. Coverage of the tools in per cents. Total, positive and negative coverage is given in the first three columns with data, respectively. The last column gives the percentage of missed negatives (i.e., negatives that are assigned the default zero score).

Model	Tot. coverage	Pos. coverage	Neg. coverage	Missed neg.
GPS	38.6±1.0%	60.6±1.5%	16.6±1.2%	83.4%
NetworKin	34.1±1.0%	40.6±1.5%	27.7±1.3%	72.3%
NetPhorest	34.1±1.0%	40.6±1.5%	27.7±1.3%	72.3%
Scansite	10.8±0.6%	18.0±1.1%	3.6±0.5%	99.5%
Phosphopredict	1.1±0.2%	1.3±0.3%	1.0±0.3%	99.0%
Netphos	28.8±1.1%	33.0±1.7%	24.7±1.2%	75.3%
LinkPhinder	64.2±0.8%	97.0±0.5%	31.4±1.5%	68.6%

https://doi.org/10.1371/journal.pcbi.1007578.t002

The coverage of the different tools, i.e., their ability to make predictions for proteins represented in PhosphoSitePlus, is given in Table 2. This table shows that our model has the highest coverage of the test splits, especially when it comes to positive statements. However, the percentage of missed negative phosphorylations is still the lowest for our model, which means that the other tools are not disadvantaged by the setup of this benchmarking test.

Thus, LinkPhinder outperforms existing popular tools in terms of sensitivity and specificity (by means of the area-under-the-curve and precision metrics used), but also in terms of the number of predictions it can make. Importantly, LinkPhinder also covers a larger fraction of the human kinome than the other tools. Comprehensive visualisation of this fact is given in Figs 4 and 5.

The results diplayed in Fig 4 clearly illustrate the superior potential of LinkPhinder for discovering new phosphorylations relevant to under-researched kinases, which is currently considered one of the most pressing challenges in phosphoproteomics [17]. This is complemented by Fig 5 that shows, among other things, the relative advantages LinkPhinder presents in numbers of kinase-substrate and site-specific kinase-substrate interactions for which it can provide predictions (having the second best and best coverage, respectively). While higher coverage of possible predictions may not mean much on its own, we believe it is a reassuring sign when combined with the presented data on the superior performance of LinkPhinder in terms of the quality of the prediction scores it can associate with such an unprecedented range of kinase-substrate interaction candidates.

To provide a complementary computational validation using a dataset independent of the one we trained our model on, we have used a very recent data on site-specific interactions of 103 human kinases with their substrates in cancer cells [19]. Table 3 presents the performance of LinkPhinder and the six related tools when using this data for validation in the same fashion as in the previously reported computational experiment.

While the performance of all tools is substantially weaker than when using the PhosphoSitePlus benchmark (i.e. only slightly above the random baseline for both area-under-the-curve metrics), LinkPhinder is still the best in three out of four metrics, and close second in the remaining one. The overall poor performance can be attributed to a relatively small coverage of the [19] gold standard exhibited by most tools when compared to the PhosphoSitePlus [14] one (detailed overlap statistics are provided in section Training of the LinkPhinder Model). In such a situation, the relative ranks of the true positives among the rather large sets of all candidate predictions provided by the tools would tend to fluctuate quite widely, which can provide at least partial explanation of the differences in the predominantly ranking-based metrics between the two benchmark datasets. Another part of the explanation may be the fact that while [14] covers a broad range of cell lines and tissues, [19] only covers three cell lines. Tools that are presumably trained using existing knowledge covering as many cell/tissue types as

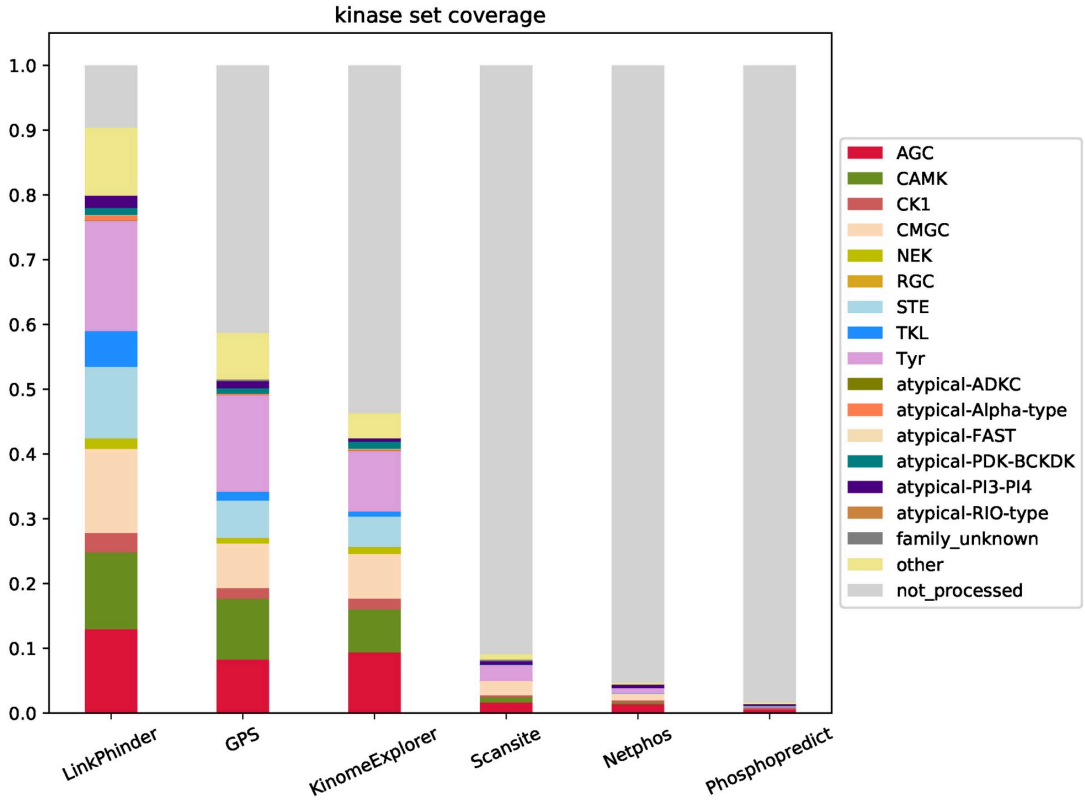


Fig 4. Coverage of the human kinome and kinase families as per PhosphoSitePlus. The "not_processed" category reflects the number of kinases for which a tool cannot produce any predictions. Note that NetPhorest and NetworKin only differ in scores assigned to predictions, while the set of phosphorylations they can produce scores for is identical. Therefore, they are grouped under a common KinomeExplorer [10] in the plot.

https://doi.org/10.1371/journal.pcbi.1007578.g004

possible (like ours) may thus be expected to perform relatively poorly on a dataset that specifically covers a limited number of cell lines and corresponding kinases. That being said, this issue may point to an interesting research avenue to be addressed by future studies in this area that would further investigate the cell line-specific performance of models for predicting kinase-substrate interactions.

Targeted experiments confirm two previously unknown phosphorylation sites targeted by LATS1 and AKT1

The rich dataset of over 11 million candidate predictions was assessed regarding its potential for discovering new phosphorylation sites for kinases that have biomedical relevance, such as AKT and LATS1. Both kinases regulate cell survival, growth, proliferation, and are frequently altered in cancer [20–22]. AKT has now become a leading drug target in cancer research, but the long term application of AKT inhibitors is still considered problematic because of AKT's essential roles in regulating glucose homeostasis [23]. The situation is similar with LATS1. Originally described as tumor suppressor, it also can have growth promoting roles [22]. In

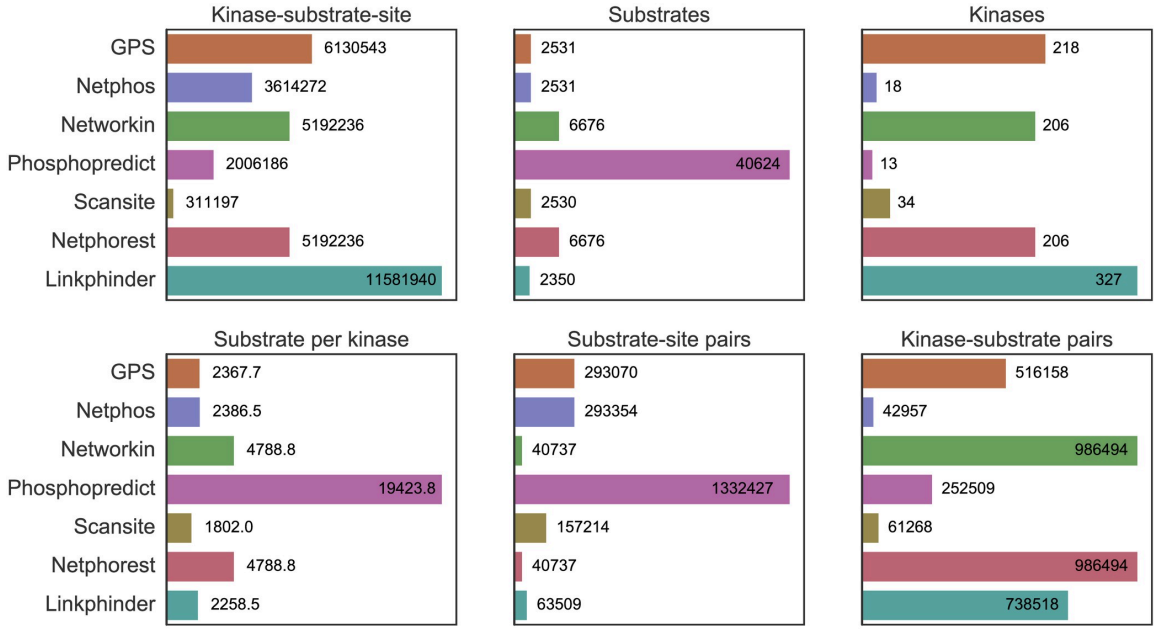


Fig 5. Complementary statistics of the coverage of different systems in terms of number of kinases, substrates, sites per substrate, etc.

https://doi.org/10.1371/journal.pcbi.1007578.g005

order to resolve unwanted from desired effects a better and more comprehensive understanding of the substrate spectrum of these kinases is needed.

To this end, we have extracted high-stringency predictions for the LATS1 and AKT1 kinases. By visual inspection of this list, we were able immediately pinpoint several known and promising substrates of these kinases. YAP1 for example is the best characterized substrate of LATS1, and our tool predicted that LATS1 would phosphorylate YAP at serine 127, which is the best studied phosphorylation site that contributes to YAP inactivation [21]. Additionally, the list of the top 200 predictions for AKT contained eight bona-fide AKT substrates [24], for which several phosphorylation reactions were predicted. This means that our tool generated interesting, biologically relevant predictions.

Therefore, we decided to validate some of the new substrates experimentally. In order to select the most promising candidates we selected three proteins that are part of the wider LATS1 signaling network and for which antibodies were commercially available. To

Table 3. Complementary computational validation of LinkPhinder using the recent dataset published in [19] as a benchmark independent of the primary training dataset (i.e. PhosphoSitePlus [14]).

Model	AUPR	AUROC	P@10	P@50
GPS	0.518 ± 0.008	0.509 ± 0.010	0.675 ± 0.171	0.663 ± 0.059
NetworKin	0.519 ± 0.008	0.511 ± 0.010	0.682 ± 0.132	0.616 ± 0.062
NetPhorest	0.519 ± 0.007	0.510 ± 0.008	**0.731 ± 0.135**	0.659 ± 0.056
Scansite	0.504 ± 0.008	0.502 ± 0.009	0.561 ± 0.170	0.563 ± 0.066
Phosphopredict	0.502 ± 0.008	0.502 ± 0.009	0.519 ± 0.137	0.507 ± 0.069
Netphos	0.508 ± 0.009	0.505 ± 0.009	0.551 ± 0.149	0.554 ± 0.074
LinkPhinder	**0.540 ± 0.009**	**0.532 ± 0.010**	0.713 ± 0.153	**0.671 ± 0.061**

https://doi.org/10.1371/journal.pcbi.1007578.t003

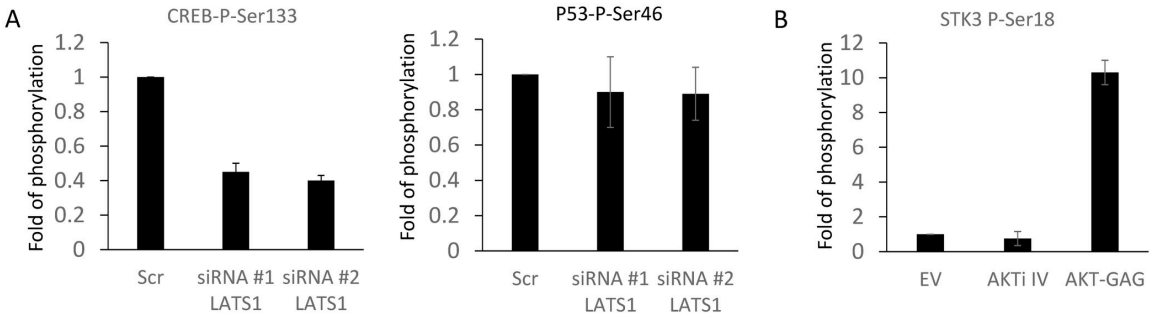


Fig 6. Experimental validation of model predictions. A) HEK293 cells were transfected with non targeted siRNA (Scr) of the indicated siRNA against LATS1. Phosphorylation of CREB or p53 was measured using specific antibodies and normalised to the level of expression of the corresponding proteins. The graph shows the fold change of the phosphorylation of the specific residues with respect to the Scr control. B) HEK293 were transfected with empty vector (EV) or GAG-AKT or treated with AKTi IV (10μM) for 1 hour. Phosphorylated proteins were immunoprecipitated using an anti-AKT antibody and the immunoprecipitates were blotted with anti-MST2. The bars show the fold change with respect to the control. The experiments were repeated at least 2 times. Error bars represent standard variations.

https://doi.org/10.1371/journal.pcbi.1007578.g006

experimentally validate predicted LATS1 substrates we used the following strategy. HEK293 cells were transfected with two specific siRNAs against LATS1 in order to downregulate LATS1 protein levels (knockdown). Following this LATS1 knockdown, we would expect to see a decrease in the phosphorylation of LATS1 substrates. For confirmation, we used a positive control where we measured the phosphorylation of the known LATS1 substrate YAP1-S127, and indeed observed a decrease in YAP1-S127 phosphorylation in total cell lysates (c.f. S1A Fig). This control experiment demonstrated that our LATS1 knockdown works as expected and can be used to confirm potential LATS1 substrates.

One of the proteins that we selected for validation was CREB which is transcription factor that is regulated by phosphorylation [25]. This transcription factor is one of the best characterized effectors of the MAPK ad PKA pathways. Evidence in the literature indicates that CREB modulates important LATS1 pathway functions by direct interaction with YAP1 and regulation of transcription [26]. Our tool predicted serine 133 of CREB (CREB-S133) as a putative substrate of LATS1. To confirm this, we used a specific antibody against CREB-S133, and saw that downregulation of LATS1 resulted in about 50% decrease of CREB-S133 phosphorylation (Fig 6, S1B Fig). This result clearly indicated that CREB is a physiological LATS1 substrate and highlights the potential of our tool to identify previously unknown kinase substrates.

In the case of AKT we decided to monitor putative substrates by manipulating the level of AKT activation using two strategies. Firstly, we inhibited endogenous AKT activity by using the specific chemical inhibitor AKTi IV. Secondly, we increased AKT activity by transfecting a kinase hyperactive form of AKT with gag-AKT [27]. One of the predicted AKT substrates is MST2 (MST2), which is an important protein kinase in the Hippo pathway that can phosphorylate and activate LATS1 [21], and which according to the prediction should be phosphorylated at serine 18. Unfortunately, no commercially available antibody exists that could measure this phosphorylation site. Therefore, we employed an indirect approach to validate this prediction. We used an antibody that specifically binds to phosphorylated AKT substrates, which will immunoprecipitate (IP) all the proteins that are phosphorylated by AKT. Next, we blotted this IP using a specific antibody against MST2 (S1C Fig). The inhibition of AKT resulted in a slight, but consistent decrease of MST2 phosphorylation (0.75 fold), while expression of active AKT resulted in a 10 fold increase (Fig 6). The results validate the prediction that AKT1 phosphorylates MST2.

After confirming the new site-specific kinase-substrate relationships involving the LATS1 and AKT1 kinases as reported above, we found out that none of the six existing systems used in our comparative validation could predict these phosphorylations on high stringency settings. This further demonstrates the unique power of LinkPhinder in the context of computational phosphorylation prediction.

Mass spectrometry experiments confirm seven previously unknown phosphorylations by LATS1

To extend the targeted validation experiments we cross-referenced our predictions with high-throughput phospho-proteomic data on the LATS1 interactome (Fig 7A). This strategy is based on the fact that in order to phosphorylate a substrate, the kinase needs to bind to it. Based on our previous observations, kinases tend to be associated with their substrates in complexes that can be isolated and characterized by mass spectrometry [28]. Thus, by isolating all proteins that are bound to LATS1 by immunoprecipitation (IP) and analyzing this interactome using mass-spectrometry based proteomics, we should be able to identify a large number of LATS1 phosphorylation targets.

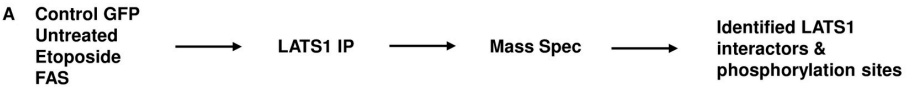
Using this approach, we obtained phospho-proteomic data on the LATS1 interactome from cells treated with two proapoptotic signals: FAS and etoposide, which both activate LATS kinase activity [29]. To identify the LATS1 interactome we transiently expressed GFP-LATS1 in HeLa cells, immunoprecipitated GFP-LATS1 with anti-GFP antibodies and identified the associated proteins using mass-spectrometry. Unspecific binding proteins were discarded by comparing with the control GFP IP. This approach identified seven proteins that were bound to LATS1 and phosphorylated on at least one residue (Fig 7B, S1 Data). These proteins are potential LATS1 substrates, but it is important to note that not all of these phosphoproteins are LATS1 targets, because LATS1 also binds to proteins that are phosphorylated by other kinases. Therefore, we cross-referenced this list of phosphorylated LATS1 interactors with our list of predicted LATS1 phosphorylation targets from LinkPhinder (Fig 7C, S1 Data). This confirmed 7 previously unknown phosphorylations on three substrates; five residues were phosphorylated on LATS1 (S613, S278, S464, S181, T17), one on MAP4 (S5), and one on ZMYM2 (T1253). Importantly, stimulation with FAS caused reduction of the phosphorylation of phosphorylation of LAST1-S464, MAP4-S5 and ZMYM2-T1253 (Fig 7D) indicating that regulation of these residues are specifpicaly regulated by the death receptor pro-apoptotic signal.

After confirming the new site-specific kinase-substrate relationships involving the LATS1 kinase as reported above, we searched for these in the prediction data provided by the existing tools. However, on high stringency settings, only GPS could predict one of the seven predictions made by us (LATS1-S464, GPS Score = 8.8, S2 Data). This further demonstrates the enhanced prediction capabilities of LinkPhinder.

Kinase assays based on mass spectrometry confirm the sensitivity of LinkPhinder

One of the challenges to show the sensitivity of our tool and how it compares with existing tools is the lack of experimental methods to validate substrates systematically on a large scale. In order to further validate LinkPhinder predictions we decided to extend our validation experiments and use an in vitro kinase assay system that can identify multiple substrates for a given kinase. This method is based on the purification of proteins that have been phosphorylated by the kinase using an ATP analogue modified with a biotin group [30]. Briefly, all the endogenous kinases are inhibited with FSBA, a pan-kinase inhibitor, and the recombinant kinase is added to protein lysates together with ATP-biotin. ATP-biotin allows the purification

A



B

Gene	Site	Protein	Modified sequence	Phospho (STY) Probabilities
ACSS1	T 4	Q6ZV30	_(ac)LKT(ph)FLFLEK_	LKT(1)FLFLEK
AFF4	S 180	Q9UHB7	_SSS(ph)PGKPQAVSSLNSSHSR_	S(0.003)S(0.003)S(0.993)PGKPQAVSSLNSSHSR
KTN1	S 75	Q86UP2	_EIQNGNLHES(ph)DSESVPR_	EIQNGNLHES(0.944)DS(0.053)ES(0.003)VPR
LATS1	**S 613**	**O95835**	**_KQITTS(ph)PITVR_**	**KQITT(0.003)S(0.997)PITVR**
LATS1	**S 278**	**O95835**	**_RYS(ph)GNMEYVISR_**	**RYS(1)GNMEYVISR**
LATS1	**S 464**	**O95835**	**_SNS(ph)FNNPLGNR_**	**SNS(1)FNNPLGNR**
LATS1	S 216	O95835	_HGPPLGESVAYHSESPNSQTDVGRPLS(ph)GSGISAFVQAHPSNGQR_	HGPPLGESVAYHSES(0.007)PNS(0.032)QT(0.032)DVGR PLS(0.645)GS(0.142)GIS(0.142)AFVQAHPSNGQR
LATS1	**S 181**	**O95835**	**_QSWKGS(ph)KESLVPQR_**	**QS(0.001)WKGS(0.976)KES(0.023)LVPQR**
LATS1	T 17	O95835	_QMRPKT(ph)FPASNYTVSSR_	QMRPKT(1)FPASNYTVSSR
MAP4	S 375	E7EVA0	_ETERAS(ph)PIKMDLAPSK_	ET(0.467)ERAS(0.533)PIKMDLAPSK
MAP4	**S 5**	**E7EVA0**	**_ADLS(ph)LADALTEPSPDIEGEIKR_**	**ADLS(0.997)LADALT(0.003)EPSPDIEGEIKR**
MAP4	S 297	E7EVA0	_DMES(ph)PTKLDVTLAK_	DMES(1)PTKLDVTLAK
SEC61B	S 13	P60468	_PGPTPSGTNVGS(ph)SGRSPSK_	PGPTPSGTNVGS(0.495)S(0.284)GRS(0.165)PS(0.055)K
SEC61B	S 14	P60468	_PGPTPSGTNVGSS(ph)GRSPSK_	PGPTPSGTNVGS(0.035)S(0.665)GRS(0.298)PS(0.003)K
SEC61B	S 17	P60468	_PGPTPSGTNVGSSGRS(ph)PSK_	PGPTPSGTNVGS(0.11)S(0.11)GRS(0.67)PS(0.11)K
ZMYM2*	**T 1253**	**Q9UBW7**	**_KNPLT(ph)M(ox)ENK_**	**KNPLT(1)MENK**

C **LinkPhinder predictions** for identified MST1 interactors

Kinase	Site	Substrate	Score
LATS1	S613	LATS1	0.751187027
LATS1	S278	LATS1	0.751187027
LATS1	S464	LATS1	0.751187027
LATS1	S181	LATS1	0.751187027

Kinase	Site	Substrate	Score
LATS1	T17	LATS1	0.751187027
LATS1	S5	MAP4	0.617851198
LATS1	T1253	ZMYM2	0.648494661

D

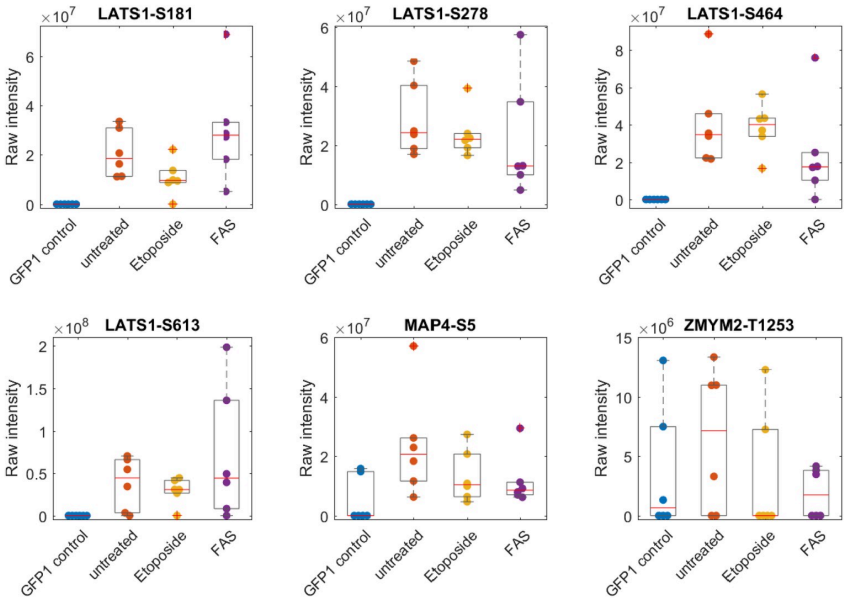


Fig 7. Mass-spectrometry validation of a subset of LinkPhinder predicted phosphorylations. A) Overview of the experimental design. B) Mass-spectrometry result: Specific LATS1 interactors and their phopshorylations. Bold rows indicate phosphorylation that were predicted by LinkPhinder. (*There is a risk that ZMYM2 binding might be unspecific. Some samples show high intensities in the GFP1 control, see panel D.) C) LinkPhinder predictions for the results in panel B. D) Mass-spec raw intensity values (dots) of the detected phosphorylation sites in GFP-LATS1 associated proteins under the indicated conditions (n = 6 replicates), and corresponding box plots indicating median (red line), upper and lower quartile (grey box), whiskers (most extreme values not defined as outliers), and outliers (plus marks) defined as values outside 1.5 times the interquartile range.

https://doi.org/10.1371/journal.pcbi.1007578.g007

Table 4. Sensitivity (*S*) of LinkPhinder substrate predictions per each of the kinase assay.

Kinase	Predicted substrate gene names	S
PKA	PKA, TGM2, PSMC5, PA2G4	0.57
MST2	MST2, MOB1A, NUP153, SNAPIN	0.13
LATS1	LATS1, RHOA, VCP, SNAP25, CCT2, HNRNPK, RPS6, HSP90AA1	0.17

https://doi.org/10.1371/journal.pcbi.1007578.t004

of phosphorylated proteins using streptavidin and the subsequent identification of these proteins as substrates using mass-spectrometry. In order to test the method we replicated the Pflum study using PKA as kinase in HeLA cells [30] in a different cell line (HEK-293). From a total of 834 identified proteins, 34 proteins were identified as putative substrates of PKA by comparing with the PKA deficient control samples (Table 4, and supplementary experimental information). Five of these proteins were previously identified in the Pflum study, and 11 of them were isoforms or proteins of the same protein family. We also identified 18 new putative substrates. These additional 18 proteins that did not occur in the Pflum study using HeLa cells may be cell-specific substrates in the here used HEK-293 cells. The overlap in the results clearly indicated that the global kinase assay is an additional tool that could be used to validate our predictions.

We then extended our validation experiments using this global kinase assay to LATS1 and MST2. First, we used LATS1 as kinase. We identified 240 putative LATS1 substrates from a total of 1397 identified proteins by comparing to the LATS1 deficient controls (Table 4, and detailed description in Section on Experimental Model and Subject Details). Secondly, we used MST2 as kinase. MST2 is another core kinases of the MST2/Hippo pathway with poorly characterised substrates. Our results identified 211 proteins as putative MST2 substrates. Strengthening our confidence into the validity of these results, five of the identified putative substrates have been described as MST2 interactors previously.

The experimentally validated PKA, MST2 and LATS1 substrate predictions made by LinkPhinder are listed in Table 4. The table also provides the sensitivity (*S*) of these predictions in the context of each specific kinase assay. The sensititivy was computed as

$$S = \frac{SUBS_{predicted}}{SUBS_{total}},$$

where $SUBS_{predicted}$ is the number of substrates for which LinkPhinder provided at least one site-specific phosphorylation prediction with a score above the high confidence threshold, and $SUBS_{total}$ is the number of substrates that were identified in the kinase assay and that are also present in the PhosphoSitePlus knowledge graph. Identified substrate proteins that were not in the knowledge graph were excluded for this analysis, because no predictions can be generated for those proteins.

The sensitivity of the PKA predictions was 0.57, which we consider a good result given that they were validated in an unbiased approach that has inherent technical limitations. For the poorly characterised MST2 and LATS1 kinases the sensitivities were lower, 0.13 and 0.17 respectively. It must be noted that generating predictions for MST2 and LATS1 is challenging because only a few substrates have been described experimentally, and most of the existing predictions tools could not generate predictions for MST2 and LATS1. Together these results indicate that LinkPhinder can be used to predict kinase-substrates interactions for poorly characterised kinases.

Finally, we wanted to benchmark LinkPhinder performance against the exsisting tools. However, we found this was not an easy task. Comparing LinkPhinder with existing tools

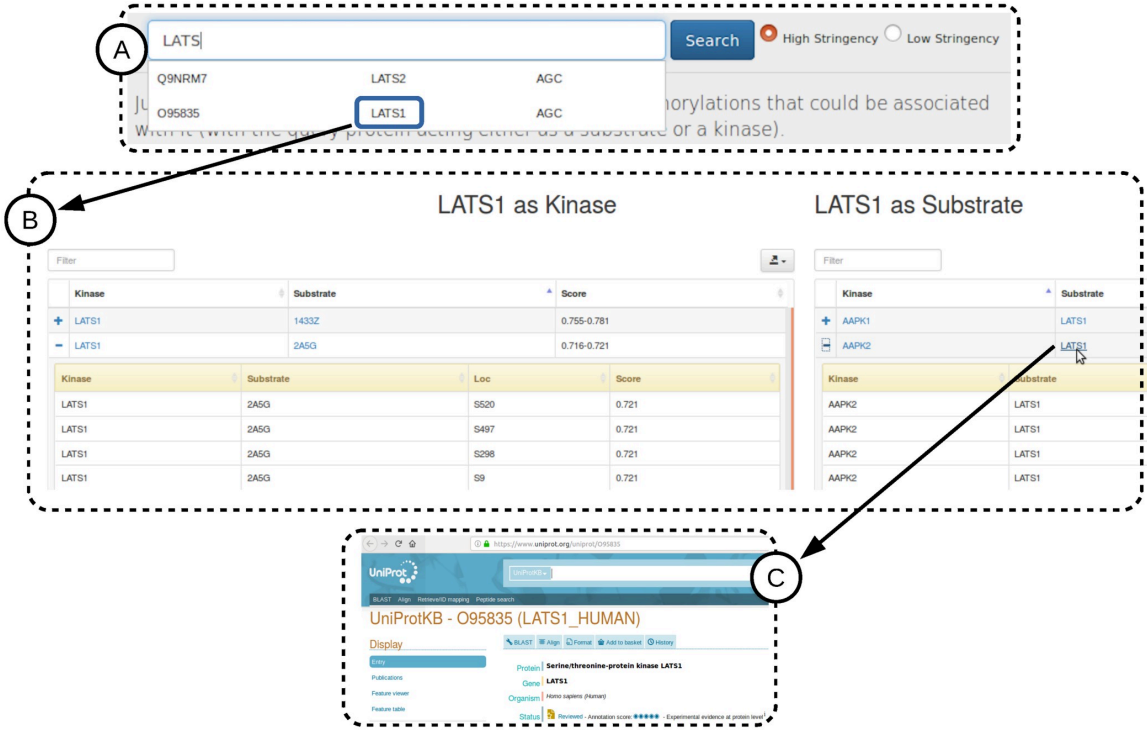


Fig 8. The LinkPhinder web interface. Shown is a typical search and browse interaction.

https://doi.org/10.1371/journal.pcbi.1007578.g008

using the results of these experiments is not as straightforward as in the cases reported before. The main reasons are conceptually different methods for determining the decision threshold employed by each of the tools. This does not allow for direct comparisons in terms of sensitivity as defined above. However, one high-level observation can be made: Only GPS matches the coverage of LinkPhinder as it can produce predictions for all the three kinases we assayed. NetworKin and NetPhorest cannot compute any predictions for LATS1, NetPhos and Phosphopredict only cover PKA, and Scansite covers none of the assayed kinases.

LinkPhinder web interface

In order to facilitate usage of LinkPhinder by the community we have developed an online interface available at https://LinkPhinder.insight-centre.org/.

A typical interaction with LinkPhinder is depicted in Fig 8. The corresponding instruction video is available in the *About* tab of the tool's web page. Briefly, the protein of interest can be entered into a search box with auto-completion (box A). Gene names and UniProt accession numbers are supported. The search is performed for high-stringency statements by default. However, all predicted statements can be searched as well (*cf.* the radio buttons in A). The query protein is evaluated by the system in two different ways, as a kinase and as a substrate and each type of predictions can be browsed independently (box B). The results can be filtered, and the predicted kinase-substrate pairs can be expanded to see the list of corresponding phosphorylation sites and prediction scores. Export of the predictions into a CSV file is also

possible. Further, users can easily access contextual information from a comprehensive protein database (UniProt) by clicking on the proteins in the results (box C).

Discussion

In this work, we have overcome several limitations of the current phosphorylation prediction tools by representing phosphorylation networks as knowledge graphs. Knowledge graphs are a relatively new approach to representing relational knowledge in the Machine Learning, Artificial Intelligence and Semantic Web communities. They have quickly gained popularity for two main reasons. First, they can represent diverse types of knowledge in a simple format. Secondly, they are amenable to robust techniques of statistical relational machine learning, that can for example be used to discover new facts. The discovery naturally makes use of the entire structure of the knowledge graph (i.e. latent features and long range, implicit relationships instead of just local, explicit features). This makes the representation very useful in domains where complex network dependencies are critical. Kinase-substrate relationships are a good example of such a domain. Our results show that knowledge graphs enabled phosphorylation predictions that were not possible with existing tools that are primarily based on local features.

In particular, we have shown that phosphorylation networks can be meaningfully captured by knowledge graphs with kinases and substrate entities linked by relationships based on phosphorylation site motifs. Therewith, modern link prediction methods can be used to predict novel phosphorylation reactions and estimate their probability based on the entire network context. The resulting predictive model allows for making predictions about any protein present in the input data. This is a substantial advantage when compared with the existing tools. These tools typically focus on substrates as initial queries and include only a limited number of kinases. LinkPhinder not only covers a much broader range of possible kinase-substrate relationships than existing tools, but also shows very high generalization power and desirable ranking properties not exhibited by other, currently gold standard approaches. This aspect has been validated in experiments showing that our tool can generate numerous biologically valid predictions. Crucially, these predictions were not possible with a representative range of state-of-the-art tools (Scansite [7], GPS [8], NetPhos [9], NetPhorest [10], NetworKin [6, 10], PhosphoPredict [11]), demonstrating the utility of our tool.

More specifically, none of the LATS1 and AKT1 discoveries validated in targeted experiments were predicted with four out of six related tools starting with LATS1 or AKT1 as kinase queries. Only GPS and PhosphoPredict support such queries, but for less than 66.4% and 1.8% of the kinases covered by LinkPhinder, respectively. Furthermore, querying for the substrates directly did not predict any of the validated discoveries using any of the existing tools using their high stringency settings (if applicable; if controlling the stringency was not offered by a particular tool, we used all predictions made by the given tool). On medium stringency, the GPS tool could identify one prediction; the CREB1 phosphorylation by LATS1. On low stringency, the NetPhosK tool could also identify one prediction; the MST2 phosphorylation by AKT1. No existing tool could identify both predictions. The LATS1 predictions validated by the mass spectrometry experiments were not be predicted by any of the existing tools but one. Specifically, the GPS tool could predict one out of the seven predictions we made (LATS1 auto-phosphorylation at S464) on high stringency (and no further ones on lower stringencies). The other five tools could not identify any of our validated predictions. When cross-referencing the list of LATS1 predictions from other tools with our predictions, no additional predictions were made, demonstrating that our tool has the best coverage. Together, these results clearly illustrate the advantages of our tool.

Experimental validation using the global PKA, MST2 and LATS1 kinase assays showed promising results in terms of LinkPhinder's sensitivity for identifying new substrates. Direct comparison with the existing tools was not possible due to disparate methods employed by each tool in determining their decision/high-stringency threshold. However, the results reconfirmed one significant benefit of LinkPhinder. We were able to produce substrate predictions for all three kinases studied, which was not possible with five of the six existing tools, with the exception of GPS, again demonstrating that LinkPhinder's increased kinase covarage is an important contribution.

To build on the work presented here, we intend to incorporate more contextual data (e.g., relevant protein interactions from STRING or pathway data from Reactome) to see whether they can bring new and/or more accurate predictions pertinent to clinically relevant pathways. We also want to develop predictive models that would utilize the biology of phosphorylation directly in the training process and not only in the knowledge graph conversion and negative example generation. As demonstrated, incorporating more network context and biological knowledge into the prediction process has great potential to further increase the coverage, predictive power, and usefulness of the resulting tools.

Another research direction to explore in future is the applicability of our predictive model to improving the accuracy and scope of methods for predicting downstream effects of kinase signalling or the kinase activity profiles. An example of such method that could benefit from our results is described in [31]. We believe follow-up experiments combining focused phosphoproteomics studies like this with our model will further demonstrate the practical relevance of the work presented here.

Materials and methods

Computational model and validation details

Datasets and tools used. To compile the phosphorylation network that is the primary input for building the LinkPhinder model, we used the PhosphoSitePlus dataset in a version available on 26th of June 2017 (c.f. https://www.phosphosite.org/staticDownloads.action). There were 10,173 phosphorylation statements on 362, 7,302 and 2,377 distinct kinases, substrate-site combinations and substrates in the compiled phosphorylation network, respectively. Note that in the construction of all datasets, we have focused only on the *Homo Sapiens* species, unless specified otherwise.

In order to convert the phosphorylation statements extracted from PhosphoSitePlus into a knowledge graph, we had to compute motifs characteristic to the context sequences of phosphorylation sites. For that task, we used the MEME tool, version 4.11.2 (c.f. http://meme-suite.org/doc/download.html?man_type=web).

We used three state of the art knowledge graph embedding and link prediction methods to train a model that can discover new links in the phosphorylation knowledge graph. The methods are TransE [32], DistMult [33] and ComplEx [15].

The PhosphoSitePlus dataset, together with UniProt (c.f. http://www.uniprot.org/) was also used for generating a mapping between substrates and their possible phosphorylation sites. This mapping was used in the conversion of the internal, motif-based knowledge graph statements to phosphorylation statements when computing scores of possible phosphorylations that have not been known before. We focused only on substrates present in our knowledge graph, which resulted in 74,142 distinct substrate-site pairs that can be used for generating candidate phosphorylations (i.e. potential discoveries).

To assess LinkPhinder in comparison with related state of the art systems, we downloaded and/or generated full sets of phosphorylation predictions that can be made with the following

tools: Scansite 3 (c.f. http://scansite3.mit.edu), KinomeExplorer (predictions produced by two tools, NetworKIN and NetPhosK, c.f. http://kinomexplorer.info/), Netphos (c.f. http://www.cbs.dtu.dk/services/NetPhos/), GPS (c.f. http://gps.biocuckoo.org/index.php) and PhosphoPredict (c.f. http://phosphopredict.erc.monash.edu/). The numbers of predictions that can be made with the corresponding tools are as follows: 6,130,542 (GPS), 5,192,235 (KinomeExplorer), 3,614,271 (Netphos), 2,006,185 (PhosphoPredict), 311,196 (Scansite 3). The numbers of high-stringency predictions are not straightforward to determine using the set of all predictions available, since some tools allow for stringency settings just at the level of manual, single-protein queries. Thus we were only able to establish the number of high-stringency predictions for Scansite, NetPhos and PhosphoPredict: 12,346, 212,107 and 132, respectively.

Construction of the phosphorylation network and knowledge graph for training the model. The construction of the phosphorylation network requires data sources containing relation information of kinase, substrate and substrate's amino acid phosphorylation site. In our experiment, we used PhosphoSitePlus kinase-substrate dataset, an experimentally determined substrates, sequences, cognate kinases, and metadata curated from the literature [14]. Only relations involving a kinase and substrate protein for the human species were considered (*KIN_ORGANISM == SUB_ORGANISM == 'human'*). Although the dataset includes phosphorylation site's amino acids context sequence of size 7, we did not use that information as we wanted to experiment with different and potentially larger context sequence sizes. Instead we extract the context sequence from UniProt (Universal Protein Resource) and more specifically from the reviewed (Swiss-Prot) main protein sequence (*uniprot_sprot.fasta*) and from isoform sequences (*uniprot_sprot_varsplic.fasta*). We discard any relation in the kinase-substrate dataset for which the phosphorylation site does not match the UniProt sequence. Table 5 presents some statistics about the phosphorylation network.

The knowledge graph conversion makes use of kinase family consensus motifs to transform phosphorylation network statements to knowledge graph relations. The kinase families classification is extracted from UniProt's human and mouse protein kinases: classification and index. Only information about human kinases which are part of the phosphorylation network are kept.

The conversion of phosphorylation network data into knowledge made use of the MEME tool in a pipeline graphically described in Fig 9.

To realise the step 3 of the above pipeline we used specifically the `meme` command line utility for sequence motif discovery, version 4.11.2. MEME was applied in parallel on batches of site context sequences drawn from substrates targeted by kinases of the same family. The size of the batches was a configurable hyper-parameter of the conversion and model training process. We used values ranging over the set {50, 100}. The static parameters used for every invocation of the MEME tool were: `-text`, `-protein`, `-mod zoops`, `-x_branch`, `-minw 2`.

Table 5. Phosphorylation network components statistics.

No. of elements in the phosphorylation network	
Phosphorylation relations	9,802
Kinases	327
Substrates	2,350
Phosphorylation sites	7,083
Avg. No. of substrate/kinase	7.19
Avg. No. of substrate's site/kinase	21.66

https://doi.org/10.1371/journal.pcbi.1007578.t005

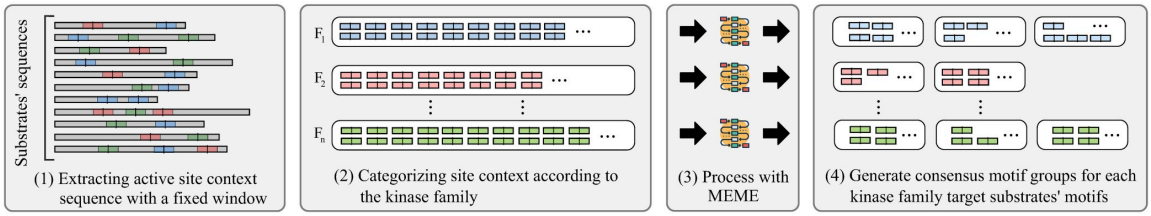


Fig 9. High-level workflow of generating predicate labels for the phosphorylation knowledge graph based on motifs extracted from the context sequences of phosphorylation sites by means of the MEME tool.

https://doi.org/10.1371/journal.pcbi.1007578.g009

The MEME parameters that were dependent on the specific properties of the sequence batch and/or hyper-parameters of the whole model were: `'-maxw MW, -maxsize MS, -nmotifs NM, -bfile BF` where `MW` was the maximum width of a sequence in the batch, `MS` was the maximum width multiplied by the number of sequences in the batch, `NM` was the maximum number of motifs to be generated (set conservatively to 10 in the reported experiments as no batch generated more motifs than that number under any tested settings) and `BF` was a background Markov model of order 5 generated from the sequence batch.

Table 6 presents some statistics about the generated knowledge graph.

Training of the LinkPhinder model. **Generating a phosphorylation knowledge graph.** Before we could train a statistical relational learning model, we had to construct a knowledge graph representing the known phosphorylation information. As the primary input into the knowledge graph, we chose a phosphorylation network compiled from the PhosphoSitePlus [14] data set (focusing on *Homo Sapiens* species only). In principle, any phosphorylation data can be used, but PhosphoSitePlus is well curated and comprehensive making it an ideal starting point. There were 10,173 site-specific phosphorylation statements on 363 and 2,377 distinct kinases and substrates, respectively, in the compiled phosphorylation network. The network consists of statements $\langle K, L, S \rangle$ where K, L, S are kinase, phosphorylation site and substrate, respectively. The biological meaning of such statements is that the kinase K phosphorylates the substrate S by binding to it and attaching a phosphoryl group to the site L.

To convert the phosphorylation network into a knowledge graph, we utilised motifs of phosphorylation sites preferred by specific kinase families. For each kinase family as defined in [34], we computed a set of consensus sequence motifs using the MEME tool run with parameters described in the previous section. The input to the tool were sets of sequences representing the local context of $2k + 1$ amino acids surrounding all phosphorylation sites in substrates targeted by the kinases in each family. The value of k was a configurable hyperparameter of the conversion algorithm representing the context size, i.e. the number of amino acids on the left and right side of the phosphorylation site. See section on Finding the Optimal Hyperparameters of the Model for details on the other hyperparameters. The output of the conversion process were motifs that characterise the local context of the kinase-substrate interaction using

Table 6. Knowledge graph components statistics.

No. of elements in the knowledge graph	
Motif-based relations	9,956
Kinase families	12
Kinase family motifs (relation types)	24
Avg. No. of motif/kinase family	2.00

https://doi.org/10.1371/journal.pcbi.1007578.t006

a position-specific scoring matrix, that quantifies the relative contribution of each amino acid in the substrate sequence. The scoring matrices were extracted from the text output of the MEME tool executed as described above. The motifs were consequently used for converting the $\langle K, L, S\rangle$ statements coming from the input phosphorylation network to labeled knowledge graph edges $\langle K, M, S\rangle$, where M is a link label (also called a typed relation) that corresponds to a motif compatible with the family of K and the site L in the substrate S. Here, compatibility means a positive score of the site's context sequence with respect to the position-specific scoring matrix of the motif M.

The end result of the conversion is a knowledge graph consisting of true positive statements $\langle K, M, S\rangle$. Here, a protein may act as kinase in several statements and as substrate in several other statements. Therewith, these statements describe entire known phosphorylation network from PhosphoSitePlus.

Generating negative statements based on the phosphorylation biology. The knowledge graph generated from the phosphorylation network can be used for discovering new kinase-substrate relationships by means of link prediction [35], which is a technique for estimating likelihood of existence of a typed relationship between two entities based on other observed relationships in the data. The typical intention is discovering new relations that are not explicitly present in a knowledge graph. Training a link prediction model is a supervised machine learning process, and therefore requires negative examples in addition to the positive statements in the phosphorylation knowledge graph. Such negative examples are typically created by corruption of the positive statements by introducing random entities as part of the positive relation statements [35]. In our case, this technique could lead to correct kinase-substrate relationships being treated as negatives because kinases are promiscuous (i.e. one kinase can phosphorylate many substrates and one phosphorylation site can be targeted by many kinases). Hence, random corruptions of true statements may generate many false negative statements. Such false negatives would adversely affect the discriminative power of the model. Therefore, we need to impose specific restrictions when generating negative statements. We based these constraints on biological knowledge as follows. Firstly, most kinases belong to families that usually share substrates, while different families tend to phosphorylate different substrates [34]. Secondly, substrates are unlikely to be phosphorylated by a kinase if they have highly incompatible phosphorylation sites with respect to the kinase consensus motif. This incompatibility directly motivates two types of corruptions. For a statement $\langle K, M, S\rangle$, valid corruptions are: i) statements $\langle \bar{K}, M, S\rangle$ such that $\bar{K}$ is from a different family than K; ii) statements $\langle K, M, \bar{S}\rangle$ such that all phosphorylation sites in $\bar{S}$ score negatively with respect to the scoring matrix of the motif M.

Training the model on the full input dataset to maximise its generalisation power. The model with best-performing hyper-parameters was retrained on the entire knowledge graph derived from PhosphoSitePlus. This is appropriate due to the excellent numerical stability reported in Table 1. The main reason for training the model on the entire dataset is that such a strategy is preferable for making new discoveries because it uses all available information.

The model can be used for computing probabilistic ranking scores (with values between 0 and 1) of predictions ranging across all possible combinations of kinases, sites and substrates present in PhosphoSitePlus, and thus contribute to the discovery of previously unknown phosphorylations.

As described in Fig 2 and the prior parts of this section, the core link prediction model works on the converted knowledge graph, which means that it can only deal with relationships that abstract the site information using motifs. Putative phosphorylations for which the model is supposed to compute scores, therefore, have to be converted to the same form. After the

converted phosphorylation statements are scored, they have to be transformed back to the form that contains the specific phosphorylation site. This conversion is dual to the knowledge graph conversion—each statement $\langle K, M, S\rangle$ corresponds to statements $\langle K, L, S\rangle$ such that L is a known phosphorylation site in S (as per the PhosphoSitePlus [36] and UniProt data sets) that scores positively with respect to the position-specific scoring matrix of the motif M.

Given a single protein as a query, the model can produce a ranked set of candidate phosphorylation sites that involve the protein either as a substrate or as a kinase. The ranked list can optionally be filtered using high- or medium-stringency thresholds. We apply a threshold derived from the manually curated phosphorylation network we use as an input—the high-stringency threshold is a value such that 99.5% of the known phosphorylations score above it (the value is 0.672 in the reported model). The medium-stringency threshold is 0.5 (i.e. a score that indicates higher-than-random plausibility of the given statement). The ranking of the results reflects the global network context of all known phosphorylation sites and kinase-substrate relationships represented in the input knowledge graph, which is a type of information that is not incorporated by any other existing tool. Moreover, the predictions can be generated on any protein, be it a kinase or substrate.

This coverage and flexibility makes our model more powerful than most existing phosphorylation prediction tools that can only be queried for substrate proteins (in the GPS and PhosphoPredict tools, one can generate predictions associated with a kinase, but the systems combined still cover only about half of the kinases covered by LinkPhinder).

In total, LinkPhinder can produce 11,581,940 predictions when applied to all putative phosphorylations that can be generated from the proteins and phosphorylation sites present in the input data (PhosphoSitePlus). Out of these, 2,009,171 and 7,232,636 are of high and medium stringency, respectively. We can make predictions for 327 human kinases, nearly twice as many predictions than the next best among six related methods we have tested (GPS [8], with 217). This shows substantial improvement in the kinome and also general proteome coverage.

Further details and information about the coverage of LinkPhinder compared to other systems can be found in Table 7.

Finding the optimal hyperparameters of the model. Prediction of phosphorylation reactions is based on models trained on the knowledge graph data consisting of positive and negative statements. Negative statements are computed via perturbation of positive statements by means of ad-hoc operators. In our experiments, two negative statements are generated from each positive statement. Data is split into training+validation and testing. In particular, eighty percent of the available data is used for training and validating the models and the remaining part is used for testing. This data is used to evaluate multiple link prediction techniques with the aim of optimising prediction performance. For each of these, a grid search within the space

Table 7. Statistics of the coverage of the different predictive systems and their overlap with the [19] gold standard. The letters S and K in the column headers denote substrates and kinases respectively.

Model	Triplets	Kinases	Substrates	K-S pairs	S-S pairs	S per K	Sites per S
Cutilass20	19066 (100.0%)	103 (100.0%)	2556 (100.0%)	15178 (100.0%)	6090 (100.0%)	147.4	2.4
GPS	6130543 (5.3%)	218 (62.1%)	2531 (35.9%)	516158 (23.7%)	293070 (42.8%)	2367.7	115.8
Netphos	3614272 (2.8%)	18 (5.8%)	2531 (35.9%)	42957 (2.7%)	293354 (43.2%)	2386.5	115.9
Networkin	5192236 (0.0%)	206 (55.3%)	6676 (70.0%)	986494 (35.1%)	40737 (0.0%)	4788.8	6.1
Phosphopredict	2006186 (0.0%)	13 (1.0%)	40624 (99.7%)	252509 (0.1%)	1332427 (25.1%)	19423.8	32.8
Scansite	311197 (0.7%)	34 (16.5%)	2530 (35.9%)	61268 (5.3%)	157214 (36.5%)	1802.0	62.1
netphorest	5192236 (0.0%)	206 (55.3%)	6676 (70.0%)	986494 (35.1%)	40737 (0.0%)	4788.8	6.1
LinkPhinder	11581940 (26.2%)	327 (84.5%)	2350 (33.2%)	738518 (35.7%)	63509 (39.7%)	2258.5	27.0

https://doi.org/10.1371/journal.pcbi.1007578.t007

of available hyperparameter values of the models is performed. For each configuration, 10-fold cross validation is run. The combination of prediction technique and parameters that delivers the best performance is selected and this information is used to train a model on all the available data in order to exploit the entire knowledge about the phosphorylation reactions that have been experimentally validated.

Three link prediction techniques have been used, they are: *TransE*, *DistMult* and *ComplEx* [15, 32, 33]. TransE is one of the earliest techniques to have been proposed and its simplicity makes it a valid reference to learn about embeddings. In our case these embeddings are entities and relation types that are represented by means of vectors of the same length. A true statement is expected to satisfy the vectorial expression *subject* + *relation type* ≈ *object*. DistMult adopts a different approach, the score is the sum of the element-wise products between the subject vector, a diagonal matrix representing the relation type and the object vector:

$$score = \sum_{i=1}^{d} subject_i \cdot relation_i \cdot object_i$$

This denotes that the score is not built considering inter-relations between different latent features. ComplEx follows the same approach as DistMult, with the difference that complex numbers are used in place of real values. The score is the real part of the score formula used in DistMult.

The hyperparameters that control model generation are, in this order: number of negatives generated for each positive statement; number of training epochs through which the model parameters are optimised; number of batches in which data for model training is divided; batch size of amino acid sequences for motif generation (it affects the number of relation types); number of dimensions of vectors; margin of the hinge loss; distance function for computing similarity (only for TransE); learning rate of the model and, ultimately, context size, namely, the number of amino acids to consider on the left and on the right of the binding site. While for some hyperparameters values are selected from a set, for others the values are fixed as they were determined by means of independent experiments. Their respective values are listed in Table 8.

The link prediction technique that delivers the best performance is ComplEx with vectors of size 50 and context size equal to 15. This configuration was used to train a model on the entire network of phosphorylations and their associated negatives. The trained model is used to predict the likelihood of unobserved phosphorylation reactions actually existing in nature.

Table 8. Hyperparameters space used by grid search to identify the best model (L_1, L_2 stand for Manhattan and Euclidean distance norms, respectively).

hyperparameter	values
number of negatives	2
number of epochs	100
number of batches for model training	10
batch size for motif generation	50
embedding size	{50, 100, 150, 250, 500}
margin	1
similarity (only TransE)	{L_1, L_2}
learning rate	0.1
context size	{7, 15}

https://doi.org/10.1371/journal.pcbi.1007578.t008

Construction of the state of the art prediction data sets. The following paragraphs describe the construction of sets of predictions computed by existing tools that are used in comparative validation of the LinkPhinder model.

Scansite 3. Scansite searches for motifs within protein substrates that are likely to be phosphorylated by a specific protein kinase. It takes as input a protein substrate ID and sequence and gives as output a confidence score for given substrate amino acid sites to be phosphorylated by one of 70 kinases handled by the system. We queried the system with all substrates contained in our phosphorylation network and separately accepted results with low and high stringency leveld.

NetworKIN and NetPhorest. KinomeXplorer framework contains results of both NetworKIN and NetPhorest systems with only the score changing. The KinomeXplorer dataset uses gene identifiers to refer to protein phosphorylation. In order to compare the results with the validation set we had first to use UniProt gene query to recover the protein identifier. After downloading the dataset, we queried UniProt using both EmbID and gene name to resolve a protein ID. In case a query did not yield any result or multiple proteins were returned, the original statement was omitted. Finally, we keept only the system protein identifier-based statement responses that pertained to the proteins contained in our phosphorylation network.

NetPhos 3.1. The NetPhos 3.1 system predicts serine, threonine or tyrosine phosphorylation sites in eukaryotic proteins using ensembles of neural networks. The system can provide predictions for 17 kinases only. Using the stand-alone software package, we queried the system with all substrates and associated sequences contained in our phosphorylation network. The results obtained ar low and high stringency levels were used seperately.

GPS 3.0. Group-based Prediction System (GPS) predicts phosphorylation sites with their cognate protein kinases using a four level kinase hierarchical structure in multiple species. We used the batch predictor of the desktop application to pull out results for all substrates and associated sequences contained in our phosphorylation network.

PhosphoPredict. The PhosphoPredict system reportedly predicts kinase-specific substrates and the corresponding phosphorylation sites for 12 human kinases, including CSNK1A1, CSNK2A1, PRKACA, ATM, AKT1 (aka. PKB), SRC, GRK, PKC, GSK, CaMK, CDKs and MAPKs. However, only six of these actually correspond to single kinases, whereas the other seven are often rather diverse families of different proteins (CDKs, MAPKs, PKC, GRK, GSK, CaMK), and thus we focused on them in our comparison. PhosphoPredict employs a feature selection method based on the minimum Redundancy and Maximum Relevance (mRMR) to select the most informative feature subsets that contribute to the prediction success of each kinase families. We keep only those system statements which referred to the proteins present in our phosphorylation network.

Comparative computational validation. A comparative evaluation was performed with the purpose of assessing the performance of LinkPhinder in the context of existing phosphorylation prediction methods (i.e. GPS, NetworKin, NetPhorest, NetPhosK, Scansite and Phosphopredict). Since the process of training LinkPhinder is stochastic, the performance changes slightly every time a new model is trained. To minimise the variability of the results, and allow for comparison and repeatability of the experiment, the results we reported in the main part of this work were averaged over 100 runs of the experiment. The dataset generated for each run consists of positive triples, extracted from PhosphoSitePlus, and negative triples, generated by randomly combining kinases with (site,substrate) pairs that appear in PhosphoSitePlus. The training split accounts for 90% of the data, the remaining 10% is used for testing. Both training and test set contain equal numbers of positive and negative instances.

To evaluate LinkPhinder, training data are used to learn a model in each run and its performance is evaluated on the test set. Triples in the test set are assigned the prediction score if this is available, otherwise a zero score is assigned.

One note to be taken into account regarding prediction score assignment is this. As stated in the main text, a very accurate model that generates predictions only on a small subset of the triples may be of limited use in phosphorylation prediction. Hence, we also assessed the rate of predictions a model is able to generate by measuring the percentage of triples in the test set for which the model is able to generate a prediction (i.e. non-zero score). We refered to this value as *coverage* in the Results section.

Concerning the existing methods to which we compare ourselves, scores are extracted from the predictions provided by each method (created as described in the previous section). This does not exclude that part of the testing triples may have been used to train the comparative models. Assuming that this is the case, this would represent a disadvantage in terms of performance for our model. Similarly to the LinkPhinder case, coverage is therefore computed over the test data and zero scores are assigned to triples for which a prediction is not available.

Verifying the stability of LinkPhinder under different conditions of the computational experiments. To make sure various decisions made in preparation of the benchmarking data do not influence the presented results in terms of comparing the performance of LinkPhinder and related existing tools, we have first experimented with a different positive-negative ratio (ten negatives per one positives, see Table 9), and then with various different train-test split ratios (Table 10).

The increase in the number of negatives per a positive typically hampers performance of ranking-based models, and Table 9 clearly shows that our experiments are no exception. However, one can also immeditaly notice that LinkPhinder remains by far the best tool, and is significantly less affected by the change. This demonstrates the superior stability of our tool in the context of changing experimental conditions.

The results in Table 10 clearly show that while the performance of LinkPhinder decreases with increasing proportion of testing over the training data, it is still superior to the corresponding

Table 9. LinkPhinder performance compared to other systems on our benchmark with 1:10 positive to negative ratio in the testing split where the training/testing splits are 90% and 10% respecitvely.

Model	AUPR	AUROC	P@10	P@50
GPS	0.259 ± 0.007	0.731 ± 0.006	0.337 ± 0.145	0.416 ± 0.063
NetworKin	0.281 ± 0.009	0.618 ± 0.007	0.798 ± 0.122	0.756 ± 0.055
NetPhorest	0.199 ± 0.007	0.597 ± 0.007	0.542 ± 0.137	0.520 ± 0.071
Scansite	0.149 ± 0.004	0.571 ± 0.006	0.132 ± 0.099	0.210 ± 0.048
Phosphopredict	0.091 ± 0.002	0.500 ± 0.006	0.029 ± 0.050	0.050 ± 0.029
Netphos	0.166 ± 0.006	0.563 ± 0.007	0.426 ± 0.149	0.390 ± 0.064
LinkPhinder	**0.875 ± 0.010**	**0.982 ± 0.002**	**0.993 ± 0.025**	**0.981 ± 0.024**

https://doi.org/10.1371/journal.pcbi.1007578.t009

Table 10. Relative LinkPhinder performance across different training-testing splits where the positive to negative ratio of the testing set is 1:10 (the relative performance results were substantially less variable for the 1:1 ratio, therefore we do not report them here).

Model	AUPR	AUROC	P@10	P@50
Train 60%, Test 40%	0.768 ± 0.006	0.969 ± 0.001	0.987 ± 0.034	0.981 ± 0.017
Train 70%, Test 30%	0.797 ± 0.006	0.974 ± 0.001	0.960 ± 0.049	0.968 ± 0.018
Train 80%, Test 20%	0.835 ± 0.005	0.978 ± 0.001	0.990 ± 0.030	0.984 ± 0.012
Train 90%, Test 10%	0.875 ± 0.010	0.982 ± 0.002	0.993 ± 0.025	0.981 ± 0.024

https://doi.org/10.1371/journal.pcbi.1007578.t010

results of the related works given in Table 9. This further corroborates our claim of LinkPhinder's stability with respect to different experimental conditions.

Generating phosphorylation data for the web interface of LinkPhinder. In order to prepare data of phosphorylation reactions for prediction, a list of known kinases and a list of known substrates with their corresponding phosphorylation sites are extracted from the phosphorylation network. The elements of the lists are combined using their Cartesian product to generate every possible combination of kinase and phosphorylation site of each substrate. These are converted into knowledge graph phosphorylation statements and are then scored using the previously trained best-performing prediction model (*i.e.* the result of the grid search described previously). Finally, knowledge graph statements with their associated scores are converted back to phosphorylation site-specific statements. If there are duplicate statements after the conversion process that only differ in the scores assigned to them by the conversion and the model, we only keep the one with the highest score determined by the model. This is motivated by the fact that the model utilises more information on the actual phosphorylations than the conversion process and therefore its scores override the scores assigned after conversion.

Experimental model and subject details

Cell culture experiments for targeted validation. Hek-293 cells were regularly grown in Dulbecco's modified medium supplemented with 10% foetal serum. Subconfluent cell were transfected with Lipofectamine (Invitrogene) following manufacturer's instructions. pSG5-gag-AKT was previously described [37]. LATS1 siRNA and AKT siRNA were from Dharmacorn and sequences have been described before [29]. Twentyfour hours after tranfection HEK293 cells were serum deprived for 16 hours. Subsiquently, cell were lysed in 20mM HEPES (pH 7.5), 150 mM NaCl, 1% NP-40, phosphatase inhibitors (2mM NaF, 10mMb-Glycerolphosphate, 2 MM Na4P2O4) and protease inhibitors (5 μg/ml Leupeptin and 2.2 μg/ml aprotinin). Cell lysates were separated by SDS-PAGE analysed by western blotting. Phosphorylated proteins were immunoprecipitated with pAKT-Substrate specific antibody. Briefly, the lysates were incubated with 1μl of antibody and 5μl of protein-G sepharose beads for 1 hour at 4C in an orbital wheel. The immunoprecipitates were washed 3 times with lysis buffer. 2 bed volumes of denaturing laemli buffer were added to the dry pelleted beads and immunocomplex were eluted by boiling the samples at 100C for 5 minutes. Anti-creb, anti-LATS1 anti-P53 anti-tub, p-YAP-S127 were obtained from commercial sources.

Mass-spectrometry experiments for extended validation. HeLa cells were transiently transfected with a GFP-tagged LATS1 construct or a GFP construct as control. After 2 days they were serum starved over-night and left untreated (control) or were treated with FasL (50nM) or Etoposide (50*mu*M) for 16 hours. Then, cells were lysed with Lysis buffer (20mM 4-(2 hydroxyethyl)-1piperazineethanesulfonic acid (HEPES) pH7.5, 150mM NaCl, 1% NP-40, phosphatase inhibitors (10 mMβ-Gycerolphosphate, 1 mM Na_3VO_4, 2mM $Na_4P_2O_7$, 2 mM NaF) and protease inhibitors (5 μg/ml Leupeptin and 2.2 μg/ml Aprotinin), and proteins were immunoprecipitated using GFP-trap_A (Chromotek) according to the manufacturer's instructions. The beads were washed 3 times with lysis buffer followed by two washes with the same buffer not containing NP-40. The proteins immunoprecipitated onto GFP-beads were prepared for masss-spectrometry analysis as previously described [38]. Briefly, the immunoprecipitates were digested in two steps. Firstly, by adding 60μl of elution buffer-1 (2M urea, 50mM Tris-HCl pH7.5, 5μg/ml Trypsin), to each sample and incubation at 27˚C on a shaker. After 30 minutes initial digestion the samples were centrifuged at 13,000 rpm in a table top centrifuge for 30 seconds and the supernatant was collected into a new Eppendorf tube. In the

second step 25μl of elution buffer-2 (2M urea, 50mM Tris-HCL pH7.5, 1mM Dithiothreitol) was added per sample followed by centrifugation as above. The supernatant was collected into a new Eppendorf tube. The elution step was repeated, and both supernatants were combined and incubated overnight at room temperature to allow trypsin digestion to go to completion. The, samples were alkylated by adding 20 μl iodoacetamide (5mg/ml), and incubation for 30 min in the dark at room temperature. The reaction was stopped by adding 1 μl 100% Trifluoracetic acid (TFA) to each sample. 100 μl of each sample was immediately loaded into equilibrated handmade C18 StageTips containing Octadecyl C18 disks (Supelco) for desalting. Tips were previously activated by washing with 50μl of 50% AcN and 0.1%TFA. After a quick centrifugation the tips were washed with 50μl of 0.1%TFA. 100μl samples was loaded onto the tip washed twice with 50μl of 0.1% TFA and eluted twice with 25μl of 50% AcN and 0.1% TFA solution. The eluates were combined and concentrated until the volume was reduced to 5μl using a CentriVap Concentrator (Labconco). Samples were diluted to obtain a final volume of 15μl by adding 0.1% TFA and centrifuged for 10 minutes at 13000rpm. 12μl of the samples were analysed by MS. The samples were analysed by liquid Chromatography-Tandem Mass Spectrometry (Nanoflow Ultimate 3000 LC and Q-Exactive mass spectrometer [Thermo]). A 10 cm long, 75 μm inner diameter, HLPC c18-reversed pahes column was used. Samples were loaded at 600nl/min and peptides were eluted at a constant flow rate of 250nl/ min for 40 min. A multisegment linear gradient of 2-135% buffer (98% Acetonitrile and 0.1% formic acid) in positive ion mode was used. Data were acquired with the mass spectrometer operating in automatic data dependent switching mode selecting the 12 most intense ions prior to MS/MS analysis. Mass spectra were analysed by MaxQuant. Label-free quantitation was performed using MaxQuant.

PKA Kinase assay. Serum straved HEK293T were lysed in a Nonidet P-40 buffer (50 mM Tris-HCl, pH 7.8, 150 mM NaCl, 1% (vol/vol) Nonidet P-40, protease inhibitors and phosphatase inhibitors). Lysates were treated at 1 mg/ml with 10 mM 5'-4-fluorosulphonylbenzoyladenosine (FSBA) solubilised in DMSO and then incubated at 31˚C for 2 hour. Samples were spun down at 200 x g to remove any precipitate. Sample were diluted down with 2 ml of PKA kinase buffer (50 mM Tris pH 7.5, 10 mM MgCl2, 0.1 mM EGTA and 2 mM DTT) and desalted using a Millipore Amicon ultrafiltration columns with a 3 kDa molecular weight cut-off. Following concentration, the samples were incubated with PKA kinase buffer (50 mM Tris pH 7.5, 10 mM MgCl2, 0.1 mM EGTA and 2 mM DTT), 500 uM ATP-biotin and 1250 units of recombinant PKA (New England Biolabs) in a total volume of 60 μl. Control samples without recombinant PKA and ATP-biotin were also made up. The controls and kinase-added samples were incubated at 31˚C for 2 hours. 300 μl of phosphate buffer was added to the samples. Streptavidin resin (100 μl of a 50% slurry) was incubated with the samples overnight at 4˚C. Samples were spun down samples at 2000 x g for 1 minute and the supernatant was removed. Samples were washed 5 times with 1 ml of phosphate buffer. Samples were analysed by mass spectrometry.

The full results of the assay are given in the kinase assays supplement (S1 Table).

MST2 Kinase assay. Serum straved HEK293T cells were treated with 3 μM of the MST2 kinase specific inhibitor, XMU-MP-1 or DMSO for 3 hours. Cells were lysed in a Nonidet P-40 buffer (50 mM Tris-HCl, pH 7.8, 150 mM NaCl, 1% (vol/vol) Nonidet P-40, protease inhibitors and phosphatase inhibitors). Lysates were treated at 1 mg/ml with 10 mM 5'-4-fluorosulphonylbenzoyladenosine (FSBA) solubilised in DMSO and then incubated at 31˚C for 2 hour. Samples were spun down at 200 x g to remove any precipitate. Sample were diluted down with 2 ml of MST2 kinase buffer (40 mM HEPES pH 8.0, 10 mM MgCl2, 0.5 mM EGTA) and desalted using a Millipore Amicon ultrafiltration columns with a 3 kDa molecular weight cut-off. Following concentration, the samples were incubated with MST2 kinase assay buffer (40

mM HEPES pH 8.0, 10 mM MgCl2, 0.5 mM EGTA), 500 uM ATP-biotin and 32 ng of recombinant MST2 (made in house) in a total volume of 60 μl. Control samples without recombinant MST2 and ATP-biotin were also made up. The controls and kinase-added samples were incubated at room temperature for 3 hours. 300 μl of phosphate buffer was added to the samples. Streptavidin resin (100 μl of a 50% slurry) was incubated with the samples for 1 hour at room temperature. Samples were spun down samples at 2000 x g for 1 minute and the supernatant was removed. Samples were washed 5 times with 1 ml of phosphate buffer. Samples were analysed by mass spectrometry.

The full results of the assay are given in the kinase assays supplement (S2 Table).

LATS1 Kinase assay. Serum straved HEK293T cells were lysed in a Nonidet P-40 buffer (50 mM Tris-HCl, pH 7.8, 150 mM NaCl, 1% (vol/vol) Nonidet P-40, protease inhibitors and phosphatase inhibitors). Lysates were treated at 1 mg/ml with 10 mM 5'-4-fluorosulphonylbenzoyladenosine (FSBA) solubilised in DMSO and then incubated at 31˚C for 2 hour. Samples were spun down at 200 x g to remove any precipitate. Sample were diluted down with 2 ml of LATS1 kinase buffer (25 mM HEPES pH 7.4, 50 mM NaCl, 5 mM MgCl2 and 5 mM MnCl2, 5 mM β-glycerophosphate and 1 mM dithiothreitol) and desalted using a Millipore Amicon ultrafiltration columns with a 3 kDa molecular weight cutoff. Following concentration, the samples were incubated with LATS1 kinase assay buffer (25 mM HEPES pH 7.4, 50 mM NaCl, 5 mM MgCl2 and 5 mM MnCl2, 5 mM β-glycerophosphate and 1 mM dithiothreitol), 500 uM ATP-biotin and 100 ng of recombinant LATS1 (Abcam) in a total volume of 60 μl. Control samples without recombinant LATS1 and ATP-biotin were also made up. The controls and kinase-added samples were incubated at 30˚C for 30 minutes. 300 μl of phosphate buffer was added to the samples. Streptavidin resin (100 μl of a 50% slurry) was incubated with the samples for 1 hour at room temperature. Samples were spun down samples at 2000 x g for 1 minute and the supernatant was removed. Samples were washed 5 times with 1 ml of phosphate buffer. Samples were analysed by mass spectrometry.

The full results of the assay are given in the kinase assays supplement (S3 Table).

Mass spectrometry sample preparation. The streptavidin resin containing the bound proteins were incubated with 400 μl of elution buffer I (50 mM Tris-HCl ph 7.5, 2 M Urea, 181 ng/μl trypsin) at 37˚C for 30 minutes. The samples were spun at 2000 x g and the superntant was retained. To the streptavidin resin 330 μl of elution buffer II (50 mM Tris-HCl ph 7.5, 2 M Urea, 1 mM DTT) at 37˚C for 1 hour. The samples were spun at 2000 x g and the superntant was retained. The two supernatant of elution buffers I and II were combined and incubated overnight at 37˚C. After the incubation 130 μl of 5 mg/ml Iodocetamide was added to each and the samples were incubated for 30 minutes at room temperature in the dark. C18 stage tips that were previously prepared were mounted into a 1.5 ml eppendorf were activated by adding 50 μl of 50% acetonitrile (AcN) and 0.1% Trifluoroacetic acid (TFA). The samples were spun at 5000 rpm for 1 minute. 50 μl of 1% TFA was added to the C18 stage tips and the samples were spun at 5000 rpm. After the Iodocetamide incubation the reaction was stopped by adding 1 μl of 100% TFA. The samples were loaded onto the C18 stage tips and they were spun at 5000 rpm. The C18 stage tips were then washed by adding 50 μl of 1% TFA and then the samples were spun at 5000 rpm, this was done twice. Before elution of the samples, the C18 stage tips were mounted into fresh 1.5 ml eppendorfs. The peptides were eluted of the C18 stage tips by adding 25 μl of 50% AcN and 0.1% TFA and spinning the samples at 5000 rpm, this was repeated twice. Samples were evaporated for 10-15 in a CentriVap concentrator until 5 μl was left. The sample was then respuspended in 20 μl of TFA. The samples were then analysed by mass spectrometry.

Mass spectrometry. Mass spectrometry was performed using a Ultimate 3000 RSLC system that was coupled to an Orbitrap Fusion Tribrid mass spectrometer (Thermo Fisher Scientific).

Following tryptic digest, the peptides were loaded onto a nano-trap column (300 μM i.d x 5mm precolumn that was packed with Acclaim PepMap100 C18, 5 μM, 100 Å; Thermo Scientific) running at a flow rate of 30 μl/min in 0.1% trifluoroacetic acid made up in HPLC water. The peptides were eluted and separated on the analytical column (75 μM i.d. × 25 cm, Acclaim PepMap RSLC C18, 2 μM, 100 Å; Thermo Scientific) after 3 minutes by a linear gradient from 2% to 30%of buffer B (80% acetonitrile and 0.08% formic acid in HPLC water) in buffer A (2% acentonitrile and 0.1% formic acid in HPLC water) using a flow rate if 300 nl/min over 150 minutes. The remaining peptides were eluted using a short gradient from 30% to 95% in buffer B for 10 minutes. The mass spectrometry parameters were as follows: for full mass spectrometry spectra, the scan range was 335-1500 with a resolution of 120,000 at m/z = 200. MS/MS acquisition was performed using top speed mode with 3 seconds cycle time. Maximum injection time was 50 ms. The AGC target was set to 400,000, and the isolation window was 1 m/z. Positive Ions with charge states 2-7 were sequentially fragmented by higher energy collisional dissociation. The dynamic exclusion duration was set at 60 seconds and the lock mass option was activated and set to a background signal with a mass of 445.12002.

Analysis of mass spectrometry data. Analysis was performed using MaxQuant (version 1.5.3.30). Trypsin was set to be the digesting enzyme with maximal 2 missed cleavages. Cysteine carbmidomethylation was set for fixed modifications and oxidation of methionine and N-thermal acetylation were specified as variable modifications. The data was then analysed with the minimum ratio count of 2. The first search peptide was set to 20, the main search peptide tolerance to 5 ppm and the "re-quantify" option was selected. For protein and peptide identification the Human subset of the SwissProt database (Release 2015_12) was used and the contaminants were detected using the MaxQuant contaminant search. A minimum peptide number of 1 and a minimum of 6 amino acids was tolerated. Unique and razor peptides were used for quantification. The match between run option was enabled with a match time window of 0.7 minutes and an alignment window of 20 minutes.

Quantification and statistical analysis

Peptide identification. MaxQuant (version 1.3.0.5.) was used to analyse raw mass spectrometric data files from LC-MS/MS for protein quantification. Default settings were used unless stated otherwise, including the following parameters: Trypsin/P digest allowing for 2 misscleavages; variable modifications included oxidation and acetylation; fixed modification included carbamidomethylation (at Cysteine); to detect phosphopeptides we included phospho (STY) as a modification; first search at 20 ppm: main search at 6 ppm mass accuracy (MS) and 20ppm mass deviation for the fragment ions. The MS data were searched against a human database (Uniprot HUMAN) with a minimum peptide length of 6, unfiltered for labelled amino acids, at a false discovery rate (FDR) of 0.01 for peptides and proteins. The results were refined through the re-quantify option; also "match between runs" was selected with a 1 min time window, and label free quantification was selected with the minimum ratio count set at 1.

Supporting information

S1 Table. PKA Kinase Assay Results (an PDF file; c.f. https://doi.org/10.6084/m9.figshare.13118441).
(PDF)

S2 Table. MST2 Kinase Assay Results (an PDF file; c.f. https://doi.org/10.6084/m9.figshare.13118477).
(PDF)

S3 Table. LATS1 Kinase Assay Results (an PDF file; c.f. https://doi.org/10.6084/m9.figshare.13118483).
(PDF)

S4 Table. Mass-spec results for the LATS1 IP (an xlsx file; c.f. https://doi.org/10.6084/m9.figshare.12173163).
(XLSX)

S5 Table. Mass-spec data for the PKA kinase assay (an xlsx file; c.f. https://figshare.com/articles/Mass_spec_data_PKA/12200681).
(XLSX)

S6 Table. Mass-spec data for the MST2 kinase assay (an xlsx file; c.f. https://doi.org/10.6084/m9.figshare.12200675.v1).
(XLSX)

S7 Table. Mass-spec data for the LATS1 kinase assay (an xlsx file; c.f. https://figshare.com/articles/Mass_spec_data_LATS_kinase_assay/12200597).
(XLSX)

S1 Data. Full set of LinkPhinder predictions (a single bzip2-archived CSV file; c.f. https://doi.org/10.6084/m9.figshare.12173100).
(BZ2)

S2 Data. Full set of predictions computed by the related works (a bzip2-archive of 6 CSV files for each of the related tools; c.f. https://doi.org/10.6084/m9.figshare.12173109).
(TBZ)

S1 Fig. Supporting details on the experimental validation of the LATS1/YAP1 phosphorylation: (A-B) HEK293 were transfected with the indicated siRNAs. 48 hours after transfection the cells were lysed and blotted with the indicated antibodies. (C) HEK293 were transfected with empty vector (EV) or GAG-AKT or treated with AKTi IV (10M) for 1 hour. Phosphorylated proteins were immunoprecipitated using an anti-AKT antibody and the immunoprecipitates were blotted with the indicated antibodies (a PDF figure, c.f. https://doi.org/10.6084/m9.figshare.13118561).
(PDF)

Author Contributions

Conceptualization: Vít Nováček, Pierre-Yves Vandenbussche, Walter Kolch, Dirk Fey.

Data curation: Piero Conca, Emir Muñoz, Luca Costabello, Kamalesh Kanakaraj, Zeeshan Nawaz, Pierre-Yves Vandenbussche.

Funding acquisition: Vít Nováček, David Matallanas, Pierre-Yves Vandenbussche, Walter Kolch, Dirk Fey.

Methodology: Vít Nováček, David Matallanas, Pierre-Yves Vandenbussche, Walter Kolch, Dirk Fey.

Project administration: Vít Nováček, Pierre-Yves Vandenbussche, Walter Kolch.

Resources: Gavin McGauran, David Matallanas, Adrián Vallejo Blanco, Walter Kolch, Dirk Fey.

Software: Vít Nováček, Piero Conca, Emir Muñoz, Luca Costabello, Kamalesh Kanakaraj, Zeeshan Nawaz, Sameh K. Mohamed.

Supervision: Vít Nováček, Pierre-Yves Vandenbussche, Walter Kolch.

Validation: Vít Nováček, Gavin McGauran, David Matallanas, Adrián Vallejo Blanco, Piero Conca, Emir Muñoz, Luca Costabello, Kamalesh Kanakaraj, Zeeshan Nawaz, Brian Walsh, Sameh K. Mohamed, Pierre-Yves Vandenbussche, Dirk Fey.

Visualization: Kamalesh Kanakaraj, Zeeshan Nawaz.

Writing – original draft: Vít Nováček, Pierre-Yves Vandenbussche, Dirk Fey.

Writing – review & editing: Colm J. Ryan, Walter Kolch.

References

1. Kolch W, Halasz M, Granovskaya M, Kholodenko BN. The dynamic control of signal transduction networks in cancer cells. Nature Reviews Cancer. 2015; 15(9):515. https://doi.org/10.1038/nrc3983
2. Ferguson FM, Gray NS. Kinase inhibitors: the road ahead. Nature Reviews Drug Discovery. 2018; 17 (5):353. https://doi.org/10.1038/nrd.2018.21
3. Cohen P, Alessi DR. Kinase drug discovery–what's next in the field? ACS chemical biology. 2012; 8 (1):96–104.
4. Wu P, Nielsen TE, Clausen MH. FDA-approved small-molecule kinase inhibitors. Trends in pharmacological sciences. 2015; 36(7):422–439. https://doi.org/10.1016/j.tips.2015.04.005
5. Dinkel H, Chica C, Via A, Gould CM, Jensen LJ, Gibson TJ, et al. Phospho. ELM: a database of phosphorylation sites—update 2011. Nucleic acids research. 2011; 39(suppl 1):D261–D267. https://doi.org/10.1093/nar/gkq1104 PMID: 21062810
6. Linding R, Jensen LJ, Ostheimer GJ, van Vugt MA, Jørgensen C, Miron IM, et al. Systematic discovery of in vivo phosphorylation networks. Cell. 2007; 129(7):1415–1426. https://doi.org/10.1016/j.cell.2007.05.052 PMID: 17570479
7. Obenauer JC, Cantley LC, Yaffe MB. Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. Nucleic acids research. 2003; 31(13):3635–3641. https://doi.org/10.1093/nar/gkg584
8. Xue Y, Ren J, Gao X, Jin C, Wen L, Yao X. GPS 2.0, a tool to predict kinase-specific phosphorylation sites in hierarchy. Molecular & cellular proteomics. 2008; 7(9):1598–1608. https://doi.org/10.1074/mcp.M700574-MCP200
9. Blom N, Sicheritz-Pontén T, Gupta R, Gammeltoft S, Brunak S. Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. Proteomics. 2004; 4(6):1633–1649. https://doi.org/10.1002/pmic.200300771
10. Horn H, Schoof EM, Kim J, Robin X, Miller ML, Diella F, et al. KinomeXplorer: an integrated platform for kinome biology studies. Nature methods. 2014; 11(6):603–604. https://doi.org/10.1038/nmeth.2968 PMID: 24874572
11. Song J, Wang H, Wang J, Leier A, Marquez-Lago T, Yang B, et al. PhosphoPredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection. Scientific Reports. 2017; 7(1):6862. https://doi.org/10.1038/s41598-017-07199-4 PMID: 28761071
12. Venter JC, Adams MD, Myers EW, Li PW, Mural RJ, Sutton GG, et al. The sequence of the human genome. science. 2001; 291(5507):1304–1351. https://doi.org/10.1126/science.1058040 PMID: 11181995
13. Wang Q, Mao Z, Wang B, Guo L. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering. 2017; 29(12):2724–2743. https://doi.org/10.1109/TKDE.2017.2754499
14. Hornbeck PV, Zhang B, Murray B, Kornhauser JM, Latham V, Skrzypek E. PhosphoSitePlus, 2014: mutations, PTMs and recalibrations. Nucleic acids research. 2015; 43(D1):D512–D520. https://doi.org/10.1093/nar/gku1267
15. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. arXiv preprint arXiv:160606357. 2016;.

16. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research. 2012; 13(Feb):281–305.
17. Needham EJ, Parker BL, Burykin T, James DE, Humphrey SJ. Illuminating the dark phosphoproteome. Sci Signal. 2019; 12(565):eaau8645. https://doi.org/10.1126/scisignal.aau8645
18. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In: Proceedings of the 23rd international conference on Machine learning. ACM; 2006. p. 233–240.
19. Hijazi M, Smith R, Rajeeve V, Bessant C, Cutillas PR. Reconstructing kinase network topologies from phosphoproteomics data reveals cancer-associated rewiring. Nature Biotechnology. 2020; p. 1–10.
20. Martini M, De Santis MC, Braccini L, Gulluni F, Hirsch E. PI3K/AKT signaling pathway and cancer: an updated review. Annals of medicine. 2014; 46(6):372–383. https://doi.org/10.3109/07853890.2014.912836
21. Fallahi E, O'Driscoll NA, Matallanas D. The MST/Hippo pathway and cell death: a non-canonical affair. Genes. 2016; 7(6):28. https://doi.org/10.3390/genes7060028
22. Gomez M, Gomez V, Hergovich A. The Hippo pathway in disease and therapy: cancer and beyond. Clinical and translational medicine. 2014; 3(1):22.
23. Mayer IA, Arteaga CL. The PI3K/AKT pathway as a target for cancer treatment. Annual review of medicine. 2016; 67:11–28. https://doi.org/10.1146/annurev-med-062913-051343
24. Technology CS. PI3K / Akt Substrates Table;. https://www.cellsignal.com/contents/resources-reference-tables/pi3k-akt-substrates-table/science-tables-akt-substrate.
25. Mantamadiotis T, Papalexis N, Dworkin S. CREB signalling in neural stem/progenitor cells: recent developments and the implications for brain tumour biology. Bioessays. 2012; 34(4):293–300. https://doi.org/10.1002/bies.201100133
26. Wang J, Ma L, Weng W, Qiao Y, Zhang Y, He J, et al. Mutual interaction between YAP and CREB promotes tumorigenesis in liver cancer. Hepatology. 2013; 58(3):1011–1020. https://doi.org/10.1002/hep.26420 PMID: 23532963
27. Romano D, Matallanas D, Weitsman G, Preisinger C, Ng T, Kolch W. Proapoptotic kinase MST2 coordinates signaling crosstalk between RASSF1A, Raf-1, and Akt. Cancer research. 2010; p. 0008–5472.
28. Von Kriegsheim A, Baiocchi D, Birtwistle M, Sumpton D, Bienvenut W, Morrice N, et al. Cell fate decisions are specified by the dynamic ERK interactome. Nature cell biology. 2009; 11(12):1458. https://doi.org/10.1038/ncb1994 PMID: 19935650
29. Matallanas D, Romano D, Yee K, Meissl K, Kucerova L, Piazzolla D, et al. RASSF1A elicits apoptosis through an MST2 pathway directing proapoptotic transcription by the p73 tumor suppressor protein. Molecular cell. 2007; 27(6):962–975. https://doi.org/10.1016/j.molcel.2007.08.008 PMID: 17889669
30. Embogama DM, Pflum MKH. K-BILDS: A Kinase Substrate Discovery Tool. ChemBioChem. 2017; 18(1):136–141. https://doi.org/10.1002/cbic.201600511
31. Hernandez-Armenta C, Ochoa D, Gonçalves E, Saez-Rodriguez J, Beltrao P. Benchmarking substrate-based kinase activity inference using phosphoproteomic data. Bioinformatics. 2017; 33(12):1845–1851. https://doi.org/10.1093/bioinformatics/btx082
32. 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; 2013. p. 2787–2795.
33. Yang B, Yih Wt, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:14126575. 2014;.
34. Manning G, Whyte DB, Martinez R, Hunter T, Sudarsanam S. The protein kinase complement of the human genome. Science. 2002; 298(5600):1912–1934. https://doi.org/10.1126/science.1075762
35. Nickel M, Murphy K, Tresp V, Gabrilovich E. A Review of Relational Machine Learning for Knowledge Graphs. Proceedings of the IEEE. 2016; 104(1):11–33. https://doi.org/10.1109/JPROC.2015.2483592
36. Hornbeck PV, Kornhauser JM, Tkachev S, Zhang B, Skrzypek Ez, Murray B, et al. PhosphoSitePlus: a comprehensive resource for investigating the structure and function of experimentally determined post-translational modifications in man and mouse. Nucleic acids research. 2011; 40(D1):11–22. https://doi.org/10.1093/nar/gkr1122 PMID: 22135298
37. Boudewijn MT, Coffer PJ, et al. Protein kinase B (c-Akt) in phosphatidylinositol-3-OH kinase signal transduction. Nature. 1995; 376(6541):599. https://doi.org/10.1038/376599a0
38. Turriziani B, Garcia-Munoz A, Pilkington R, Raso C, Kolch W, von Kriegsheim A. On-beads digestion in conjunction with data-dependent mass spectrometry: a shortcut to quantitative and dynamic interaction proteomics. Biology. 2014; 3(2):320–332. https://doi.org/10.3390/biology3020320

Chapter 13

One Method to Rule Them All

Briefings in Bioinformatics, 00(0), 2020, 1–15

doi: 10.1093/bib/bbaa012
Advance Access Publication Date: 00 Month 20xx
Review Article

Biological applications of knowledge graph embedding models

Sameh K. Mohamed, Aayah Nounu and Vít Nováček

Corresponding author: Sameh K. Mohamed, Insight Centre for Data Analytics, IDA Business Park, Lower Dangen, Galway, Ireland. Tel.: +353 91 495730. Email: sameh.kamal@insight-centre.org

Abstract

Complex biological systems are traditionally modelled as graphs of interconnected biological entities. These graphs, i.e. biological knowledge graphs, are then processed using graph exploratory approaches to perform different types of analytical and predictive tasks. Despite the high predictive accuracy of these approaches, they have limited scalability due to their dependency on time-consuming path exploratory procedures. In recent years, owing to the rapid advances of computational technologies, new approaches for modelling graphs and mining them with high accuracy and scalability have emerged. These approaches, i.e. knowledge graph embedding (KGE) models, operate by learning low-rank vector representations of graph nodes and edges that preserve the graph's inherent structure. These approaches were used to analyse knowledge graphs from different domains where they showed superior performance and accuracy compared to previous graph exploratory approaches. In this work, we study this class of models in the context of biological knowledge graphs and their different applications. We then show how KGE models can be a natural fit for representing complex biological knowledge modelled as graphs. We also discuss their predictive and analytical capabilities in different biology applications. In this regard, we present two example case studies that demonstrate the capabilities of KGE models: prediction of drug–target interactions and polypharmacy side effects. Finally, we analyse different practical considerations for KGEs, and we discuss possible opportunities and challenges related to adopting them for modelling biological systems.

Key words: biomedical knowledge graphs; knowledge graph embeddings; tensor factorization; link prediction; drug–target interactions; polypharmacy side effects.

Sameh K. Mohamed is a PhD student in Computer Science at Insight Centre, National University of Ireland Galway, and a researcher at the Data Science Institute in Galway, Ireland. His main interests lie within the areas of machine learning and bioinformatics. He is currently focused on representational learning and knowledge graphs mining.
Vit Novacek holds a PhD from National University of Ireland Galway and currently leads the Biomedical Discovery Informatics Unit at Data Science Institute, National University of Ireland Galway where he also is a research fellow and adjunct lecturer. His personal research interests revolve around developing machine-aided discovery solutions by means of machine/representation learning, explainable AI and text mining, with a strong focus on biomedical use cases.
Aayah Nounu holds a PhD from The University of Bristol. Her background is in combining both laboratory-based methods and epidemiological methods to understand drug mechanisms associated with cancer prevention. She is currently working in a research post looking at the effect of aspirin for the prevention of colorectal cancer.
Data Science Institute is specialized in research technologies at the convergence of computer science, web science and artificial intelligence to build a fundamental understanding of how information and knowledge are increasingly driving society through digital processes and of the tools, techniques and principles supporting a data-enhanced world.
Submitted: 18 September 2019; **Received (in revised form):** 10 January 2020

Introduction

Biological systems consist of complex interconnected biological entities that work together to sustain life in living systems. This occurs through complex and systematic biological interactions of the different biological entities. Understanding these interactions is the key to elucidating the mechanism of action of the different biological functions (e.g. angiogenesis, metabolism, apoptosis, etc.) and thus understanding causes and activities of diseases and their possible therapies. This encouraged the development of multiple physical and computational methods to assess, verify and infer different types of these interactions. In this study, we focus on the use of computational methods for assessing and inferring interactions (associations) between different biological entities at the molecular level. We hereof study the use of knowledge graphs and their embedding models for modelling molecular biological systems and the interactions of their entities.

Initially, basic networks, i.e. uni-relational graphs, were adopted by early efforts for modelling complex interactions in biological systems [1–4]. Despite their initial success [5], these networks could not preserve the semantics of different types of associations between entities. For example, protein–protein interaction networks modelled with basic networks cannot differentiate between different types of interactions such as inhibition, activation, phosphorylation, etc. Therefore, more recent works modelled biological systems using heterogeneous multi-relational networks i.e. knowledge graphs, where they utilized different visual [6, 7] and latent representations [8, 9] of graph entities to infer associations between them.

In the context of biological applications, knowledge graphs were used to model biological data in different projects such as the UNIPROT [10], Gene Ontology [11] and Bio2RDF [12] knowledge bases. Moreover, they were the basis of multiple predictive models for drug adverse reactions [6, 8], drug repurposing [9, 13] and other predictions for different types of biological concepts associations [13, 14]. The task of learning biological associations in this context is modelled as link prediction in knowledge graphs [15]. Predictive models then try to infer a typed link between two nodes in the graph using two different types of features: graph features and latent-space vector representations.

Graph features models (i.e. visual feature models) are part of the network analysis methods, which learn their predictions using different feature types such as random walks [16, 17], network similarity [18], nodes connecting paths [19] and subgraph paths [19, 20]. They are used in multiple biological predictive applications such as predicting drug targets [21] and protein–protein interaction analysis [18]. Despite the expressiveness of graph feature models predictions, they suffer from two major drawbacks: limited scalability and low accuracy [22, 23]. They are also focused on graph local features compared to embedding models, which learn global latent features of the processed graph.

Latent feature models i.e. embedding models, on the other hand, express knowledge graphs' entities and relations using low-rank vector representations that preserve the graph's global structure. Knowledge graph embedding (KGE) models on the contrary are known to outperform other approaches in terms of both the accuracy and scalability of their predictions despite their lack of expressiveness [23–25].

In recent years, KGE models witnessed rapid developments that allowed them to excel in the task of link prediction [24–30]. They have then been widely used in various applications including computational biology in tasks like predicting drug–target interactions (DTIs) [9] and predicting drug polypharmacy side effects [8]. Despite their high-accuracy predictions in different biological inference tasks, KGEs are in their early adoption stages in computational biology. Moreover, many computational biology studies that have used KGE models adopted old versions of these models [31, 32]. These versions have then received significant modifications through recent computer science research advances [25].

In a previous study, Su *et al.* [14] have introduced the use of network embedding methods in biomedical data science. The study compiles a taxonomy of embedding methods for both basic and heterogeneous networks where it discusses a broad range of potential applications and limitation. The study's objective was to introduce the broad range of network embedding methods; however, it lacked deeper investigation into the technical capabilities of the models and how can they be integrated with a specific biological problem. The study also did not compare the investigated models in terms of their accuracy and scalability, which is essential to assist reader from the biological domain to understand the key differences between these methods as to their applicability.

In this study, we exclusively explore KGE models, focusing on the best performing models in terms of both scalability and accuracy across various biological tasks. We use these case studies to demonstrate the analytical capabilities of KGE models, e.g. learning clusters and similarity measures in different biological problems. We also explore the process of building biological knowledge graphs for generic and specific biological inference tasks. We then present computer-based experimental evaluation of KGE models on different tasks such as predicting DTIs, drug polypharmacy side effects and prediction of tissue-specific protein functions.

The rest of this study is organized as follows: Section 2.1 discusses knowledge graphs as a data modelling technique and their applications in the biological domain. Section 2.2 discusses KGE models, their design and how they operate on different types of data. Section 3 presents the example case studies that we will use throughout the study. Section 4 discusses the predictive and analytical capabilities of KGE models on the designated case studies discussed in Section 3. Section 5 discusses the performance of KGE models on biological data in terms of the predictive accuracy and scalability. Section 6 discusses the current challenges and possible opportunities of the use of KGE models to model the different types of biological systems. Finally, we discuss our conclusions in Section 7.

Background

In this section, we discuss both knowledge graphs and KGE models in the context of biological applications.

Knowledge graphs

A knowledge graph is a data modelling technique that models linked data as a graph, where the graph's nodes represent data entities and its edges represent the relations between these entities. In recent years, knowledge graphs became a popular means for modelling relational data where they were adopted in various industrial and academic applications such as semantic search engines [33], question answering systems [34] and general knowledge repositories [35]. They were also used to model data from different types of domains such as general human knowledge [35], lexical information [36] and biological systems [12].

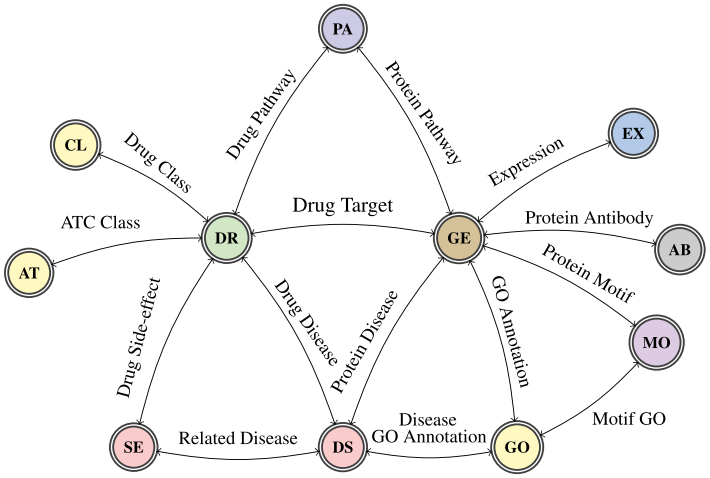


Figure 1. A schema of a knowledge graph that models a complex biological system of different types of entities and concepts. The abbreviation DR represents drugs, GE represents proteins (their genes), EX represents protein expressions (tissues and cell-lines), AB represents protein antibodies, MO represents protein motifs and other sequence annotations, GO represents gene ontology, DS represents diseases, SE represents drug side-effects, AT represents ATC classes, CL represents drug classes and PA represents pathways.

Knowledge graphs model facts as subject, predicate and object (SPO) triples, where subjects and objects are the knowledge entities and predicates are the knowledge relations. In this context, the subject entity is associated to the object entity with the predicate relation e.g. (*Aspirin*, *drug_target* and *COX1*). Figure 1 shows an illustration of a schema of a knowledge graph that models complex associations between different types of biological entities such as drugs, proteins, antibodies, etc. It also models different types of relations between these entities, where these relations carry different association semantics.

In our study, we use $\mathcal{G}$ to denote a knowledge graph, $\mathcal{E}$ to denote entities and $\mathcal{R}$ to denote relations i.e. predicates. We also use $\mathcal{N}_e$ and $\mathcal{N}_r$ to denote the total count of both entities and relations in a knowledge graph, respectively.

Popular Biological Sources. Online knowledge bases are a popular means for publishing large volumes of biological data [37]. In recent years, the number of these knowledge bases has grown, where they cover different types of data such as paper abstracts [38], raw experimental data [39], curated annotations [10, 40, 41], etc. Biological knowledge bases store data in different structured and unstructured (free text e.g. comments) forms. Although both data forms can be easily comprehended by humans, structured data are significantly easier for automated systems. In the following, we explore popular examples of these knowledge bases that offer structured data that can be easily and automatically consumed to generate knowledge graphs.

Table 1 summarizes the specializations and the different types of covered biological entities of a set of popular biological knowledge bases. The table also shows that most of the current knowledge bases are compiled around proteins (genes). However, it also shows their wide coverage of the different types of biological entities such as drugs, their indications, gene ontology annotations, etc.

Building Biological Knowledge Graphs. Knowledge graphs store information in a triplet form, where each triplet (i.e. triple) model a labelled association between two unique unambiguous entities. Data in biological knowledge bases, however, lack these association labels. Different knowledge bases also use different identifier systems for the same entity types, which results in the ambiguity of entities of merged databases. Building biological knowledge graph process therefore mainly deals with these two issues.

In the association labelling routine, one can use different techniques to provide meaningful labels for links between different biological entities. This, however, is commonly achieved by using entity types of both subject and object entities to denote the relation labels as shown in Figure 1 (e.g. 'Drug Side-effect' as a label for link between two entities that are known to be types of drug and side effect, respectively).

The ambiguity issue, i.e. merging entities of different identifier systems, is commonly resolved using identifier mapping resource files. Different systems study entities on different speciality levels. As a result, the links between their different identifier systems is not always in a form of one-to-one relationships. In such cases, a decision is made to apply a specific filtering strategy based on either expert's opinion or problem-specific properties (for instance, deciding on an authoritative resource such as UniProt for protein entities and resolving all conflicts by sticking to that resource's naming scheme and conventions).

To complement the basic principles introduced in the previous paragraphs, we refer the reader to the Bio2RDF initiative [55] that has extensively studied the general topic of building interlinked biological knowledge graphs [see also Bio2RDF scripts (https://github.com/bio2rdf/bio2rdf-scripts/wiki) for corresponding scripts and conversion convention details]. General principles as well as an example of actual implementation of conversion from (relational) databases into RDF (i.e. knowledge graphs) are discussed in the study of Bizer *et al.* [56]. Possible solutions to the problem of aligning and/or merging several such knowledge graphs are reviewed in the study of Amrouch *et al.* [57] that focuses on ontology matching. An example of a more data-oriented method is for instance LIMES [58]. All these approaches may provide a wealth of inspiration for building bespoke approaches to building knowledge graphs in specific biomedical use cases, should the information we provide in this section be insufficient.

Table 1. A comparison between popular biological knowledge graph in terms of the coverage of different types of biological entities. The abbreviation S represents structured data, U represents unstructured data, DR represents drugs, GE represents proteins, GO represents gene ontology,PA represents pathways and CH denotes chemicals

	Properties		Entity coverage								
Knowledge base	**Format**	**Speciality**	**Proteins**	**Drugs**	**Indications**	**Diseases**	**Gene ontology**	**Expressions**	**Antibodies**	**Phenotypes**	**Pathways**
UNIPROT [10]	S/U	GE	✓	✓		✓	✓	✓	✓		✓
REACTOME [42]	S	PA	✓				✓				✓
KEGG [40, 43]	S	PA	✓	✓		✓					✓
DrugBank [44]	S/U	DR	✓	✓							✓
Gene Ontology [11]	S	GO	✓				✓				✓
CTD [45]	S/U	CH	✓	✓			✓			✓	✓
ChEMBL [46]	S/U	CH	✓	✓	✓	✓		✓			
SIDER [47]	S	DR		✓	✓						
HPA [48]	S/U	GE	✓				✓	✓	✓		
STRING [49]	S	GE	✓								
BIOGRID [50]	S	GE	✓								
InAct [41]	S	GE	✓								
InterPro [51]	S	GE	✓								
PharmaGKB [52]	S	DR	✓	✓							
TTD [53]	S	DR	✓	✓							
Supertarget [54]	S	DR	✓	✓							

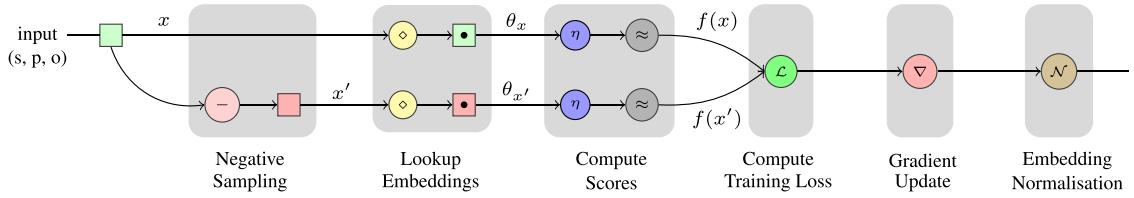


Figure 2. An illustration of the training network of one training instance of a KGE model.

Knowledge graph embeddings

In this section, we discuss KGE models where we briefly explore their learning procedure. We then explore different embedding representation types and their potential uses and application.

The learning procedure. Multiple studies have explored KGE models, their technical design, training objectives and predictive capabilities on general benchmarking settings [15, 24, 59]. Therefore, in the following, we only focus on providing a brief and concise description of how KGE models work.

KGE models operate by learning low-rank representations of knowledge graph entities and relations. The KGE learning step is a multi-phase procedure as shown in Figure 2, which is executed iteratively on knowledge graph data. Initially, all entities and relations are assigned random embeddings (noise). They are then updated using a multi-phase learning procedure.

KGE models consume knowledge graphs in the form of SPO triplets. They first generate negative samples from the input true triplets using uniform random corruptions of the subjects and objects [60]. KGE models then lookup corresponding embedding of both the true and corrupted triplets. The embeddings are then processed using model-dependent scoring functions (cf. mechanism of action in Table 2) to generate scores for all the triplets. The training loss is then computed using model-dependent loss functions where the objective is to maximize the scores of true triplets and minimize the scores of corrupted triplets. This objective can be formulated as follows:

$$\forall_{t\in\mathbb{T},t'\in\mathbb{T}'} f(\theta_t) > f(\theta_{t'}), \tag{1}$$

where $\mathbb{T}$ denotes the set of true triplets, $\mathbb{T}'$ denotes the set of corrupted triplets, f denotes the model-dependent scoring function and θ_t denotes the embeddings of the triplet t.

Traditionally, KGE models use a ranking loss, e.g. hinge loss or logistic loss, to model the objective training cost [26, 28, 29]. This strategy allows KGE models to efficiently train their embeddings in linear time, $\mathcal{O}(d)$, where K denotes the size of the embedding vectors. On the other hand, some KGE models such as the ConvE [30] and the ComplEx-N3 [25] models adopt multi-class based strategies to model their training loss. These approaches have shown superior predictive accuracy compared to traditional ranking-based loss strategies [25, 30]. However, they suffer from limited scalability as they operate on the full entity vocabulary.

The KGE models minimize their training loss using different variations of the gradient descent algorithm e.g. Adagrad, AMSGrad, etc. Finally, some KGE models normalize their embeddings as a regularization strategy to enhance their generalization. This strategy is often associated to models, which adopt ranking-based training loss strategies such as the TransE and DistMult models [26, 28].

The learning multi-phase procedure is executed iteratively to update the model's embeddings until they reach an optimal state that satisfies the condition in Equation 1. Table 2 also provides a summary of properties of popular KGE models, their mechanism of action i.e. scoring mechanism, output embeddings format, runtime complexity, release year and available code bases.

KGE models ingest graph data in triplets form where they learn global graph low-rank latent features, which preserve the

Table 2. A comparison between popular KGE models, their learning mechanism, published year and available code bases. Em. format column denotes the format of the model embeddings in the form $(g(d), h(d))$, where d denotes the embeddings size, $g(d)$ denotes the shape of the entities embeddings and $h(d)$ denotes the shape of the relations embeddings. n and m denote the number of entities and relations, respectively, in the space complexity column

Model	Scoring mechanism	Em. Format	Time complexity	Space complexity	Year	Repository (Python)
RESCAL [27]	Tensor factorization	(d, d^2)	$\mathcal{O}(d^2)$	$\mathcal{O}(nd + md^2)$	2011	mnick/rescal.py
TransE [26]	Linear translation	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2014	ttrouill/complex
DistMult [28]	Bilinear dot product	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2015	ttrouill/complex
HolE [62]	Fast Fourier transformation	(d, d)	$\mathcal{O}(d \log d)$	$\mathcal{O}(nd + md)$	2016	mnick/holographic-embeddings
ComplEx [29]	Complex product	$(2d, 2d)$	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2016	ttrouill/complex
ANALOGY [63]	Analogical structure	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2017	quark0/ANALOGY
ConvE [30]	Convolutional filters	(d, d)	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2018	TimDettmers/ConvE
TriModel [64]	Multi-part embeddings	$(3d, 3d)$	$\mathcal{O}(d)$	$\mathcal{O}(nd + md)$	2019	samehkamaleldin/libkge

graph's coherent structure. These features encode semantics such as node types and their neighbours by isolating nodes' embeddings on different embedding dimensions [23]. However, they have limited ability to encode indirect semantics such as logical rules and in-direct relations [61].

Embedding representation. KGEs have different formats e.g. vectors, matrices, etc., which serve as numerical feature representations of their respective objects. These representations can be used in both general tasks such as clustering and similarity analysis, as well as in specific inference tasks such as predicting different association types. Similarly, in computational biology, they can be used to cluster biological entities such as protein, drugs, etc., as well as to learn specific biological associations such as drug targets, gene-related diseases, etc. Embeddings of biological entities can also be used as representative features in traditional regression and classification models e.g. logistic regression or SVM classifiers.

Popular KGE models. Table 2 presents a comparison between a set of popular KGE models, their scoring mechanism, embeddings format, time complexity, space complexity, year of publication and corresponding source code repository. These models use different approaches to learn their embeddings where they can be categorized into three categories: distance-based models, factorization-based models and convolutional models. Distance-based models such as the TransE model use linear translations to model their embeddings interactions using a linear time and space complexity procedure. Convolution-based methods such as the ConvE use convolutional neural networks to model embedding interactions, which also have a linear time and space complexity. Factorization-based models, on the other hand, use dot product-based procedures to model embedding interactions, where they also have linear time and space complexity. However, tensor factorization-based models commonly use higher rank embeddings than convolution and distance-based models [29, 64].

In this study, we are focused on embedding methods, which operate on multi-relational graphs as we mentioned in the introduction of the paper. The DeepWalk [65], Node2Vec [66], etc. are uni-relational graphs embedding methods; thus, they we do not include them in this study.

Examples of biological case studies

In the following, we present two example biological case studies that we use through this study to demonstrate the capabilities of KGE models. Firstly, we discuss the task of predicting DTIs where we model biological information as a knowledge graph. We then evaluate the predictive accuracy of KGE models, and we compare them to other state-of-the-art approaches. Secondly, we discuss the task of predicting drug polypharmacy side effects, where we model the investigated drug polypharmacy data as a 3D tensor.

Predicting DTIs

The study of drug targets has become very popular with the objective of explaining mechanisms of actions of current drugs and their possible unknown off-target activities. Knowing targets of potential clinical significance also plays a crucial role in the process of rational drug development. With such knowledge, one can design candidate compounds targeting specific proteins to achieve intended therapeutic effects. Large-scale and reliable prediction of DTIs can substantially facilitate development of such new treatments. Various DTI prediction methods have been proposed to date. Examples include chemical genetic [67] and proteomic methods [68] such as affinity chromatography and expression cloning approaches. These, however, can only process a limited number of possible drugs and targets due to the dependency on laboratory experiments and available physical resources. Computational prediction approaches have therefore received a lot of attention lately as they can lead to much faster assessments of possible DTIs [69, 70].

Data. We consider the DrugBank_FDA [71] benchmarking data set as an example to evaluate the predictive accuracy of KGE models and to compare them to other approaches. We also utilize the UNIPROT [10] database to provide richer information about both drugs and their protein targets in the input knowledge graph. The data set contains 9881 known DTIs, which involve 1482 drugs and 1408 protein targets.

Related work. The work of Yamanishi *et al.* [69] was one of the 1st approaches to predict drug targets computationally. Their approach utilized a statistical model that infers drug targets based on a bipartite graph of both chemical and genomic information. The BLM-NII [70] model was developed to improve the previous approach by using neighbour-based interaction-profile inference for both drugs and targets. More recently, Cheng *et al.* [72, 73] proposed a new way for predicting DTIs, where they have used a combination of drug similarity, target similarity and network-based inference. The COSINE [74] and NRLMF [75] models introduced the exclusive use of drug–drug and target–target similarity measures to infer possible drug targets. This has an advantage of being able to compute predictions even for drugs and targets with limited information about their interaction data. However, these methods only utilized a single measure to model components similarity. Other approaches such as the KronRLS-MKL [76] model used a linear combination of multiple

similarity measures to model the overall similarity between drugs and targets. Non-linear combinations were also explored in an early study [70] and shown to provide better predictions. Recently, further predictive models were developed to utilize matrix factorization [77] and biological graph path features [7] to enable more accurate drug–target prediction.

Predicting polypharmacy side effects

Polypharmacy side effects are a specific case of adverse drug reactions that can cause significant clinical problems and represent a major challenge for public health and pharmaceutical industry [78]. Pharmacology profiling leads to identification of both intended (target) and unintended (off-target) drug-induced effects, i.e. biological system perturbations. While most of these effects are discovered during pre-clinical and clinical trials before a drug release on the market, some potentially serious adverse effects only become known when the drug is in use already.

When more drugs are used jointly (i.e. polypharmacy), the risk of adverse effects rises rather rapidly [79, 80]. Therefore, reliable automated predictions of such risks are highly desirable to mitigate their impact on patients.

Data. In this case study, we consider the data set compiled by Zitnik *et al.* [8] as an example benchmark. The data set includes information about multiple polypharmacy drug side effects (http://snap.stanford.edu/decagon/). The data set also contains facts about single drug side effects, protein–protein interactions and protein–drug targets. The drug side effects represented in the data set are collected from the SIDER (Side Effect Resource) database [47] and the OFFSIDES and TWOSIDES databases [80]. These side effects are categorized into two groups: mono-drug and polypharmacy drug–drug interaction side effects.

In our study, we only consider the polypharmacy side effects, and we filter out both the mono-side effects and drug targets data.

Related work. The research into predictive approaches for learning drug polypharmacy side effects is in its early stages [8]. The Decagon model [8] is one of the 1st introduced methods for predicting polypharmacy side effects, which models the polypharmacy side-effect data as a knowledge graph. It then solves the problem as a link prediction problem using a generative convolution-based strategy. Despite its effectiveness, this approach still suffers from a high rate of false positives. Furthermore, other approaches considered using a multi-source embedding model [81] to learn representations of drugs and polypharmacy side effects. These approaches achieved similar performance to the Decagon model with a more scalable training procedure [81].

Predicting tissue-specific protein functions

Proteins are usually expressed in specific tissues within the body where their precise interactions and biological functions are frequently dependent on their tissue context [82, 83]. The disorder of these interactions and functions results in diseases [84, 85]. Deep understanding of tissue-specific protein activities is therefore essential to elucidate the causes of diseases and possible treatments.

Data. We consider the tissue-specific data set compiled by Zitnik *et al.* [86] to study tissue-specific protein functions. The data set contains protein–protein interactions and protein functions of 144 tissue types (http://snap.stanford.edu/ohmnet/).

Related work. Recently, Zitnik *et al.* have developed the state-of-the-art model, the OhmNet model [86], a hierarchy-aware unsupervised learning method for multi-layer networks. It models each tissue information as a separate network and learns efficient representations for proteins and functions by generating their embeddings using the tissue-specific protein–protein interactome and protein functions. They have also examined other different approaches such as the LINE model [87], which uses a composite learning technique where it learns half of the embeddings' dimensions from the direct neighbour nodes and the other half from the 2nd hop connected neighbours. The GeneMania model [88] is another model that has suggested a propagation-based approach for predicting tissue-specific protein functions. In this method, the tissue-specific networks are firstly combined into one weighted network, and they are then propagated to allow predicting other unknown protein functions.

Capabilities of KGE models

KGE models can be used in different supervised and unsupervised applications where they provide efficient representations of biological concepts. They can be used in applications such as learning biological associations, concepts similarity and clustering biological entities. In this section, we discuss these applications in different computational biology tasks. We provide a set of example uses cases where we present the data integrated in each example, how the KGE models were utilized and we report the predictive accuracy of the KGE models and we compare it to other approaches when possible.

Learning biological associations

KGE models can process data in the form of a knowledge graph. They then try to learn low-rank representations of entities and relations in the graph, which preserve its coherent structure. They can also process data in a three-dimensional (3D) tensor form where they learn low-rank representations for the tensor entities that preserve true entity combination instances in the tensor.

In the following, we provide two examples for learning biological associations on a knowledge graph and a 3D tensor in a biological application. First, we discuss the task of predicting DTIs where we model biological information as a knowledge graph. We then evaluate the predictive accuracies of KGE models, and we compare them to other state-of-the-art approaches. Secondly, we discuss the task of predicting drug polypharmacy side effects, where we model the related data as a 3D tensor. We then apply KGE models to perform tensor factorization, and we evaluate their predictive accuracy in learning new polypharmacy side effects compared to other state-of-the-art approaches.

- Drug–target prediction benchmark. We present a comparison between state-of-the-art drug–target predictors and KGE models in predicting DTIs. The KGE models in this context utilize the fact that the current drug–target knowledge bases like DrugBank [71] and KEGG [40] are largely structured as networks representing information about drugs and their relationship with target proteins (or their genes), action pathways and targeted diseases. Such data can naturally be interpreted as a knowledge graph. The task of finding new associations between drugs and their targets can then be formulated as a link prediction problem on a biological knowledge graph.

  We use the standard evaluation protocol for the DTI task [7] on the DrugBank_FDA data set that we introduced in Section 3.1. We use a 5-fold cross-validation evaluation on the DTIs where they are divided into splits with uniform

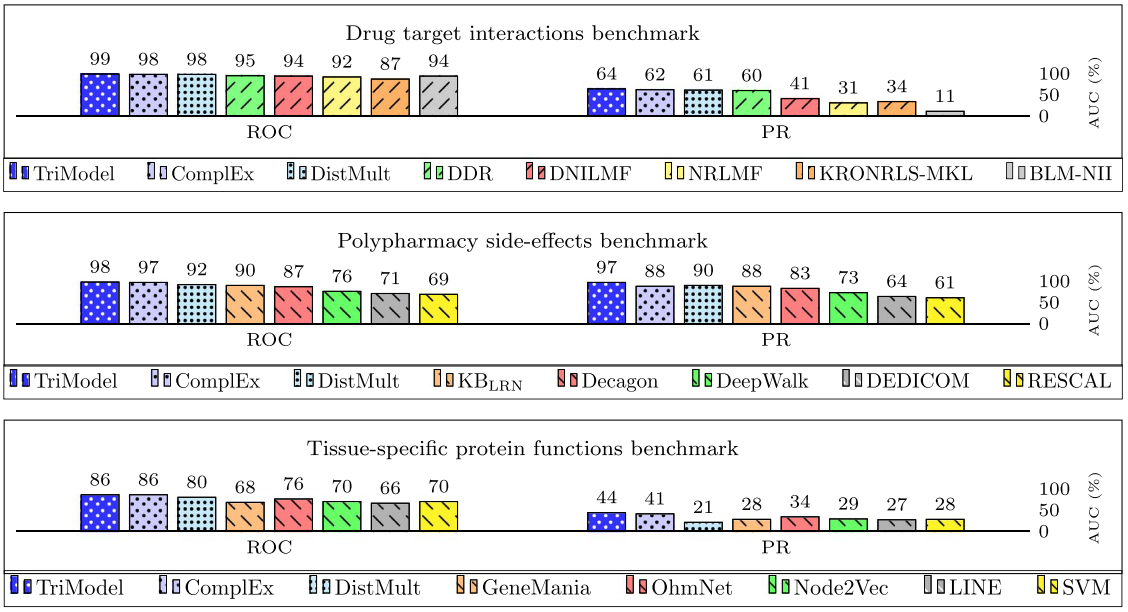


Figure 3. A summary of results of an evaluation of the predictive accuracy of knowledge graph embedding models compared to other models on two biological inference tasks: predicting drug targets and predicting polypharmacy side-effects. The reported results represent the score percentage of the area under the ROC and precision recall curves for the left and right side bars respectively.

random sampled negative instances with a 1:10 positive to negative ratio.

Figure 3 presents the outcome results of the KGE models (DistMult, ComplEx and TriModel) compared to other approaches (DDR [7], DNILMF [77], NRLMF [77], NRLMF [75], KRONRLS-MKL [76], COSINE [89] and BLM-NII [70]) on the DrugBank_FDA data set. The figure shows that the KGE models outperform all other approaches in terms of both the area under the ROC and precision recall curves.

- Polypharmacy side effects prediction benchmark. In Section 3.2, we discussed the problem of predicting polypharmacy side effects, the currently available data and related works. In the following, we present an evaluation benchmark for present polypharmacy side effects where we compare the KGE models with current state-of-the-art approaches. We first split the data into two sets, train and test splits, where the two splits represent 90% and 10% of the data, respectively. We then generate random negative polypharmacy side effects by randomly generating combinations of drugs for each polypharmacy side effect where the ratio between negative and positive instances is 1:1. We only consider drug combinations that did not appear in both training and test splits to enhance the quality of sampled negatives and decrease the ratio of false negatives.

  We use the holdout test defined by Zitnik *et al.* [8] where we train the predictive models on the training data and test their accuracy on the testing data split. We also run a 5-runs averaged 5-fold cross-validation evaluation to ensure the consistency of the model reported results over the different folds; however, we only report the holdout test results, which are comparable with state-of-the-art methods. Our k-fold cross validation experiments confirm that the model results are similar or insignificantly different across different random testing splits.

  We use the area under the ROC and precision recall metrics to assess the quality of the predicted scores. Figure 3 presents the results of our evaluation where we compare KGE models such as the DistMult, ComplEx and TriModel models to the current popular approaches (Decagon [8], KB_LRN [91], RESCAL [27], DEDICOM [92] and DeepWalk [65]). The results show that KGE models outperform other state-of-the-art approaches in terms of both the area under the ROC and precision recall curves.
- Tissue-specific protein function prediction benchmark. In Section 3.3, we have presented the problem of tissue-specific protein function prediction benchmark where we have discussed current predictive models and established benchmarking data sets. In the following, we present an evaluation benchmark between a set of traditional approaches such as the OhmNet [86], LINE [87], GeneMania [88] and SVM [86] models and other KGE models. We use the data set generated by Zitnik *et al.* [86], which provides training and testing data with both positive and negative instances where the negative to positive ratio is 1 to 10.

  We conduct a holdout test using the provided training and testing data set where we train our models on the training split and evaluate them on the testing using the area under the ROC and precision recall curves. Figure 3 presents the outcome of our experiments where it shows that KGE models such as the TriModel and ComplEx models achieve the best results in terms of both the area under the ROC and precision recall curves. Similar to the previous experiments, we also ran a 5-runs 5-fold cross-validation test to ensure the consistency of our results, and the results of our experiments confirm the results reported in the holdout test. However, we only report the holdout test results to be able to compare to other approaches.

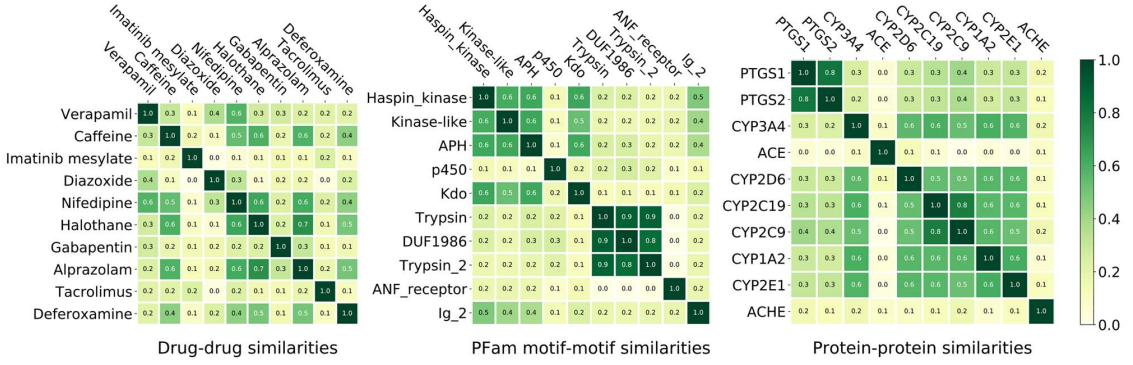


Figure 4. Three similarity matrices that denote the drug–drug similarities, motif–motif similarities and protein–protein similarities. The similarity values are generated by computing the cosine similarity between the embeddings of the pairs of compared entities. All the embeddings used to generate this figure are computed on the DrugBank_FDA data sets with the proteins associated to their PFam [90] motifs and protein families.

In all of our holdout test experiments, we learn the best hyperparameters using a grid search on the validation data split, where the training set is divided into two sets for training and validation (90% and 10%, respectively) in the absence of a validation set. On the other hand, in the cross validation experiments, we re-split each into training and validation splits (90% and 10%, respectively) in order to learn the model's best hyperparameters. We have found the embedding size is the most sensitive hyperparameters where it correlates with the graph size. The regulation weight and embedding dropout also are important hyperparameters, which affect the generality of the models from the validation to the testing split.

Example source code scripts and data sets of the experiments, which we executed in this study, are available at https://github.com/samehkamaleldin/bio-kge-apps.

Learning similarities between biological entities

The KGE models enable a new type of similarity that can be measured between any two biological entities using the similarity between their vector representation. The similarity between vectors can be computed using different techniques such as the cosine and p-norm similarities. Since the KGE representation is trained to preserve the knowledge graph structure, the similarity between two KGE representations reflects their similarity in the original knowledge. Therefore, the similarities between vector representations of KGE models, which are trained on a biological knowledge graphs, represent the similarities between corresponding entities in the original knowledge graph.

In the following, we explore a set of examples for using KGE similarities on biological knowledge graphs. We have used the drug–target knowledge graph created for the drug–target prediction task to learn embeddings of drugs, their target proteins and the entities of the motifs of these proteins according to the PFam database [90]. We have then computed the similarities between embeddings of entities of the same type such as drugs, proteins and motifs as shown in Figure 4. All the similarity scores in the illustration are computed using cosine similarity between the embeddings of the corresponding entity pair. The results show that the similarity scores are distributed from 0.0 to 1.0, where the 0.0 represents the least similar pairs and the 1.0 scores represent the similarity between the entity and itself. We then assess the validity of resulting scores by investigating the similarity of attributes of a set of the examined concepts with highest and lowest scores.

- Drug–drug embedding similarity. The left similarity matrix in Figure 4 illustrates the drug–drug similarity scores between the set of the most frequent drugs in the DrugBank_FDA data set. The scores are computed on the embeddings of drugs learnt in the DTI training pipeline. The figure shows that the majority of drug pairs have a low similarity (0.0 $\sim$ 0.2). For example, the similarity score between the drug pairs (diazoxideand caffeine) and (tacromlimus and diazoxide) is zero. We asses these results by assessing the commonalities between the investigated drugs in terms of indications, pharmacodynamics, mechanism of action, targets, enzymes, carriers and transporters. The caffeine and diazoxide in this context have no commonalities except for that they are both diuretics [93, 94]. On the other hand, halothane and alprazolam does not share any of the investigated commonalities.

  The results also shows a few drug–drug similarities with relatively higher scores (0.6 $\sim$ 0.7). For example, the similarity scores of the drug pairs (alprazolam and halothane), (alprazolam and caffeine) and (halothane and caffeine) are 0.7, 0.6 and 0.6, respectively. These findings can be supported by the fact that the two drug pairs share common attributes in terms of their targets, enzymes and carriers. For example, both alprazolam and halothane act on sedating individuals, and they target the GABRA1 protein [95, 96]. They are also broken by CYP3A4 and CYP2C9 enzymes and carried by albumin [97]. Similarly, the (alprazolamand caffeine) and (halothaneand caffeine) pairs have common associated enzymes.
- Motif–motif embedding similarity. The middle similarity matrix in Figure 4 illustrates the motif–motif similarity scores between the set of the most frequent PFam motifs associated with protein targets from the DTI benchmark. The lowest motif–motif KGE-based similarity scores correspond to the pairs (*ANF_receptor* and *Trypsin*), (*ANF_receptor* and *DUF1986*) and (*ANF_receptor* and *Trypsin_2*).
- On the other hand, the highest similarity scores (0.8, 0.9 and 0.9) exist between the pairs (*Trypsin*and *DUF1986*), (*Trypsin_2* and *DUF1986*) and (*Trypsin* and *Trypsin_2*), respectively.

  We assess the aforementioned findings by investigating the nature and activities of each of the discussed motifs. For example, *Trypsin* is a serine protease that breaks down proteins and cleaves peptide chains while *Trypsin_2* is

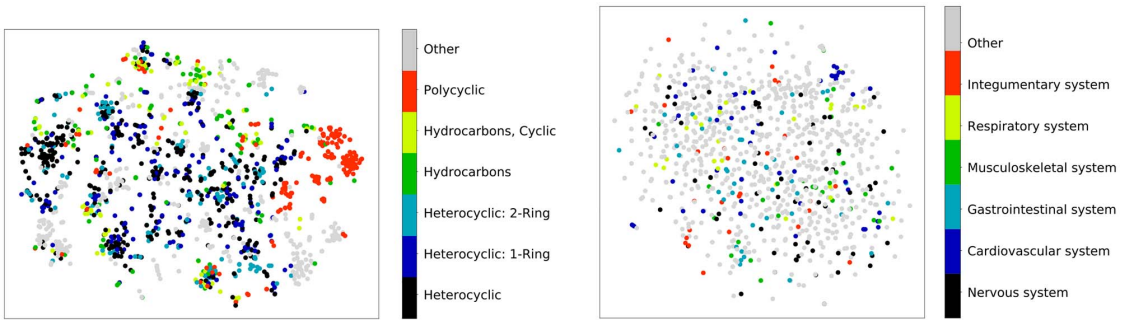


Figure 5. Three similarity matrices that denotes the drug–drug similarities, motif–motif similarities and protein–protein similarities. The similarity values are generated by computing the cosine similarity between the embeddings of the pairs of compared entities. All the embeddings used to generated this figure are computed on the DrugBank_FDA data sets with the proteins associated to their PFam [90] motifs and protein families.

an isozyme of *Trypsin*, which has a different amino acid sequence but catalyzes the same chemical reaction as *Trypsin* [98].

Moreover, the DUF1986 is a domain that is found in both of these motifs, which supports the high similarity scores. On the other hand, the *ANF receptor* is an atrial natriuretic factor receptor that binds to the receptor and causes the receptor to convert *GTP* to *cGMP*, and it plays a completely different role to trypsin, which supports its reported low similarity scores with trypsin.

- Protein–protein embedding similarity. The right similarity matrix in Figure 4 illustrates the protein–protein similarity scores between the set of the most frequent protein targets from the DTI benchmark. The highest-scored protein–protein pairs are (*PTGS1*, *PTGS2*) and (*CYP2C19*, *CYP2C9*) with the scores 0.8 and 0.8, respectively. This can be supported by the fact that the proteins CYP2C9, CYP1A2 and CYP2E1 belong to the same family of enzymes and thus they have similar roles.

  On the other hand, the *ACE* protein have the lowest similarity scores with the *CYP2C9*, *CYP1A2* and *CYP2E1* proteins with 0.0 similarity score. This can be supported by the fact that ACE is a hydrolase enzyme, which is completely different from *CYP2C9*, *CYP1A2* and *CYP2E1*, which are Oxidoreductases enzymes.

Clustering biological entities

In the following, we demonstrate the possible uses of embeddings based clustering in different biological tasks. We explore two cases where we use the embeddings of KGE models to generate clusters of biological entities such as drugs and polypharmacy side effects. We use visual clustering as an example to demonstrate cluster separation on a 2D space. However, in real scenarios, clustering algorithms utilize the full dimensionality of embedding vectors to build richer semantics of outcome clusters. Figure 5 shows two scatter plots of the embeddings of drugs from the DrugBank_FDA data set and the polypharmacy side effects reduced to a 2D space. We reduced the original embeddings using the T-SNE dimensionality reduction module [99] with the cosine distance configuration to reduce the embedding vectors to a 2D space.

The following examples examines two cases that differs in terms of the quality of generated clusters where we examine both drugs and polypharmacy side effects according to different properties. In the 1st example (drug clustering), the generated embeddings is able to provide efficient clustering. On the other hand, in the 2nd example, the polypharmacy side effects, the learnt embeddings could not be separated into visible clusters according to the investigated property.

- Clustering drugs. The left plot in Figure 5 shows a scatter plot of the reduced embedding vectors of drugs coloured according to their chemical structure properties. The drugs are annotated with seven different chemical structure annotations: *Polycyclic*, *Hydrocarbons Cyclic*, *Hydrocarbons*, *Heterocyclic*, *Heterocyclic 1-Ring*, *Heterocyclic 2-Ring* and other chemicals. These annotations represent the six most frequent drug chemical structure category annotation extracted from the DrugBank database.

  We can see in the plot that the *Polycyclic* chemicals are located within a distinguishable cluster in the right side of the plot. The plot also shows that other types of *Hydrocarbons* and *Heterocyclic* chemicals form different micro-clusters in different locations in the plot.

  These different clusters can be used to represent a form of similarity between the different drugs. It can also be used to examine the relation between the embeddings as a representation with the original attributes of the examined drugs.
- Clustering polypharmacy side effects. The right plot in Figure 5 shows a scatter plot of the reduced embedding vectors of polypharmacy side effects. The plot polypharmacy side-effect points are coloured according to the human body systems they affect. The plot includes a set of six categories of polypharmacy side effects that represent six different human body systems e.g. nervous system.

  Unlike the drug clusters illustrated in the left plot, the polypharmacy side-effect system-based categorization does not yield obvious clusters. They, however, form tiny and scattered groups across the plot. This shows that the KGE models are unable to learn representations that can easily separate polypharmacy side effects according to their associated body system.

Practical considerations for KGE models

In this section, we discuss different practical considerations related to the use of KGE models. We discuss their scalability

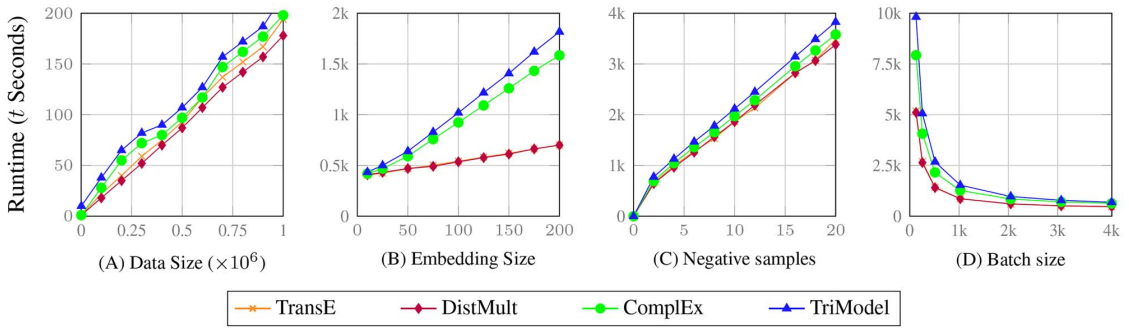


Figure 6. A set of line plots that describe the relation between the training runtime and the data size and configurable parameters of the TransE, DistMult, ComplEx and TriModel KGE models. The y-axis in all the plots represents the training time in seconds with different scales while the x-axis represents the data size and the models' parameters embedding size, negative samples and batch size respectively. The reported results are acquired by running the KGE models on the polypharmacy side-effects' full dataset (≈ 4.5M instances).

on different experimental configurations, and we explore their different training and implementation strategies.

Scalability

Not only KGE models outperform other approaches in biological knowledge graphs completion tasks but they also have better scalability compared to usual graph exploratory approaches. Often, complex biological systems are modelled as graphs where exploratory graph analytics methods are applied to perform different predictive tasks [5–7]. These models however suffer from limited scalability as they depend on graph traversal techniques that require complex training and predictions times [100, 101]. On the other hand, KGE models operate using linear time and space complexity [29, 59].

On the other hand, explanatory graph models use graph path searches, which require higher time and space complexity [22]. For example, the DDR model [21] is an exploratory graph drug–target predictor, which uses graph random walks as features. A recent study [102] has shown that KGE models can outperform such models with higher scalability and better predictive accuracy. This is due to their linear time and space complexity procedures [29] compared to other exploratory models, which use polynomial and exponential time and space procedures [23, 103].

In the following, we provide an empirical study of the scalability of KGE models in terms of different experimental configuration. We have studied the relation between the training runtime of KGE models and several training configuration parameters to examine their scalability capabilities. We have investigated the relation between the training runtime and the data size, embedding size, training negative samples and the training data batch size. We have performed our study on the polypharmacy side-effect data where the objective was to learn embeddings of drugs and polypharmacy side effects.

Figure 6 shows the outcome results of our study across the different investigated attributes. Plot 'A' shows the relation between the training runtime and the size of the processed data. The plot shows that all the four investigated have a linear relation between their training runtime and the investigated data size. The plot also shows that the investigated models have a consistent growth in terms of their runtime across all the data sizes. The DistMult model consistency achieves the smallest runtime followed by the TransE, ComplEx and TriModel models, respectively.

Plot 'B' shows the relationship between the training runtime and the model embedding size. The plot shows that all the investigated models have a linear growth of their training runtime corresponding to the growth of the embeddings size. However, the growth rate of the TransE and DistMult models is considerably smaller than the growth of both the ComplEx and TriModel models. This occurs as both the TransE and DistMult models use a single vector to represent each of their embeddings while the ComplEx and TriModel models use two and three vectors, respectively. Despite the better scalability of both the TransE and DistMult models, the ComplEx and TriModel models generally achieve better predictive accuracy than the TransE and DistMult models [64].

Plot 'C' shows the relation between the runtime of KGE models and the number of negative samples they use during training. The plot shows that there is a positive linear correlation between training runtime and the number of negative samples—where all the KGE models have similar results across all the investigated sampling sizes. The TriModel, however, consistently have the highest runtime compared to other models.

Plot 'D' shows the effects of the size of the batch on the training runtime. The plot shows an exponential decay of the training runtime with the linear growth of the data batch size. The KGE models process all the training data for each training iteration i.e. epoch, where the data are divided into batches for scalability and generalization purposes. Therefore, the increase of the training data batch sizes leads to a decrease of the number of model executions for each training iteration. Despite the high scalability that can be achieved with large batch sizes, the best predictive accuracy is often achieved using small data batch sizes. Usually, the most efficient training data batch size is chosen during a hyperparameter grid search along with other parameters such as the embedding size and the number of negative samples.

Implementation and training strategies

Different implementations of KGE models are available online in different repositories as shown in Table 2. The high scalability of KGE models allows them to be ported to both CPUs and GPUs where they can benefit from the high-performance capabilities of GPU cores. They can also be implemented to operate in a multi-machine design, where they perform embedding training in a distributed fashion [104]. This configuration is better suited

for processing knowledge graph of massive volumes that is hard to fit into one machine.

In this study, all our experiments are implemented in Python 3.5 using the Tensorflow library where we train our models on a single GPU card on one machine. We run our experiments on a Linux machine with an Intel(R) Core(TM) i7 processor, 32 GB RAM and an nVidia Titan Xp GPU.

Opportunities and challenges

In this section, we discuss the challenges and opportunities related to the general and biological applications of KGE models. We begin by discussing the scope of input data for these models. We then discuss possible applications of KGE models in the biological domain. We conclude by discussing the limited interpretability of KGE models and other general limitations related to their biological applications.

Potential applications

KGE models can build efficient representations of biological data, which are modelled as 3D tensors or knowledge graphs. This includes multiple types of biological data such as protein interactome and DTIs. In the following, we discuss examples of biological tasks and applications that can be performed using KGE models.

- Modelling proteomics data. KGE models can be used to model the different types of protein–protein interactions such as binding, phosphorylation, etc. [105, 106]. This can be achieved by modelling these interactions as a knowledge graphs and applying the KGE models to learn the embeddings of the different proteins and interaction types. They can also be used to model the tissue context of interactions where different body tissues have different expression profiles of proteins, and these differences in expression affect the proteins' interaction network. KGE can be used to model these interactions with their associated contexts as tensors [6].

  The biological activities of proteins also differ depending on their tissue context [86]. This type of information can easily be modelled using tensors where KGE models can be used to analyse the different functions of proteins depending on their tissue context [107].
- Modelling genomics data. Genomics data have been widely used to predict multiple gene associated biological entities such as gene–disease and gene–function associations [108, 109]. These approaches model the gene association in different ways including tensors and graph-based representations [110]. KGE models can be easily utilized to process such data and provide efficient representations of genes and their associated biological objects. They can be further used to analyse and predict new disease–gene and gene–function associations.
- Modelling pharmacological systems. Information on pharmaceutical chemical substances is becoming widely available on different knowledge bases [46, 71]. This information includes the drug–drug and drug–protein interactome. In this context, KGE models can be a natural fit, where they can be used to model and extend the current pharmacological knowledge. They can also be used to model and predict both traditional and polypharmacy side effects of drugs as shown in recent works [8, 111].

More details and discussion of the possible uses of KGE models and other general network embedding methods can be found in the study of Su *et al.* [14], which discusses further potential uses of these methods in the biological domain.

Limitations of the KGE models

In the following, we discuss the limitations of the KGE models in both general and biological applications.

- Lack of interpretability. In KGE models, the learning objective is to model nodes and edges of the graph using low-rank vector embeddings that preserve the graph's coherent structure. The embedding learning procedure operates mainly by transforming noise vectors to useful embeddings using gradient decent optimization on a specific objective loss. Despite the high accuracy and scalability of this procedure, these models work as a black box and they are hard to interpret. Some approaches have suggested enhancing the interpretability of KGE models by using constraining training with a set of predefined rules such as type constraints [112], basic relation axioms [113], etc. These approaches thus enforce the KGE models to learn embeddings that can be partially interpretable by their employed constraints.

  In recent studies, researchers have also explored the interpretability of KGE models through new predictive approaches on top of the KGE models. For example, Gusmão *et al.* [114] suggested the use of pedagogical approaches where they have used an alternative graphical predictive model, the SFE model [19], to link the learnt graph embeddings to the original knowledge graph. This approach was able to provide a new way for finding links between the embeddings and the original knowledge; however, the outcomes of these methods are still limited by the expressibility and feature coverage of the newly employed predictive models. The interpreting method in this context also depends on graph traversal methods, which have limited scalability on large knowledge graphs [20].
- Data quality. KGE models generate vector representations of biological entities according to their prior knowledge. Therefore, the quality of this knowledge affects the quality of the generated embeddings. For example, there is a high variance in the available prior knowledge on proteins where well-studied proteins have significantly higher coverage in most databases [115]. This has a significant impact on quality of the less represented proteins as KGE models will be biassed towards more studied proteins (i.e. highly covered proteins).

  In recent years, multiple works have explored the quality of currently available knowledge graphs [116] and the effect of low quality graphs on embedding models [117]. These works have shown that the accuracy KGE predictions degrade as sparsity and unreliability increase [117].

  This issue can be addressed by extending the available knowledge graph facts through merging knowledge bases of similar content. For example, drug–target prediction using KGE models can be enhanced by extending the knowledge of protein–drug interactions by extra information such as protein–protein interactions and drug properties [102].
- Knowledge evolution. Biological knowledge evolves everyday, where new chemicals and drugs are introduced and different associations between biological entities are discovered. However, KGE models in this context are unable to encode the newly introduced entities. This results from their

dependence on prior knowledge instead of the structural informations of proteins and chemical substances.

This issue can be addressed by combining KGE scoring procedure with other sequence- and structure-based scoring mechanisms. This can allow informed prediction on new unknown objects. However, such a strategy will affect the scalability of predictions due to the newly introduced sequence- and structure-based features.

- Hyperparameter sensitivity. The outcome predictive accuracy of KGE embeddings is sensitive to their hyperparameters [118]. Therefore, minor changes in these parameters can have significant effects on the outcome predictive accuracy of KGE models. The process of finding the optimal parameters of KGE models is traditionally achieved through an exhausting brute-force parameter search. As a result, their training may require rather time-consuming grid search procedure to find the right parameters for each new dataset.

  In this regard, new strategies for hyperparameter tuning such as differential evolution [119], random searches [120] and Bayesian hyperparameter optimization [121]. These strategies can yield a more informed parameter search results with less running time.
- Reflecting complex semantics of biological data in models based on knowledge graphs. KGE methods are powerful in encoding direct links between entities; however, they have limited ability in encoding simple indirect semantics such as types at different abstraction levels (i.e. taxonomies). For example, a KGE model can be very useful in encoding networks of interconnecting proteins, which are modelled using direct relations. However, it has limited ability in encoding compound, multi-level relationships such as protein involvement in diseases due to their involvement in pathways that cause this disease. Such compound relationships that could be used for modelling complex biological knowledge are notoriously hard to reflect in KGE models [122]. However, the KGE models do have some limited ability to encode for instance type constraints [123], basic triangular rules [122] or cardinality constraints [124]. This could be used for modelling complex semantic features reflecting biological knowledge in future works. One has to bear in mind, though, that the designs of these semantics-enhanced KGE models typically depends on an extra computational routines to regularize the learning process, which affects their scalability.

In their study, Su *et al.* [14] have also discussed further general limitations of network embedding methods and the effects and consequences of such limitations on the use of network embedding methods in the biological domain.

Conclusions

In this study, we discussed KGE models and their biological applications. We presented two biological case studies, predicting drug targets and predicting polypharmacy side-effects, to demonstrate the predictive and analytical capabilities of KGE models. We demonstrated by computational experimental evaluation that KGE models outperform state-of-the-art approaches in solving the two studied problems on standard benchmarks. We also demonstrated the analytical capabilities of KGE such as clustering and measuring concept similarities. In this regard, we demonstrated KGE models' abilities to learn efficient similarities between different biological entities such as drugs and proteins. We also showed that the KGE models can efficiently be used as clustering methods for biological entities.

Furthermore, we discussed different practical considerations regarding the scalability and training strategies of KGE models. We also discussed the potential applications of KGE models in the biological domain. We finally discussed the challenges and limitations that face KGE models where we explored both their general limitations and the challenges that face them in the biological domain. In conclusion, we believe that the presented study can be a solid stepping stone towards many promising applications of the emergent KGE technology in the field of computational biology.

Key Points

- Knowledge graphs allow easy, automated integration of multiple diverse biological data sets in order to model complex biological systems.
- KGE models enable scalable and efficient predictions on biological knowledge graphs.
- KGE models provide state-of-the-art predictive accuracy in learning biological associations with high scalability.
- KGE models provide high-quality analytics, e.g. clustering and concept similarities, of complex biological systems that can be modelled as graphs or 3D tensors.
- KGE models can be utilized to model and analyse different types of biological data including genomics, proteomics and pharmacological data.
- Despite their accurate and scalable predictive capabilities, however, KGE models have limited interpretability. They are also sensitive to data quality, knowledge evolution and training configurations.

Funding

The work presented in this paper was supported by the CLARIFY project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 875160, and by Insight research centre supported by the Science Foundation Ireland (SFI) grant (12/RC/2289_2).

Conflict of interest

None.

References

1. Cohen JD, Servan-Schreiber D. Context, cortex, and dopamine: a connectionist approach to behavior and biology in schizophrenia. *Psychol Rev* 1992;**99**(1):45–77.
2. Gibrat J-F, Madej T, Bryant SH. Surprising similarities in structure comparison. *Curr Opin Struct Biol* 1996;**6**(3):377–85.
3. Barabási A-L, Oltvai ZN. Network biology: understanding the cell's functional organization. *Nat Rev Genet* 2004;**5**:101–13.
4. Albert R. Scale-free networks in cell biology. *J Cell Sci* 2005;**118**(Pt 21):4947–57.
5. Janjic V, Przulj N. Biological function through network topology: a survey of the human diseasome. *Brief Funct Genomics* 2012;**11**(6):522–32.
6. Muñoz E, Novácek V, Vandenbussche P-Y. Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. *Brief Bioinform* 2019;**20**(1): 190–202.

7. Olayan RS, Ashoor H, Bajic VB. Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics* 2017;**34**(7):1164–73.
8. Zitnik M, Agrawal M, Leskovec J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 2018;**34**(13): i457–66 .
9. Mohamed SK, Nováček V, Nounu A. Drug target discovery using knowledge graph embeddings. In: *Proceedings of the 34th Annual ACM Symposium on Applied Computing, SAC '19*, pp. 11–18. Limassol, Cyprus: ACM, 2019.
10. The UniProt Consortium. Uniprot: the universal protein knowledgebase. *Nucleic Acids Res* 2017;**45**(D1): D158–69.
11. The Gene Ontology Consortium. The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res* 2019;**47**(D1): D330–8.
12. Dumontier M, Callahan A, Cruz-Toledo J, *et al.* Bio2rdf release 3: a larger, more connected network of linked data for the life sciences. In: *Proceedings of the ISWC 2014 Posters & Demonstrations*, pp. 401–4. Riva del Garda, Italy: CEUR-WS.org, 2014.
13. Alshahrani M, Khan MA, Maddouri O, *et al.* Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics* 2017;**33**(17): 2723–30.
14. Su C, Tong J, Zhu Y, *et al.* Network embedding in biomedical data science. *Brief Bioinform* 2018. doi: 10.1093/bib/bby117
15. Nickel M, Murphy K, Tresp V, *et al.* A review of relational machine learning for knowledge graphs. *Proc IEEE* 2016;**104**(1):11–33.
16. Lao N, Mitchell TM, Cohen WW. Random walk inference and learning in a large scale knowledge base. In: *EMNLP*, Edinburgh, UK: ACL, 2011.
17. Xu B, Guan J, Wang Y, *et al.* Essential protein detection by random walk on weighted protein-protein interaction networks. *IEEE/ACM Trans Comput Biol Bioinform* 2017;**16**: 377–87.
18. Raman K. Construction and analysis of protein-protein interaction networks. *Autom Exp* 2010;**2**;12.
19. Gardner M, Mitchell TM. Efficient and expressive knowledge base completion using subgraph feature extraction. In: *EMNLP*, pp. 1488–98. Lisbon, Portugal: The Association for Computational Linguistics, 2015.
20. Mohamed SK, Nováček V, Vandenbussche P-Y. Knowledge base completion using distinct subgraph paths. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC '18*, pp. 1992–9. Pau, France: ACM, 2018.
21. Olayan RS, Ashoor H, Bajic VB. DDR: efficient computational method to predict drug-target interactions using graph mining and machine learning approaches. *Bioinformatics* 2018;**34**(7):1164–73.
22. 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*, pp. 57–66. Beijing, China: ACL, 2015.
23. Nickel M, Murphy K, Tresp V, *et al.* A review of relational machine learning for knowledge graphs. *Proc IEEE* 2016;**104**(1):11–33.
24. Wang Q, Mao Z, Wang B, *et al.* Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans Knowl Data Eng* 2017;**29**(12):2724–43.
25. Lacroix T, Usunier N, Obozinski G. Canonical tensor decomposition for knowledge base completion. In: *ICML*, pp. 2869–78. JMLR Workshop and Conference Proceedings, Vol. 80. Stockholm, Sweden: JMLR.org, 2018.
26. Bordes A, Usunier N, García-Durán A, *et al.* Translating embeddings for modeling multi-relational data. In: *NIPS*, 2013, pp. 2787–95. Lake Tahoe, Nevada, United States: NIPS.
27. Nickel M, Tresp V, Kriegel H-P. A three-way model for collective learning on multi-relational data. In: *ICML*, pp. 809–16. Bellevue, Washington, USA: Omnipress, 2011.
28. Yang B, Yih W-T, He X, *et al.* Embedding entities and relations for learning and inference in knowledge bases. In: *ICLR*, San Diego, CA, USA: ICLR, 2015.
29. Trouillon T, Welbl J, Riedel S, *et al.* Complex embeddings for simple link prediction. In: *ICML*, pp. 2071–80. JMLR Workshop and Conference Proceedings, Vol. 48. New York City, NY, USA: JMLR.org, 2016.
30. Dettmers T, Pasquale M, Pontus S, *et al.* Convolutional 2d knowledge graph embeddings. In: *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA: AAAI Press, 2018.
31. Zitnik M, Zupan B. Collective pairwise classification for multi-way analysis of disease and drug data. *Pac Symp Biocomput* 2016;**21**:81–92.
32. Abdelaziz I, Fokoue A, Hassanzadeh O, *et al.* Large-scale structural and textual similarity-based mining of knowledge graph to predict drug-drug interactions. *J Web Semant* 2017;**44**:104–17.
33. Qian R. Understand your world with bing, 2013. Bing Blogs.
34. Ferrucci DA, Brown EW, Chu-Carroll J, *et al.* Building Watson: an overview of the deepqa project. *AI Magazine* 2010;**31**(3):59–79.
35. Mitchell TM, Cohen WW, Hruschka ER, Jr, *et al.* Never-ending learning. In: *AAAI*, pp. 2302–10. New Orleans, Louisiana, USA: AAAI Press, 2015.
36. Miller GA. Wordnet: a lexical database for english. *Commun ACM* 1995;**38**(11):39–41.
37. Zhu Y, Elemento O, Pathak J, *et al.* Drug knowledge bases and their applications in biomedical informatics research. *Brief Bioinform* 2019;**20**(4): 1308–21.
38. Aronson AR, Mork JG, Gay CW, *et al.* The nlm indexing initiative's medical text indexer. *Stud Health Technol Informatics* 2004;**107**(Pt. 1):268–72.
39. Landrum MJ, Lee JM, Riley GR, *et al.* Clinvar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res* 2014;**42**(D1): D980–5.
40. Kanehisa M, Furumichi M, Tanabe M, *et al.* Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res* 2017;**45**(D1):D353–61.
41. Orchard SE, Ammari MG, Aranda B, *et al.* The mintact project intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res* 2014;**42**(D1): 358–63.
42. Fabregat A, Jupe S, Matthews L, *et al.* The reactome pathway knowledgebase. *Nucleic Acids Res* 2018;**46**(D1): D649–55.
43. Kanehisa M, Sato Y, Kawashima M, *et al.* Kegg as a reference resource for gene and protein annotation. *Nucleic Acids Res* 2016;**44**(D1):D457–62.
44. Wishart DS, Knox C, Guo AC, *et al.* Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res* 2008;**36**:D901–6.
45. Mattingly CJ, Colby GT, Forrest JN, *et al.* The comparative toxicogenomics database (CTD). *Environ Health Perspect* 2003;**111**:793–5.
46. Gaulton A, Hersey A, Nowotka M, *et al.* The chembl database in 2017. *Nucleic Acids Res* 2017;**45**(D1):D945–54.

47. Kuhn M, Letunic I, Jensen LJ, *et al.* The sider database of drugs and side effects. *Nucleic Acids Res* 2016;**44**(D1): D1075–9.
48. Uhlén M, Fagerberg L, Hallstrom BM, *et al.* Tissue-based map of the human proteome. *Science* 2015;**347**(6220):1260419.
49. Szklarczyk D, Morris JH, Cook HV, *et al.* The string database in 2017: quality-controlled protein-protein association networks, made broadly accessible. *Nucleic Acids Res* 2017;**45**(D1): D362–8.
50. Stark C, Breitkreutz B-J, Chatr aryamontri A, *et al.* The BioGRID interaction database: 2011 update. *Nucleic Acids Res* 2007;**39**:D698–704.
51. Mitchell AL, Attwood TK, Babbitt PC, *et al.* Interpro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Res* 2019;**47**(D1): D351–60.
52. Hewett M, Oliver DE, Rubin DL, *et al.* Pharmgkb: the pharmacogenetics knowledge base. *Nucleic Acids Res* 2002;**30**(1):163–5.
53. Chen X, Ji ZL, Chen YZ. TTD: therapeutic target database. *Nucleic Acids Res* 2002;**30**(1):412–5.
54. Hecker N, Ahmed J, von Eichborn J, *et al.* Supertarget goes quantitative: update on drug-target interactions. *Nucleic Acids Res* 2012;**40**(D1): 1113–7.
55. Belleau F, Nolin M-A, Tourigny N, *et al.* Bio2RDF: towards a mashup to build bioinformatics knowledge systems. *J Biomed Inform* 2008;**41**(5):706–16.
56. Bizer C, Cyganiak R. D2R server-publishing relational databases on the semantic web. In: *Poster at the 5th International Semantic Web Conference*, Vol. 175, 2006.
57. Amrouch S, Mostefai S. Survey on the literature of ontology mapping, alignment and merging. In: *2012 International Conference on Information Technology and e-Services*, pp. 1–5. Sousse, Tunisia: IEEE, 2012.
58. Ngomo A-CN, Auer S. Limes—a time-efficient approach for large-scale link discovery on the web of data. In: *Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, Spain: IJCAI, 2011.
59. Mohamed SK, Muñoz E, Nováček V, *et al.* Loss functions in knowledge graph embedding models. In: *DL4KGS@ESWC*. CEUR Workshop Proceedings, Vol. 2106. Portoroz, Slovenia: CEUR-WS.org, 2019.
60. Bordes A, Glorot X, Weston J, *et al.* A semantic matching energy function for learning with multi-relational data—application to word-sense disambiguation. *Mach Learn* 2014;**94**(2):233–59.
61. Guo S, Wang Q, Wang L, *et al.* Jointly embedding knowledge graphs and logical rules. In: *EMNLP*, Austin, Texas, USA: ACL, 2016.
62. Nickel M, Rosasco L, Poggio TA. Holographic embeddings of knowledge graphs. In: *AAAI*, pp. 1955–61. Phoenix, Arizona USA: AAAI Press, 2016.
63. Liu H, Wu Y, Yang Y. Analogical inference for multi-relational embeddings. In: *ICML*, Sydney, Australia: ICML, 2017.
64. Mohamed SK, Novácek V. Link prediction using multi part embeddings. In: *ESWC*, pp. 240–54. Lecture Notes in Computer Science, Vol. 11503. Springer, 2019.
65. Perozzi B, Al-Rfou' R, Skiena S. Deepwalk: online learning of social representations. In: *SIGKDD*. 701–10. New York, USA: ACM, New York, 2014.
66. Grover A, Leskovec J. node2vec: scalable feature learning for networks. *KDD: Proceedings International Conference on Knowledge Discovery & Data Mining* 2016;**2016**: 855–64.
67. Terstappen GC, Schlüpen C, Raggiaschi R, *et al.* Target deconvolution strategies in drug discovery. *Nat Rev Drug Discov* 2007;**6**(11):891.
68. Sleno L, Emili A. Proteomic methods for drug target discovery. *Curr Opin Chem Biol* 2008;**12**(1):46–54.
69. Yamanishi Y, Araki M, Gutteridge A, *et al.* Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 2008;**24**(13):i232–40.
70. Mei J-P, Kwoh C-K, Yang P, *et al.* Drug–target interaction prediction by learning from local information and neighbors. *Bioinformatics* 2012;**29**(2):238–45.
71. Wishart DS, Knox C, Guo AC, *et al.* Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 2006;**34**:D668–72.
72. Cheng F, Zhou Y, Li W, *et al.* Prediction of chemical-protein interactions network with weighted network-based inference method. *PLoS One* 2012;**7**(7): e41064. https://doi.org/10.1371/journal.pone.0041064.
73. Cheng F, Liu C, Jiang J, *et al.* Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS Comput Biol* 2012;**8**(5): e1002503. https://doi.org/10.1371/journal.pcbi.1002503.
74. Rosdah AA, Holien JK, Delbridge LMD, *et al.* Mitochondrial fission–a drug target for cytoprotection or cytodestruction? *Pharmacol Res Perspect* 2016;**4**(3):e00235.
75. Liu H, Sun J, Guan J, *et al.* Improving compound–protein interaction prediction by building up highly credible negative samples. *Bioinformatics* 2015;**31**(12):i221–9.
76. Nascimento ACA, Prudêncio RBC, Costa IG. A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinform* 2016;**17**(1):46.
77. Hao M, Bryant SH, Wang Y. Predicting drug-target interactions by dual-network integrated logistic matrix factorization. *Sci Rep* 2017;**7**:40376.
78. Bowes J, Brown AJ, Hamon J, *et al.* Reducing safety-related drug attrition: the use of in vitro pharmacological profiling. *Nat Rev Drug Discov* 2012;**11**(12):909–22.
79. Kantor ED, Rehm CD, Haas JS, *et al.* Trends in prescription drug use among adults in the United States from 1999-2012. *JAMA* 2015;**314**(17):1818–31.
80. Tatonetti NP, Ye P, Daneshjou R, *et al.* Data-driven prediction of drug effects and interactions. *Sci Transl Med* 2012;**4**(125):125ra31.
81. García-Durán A, Niepert M. Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In: *UAI*, Monterey, California, USA: AUAI Press, 2018.
82. Fagerberg L, Hallström BM, Oksvold P, *et al.* Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. *Mol Cell Proteomics* 2014;**13**(2):397–406.
83. Greene CS, Krishnan A, Wong AK, *et al.* Understanding multicellular function and disease with human tissue-specific networks. *Nat Genet* 2015;**47**(6):569.
84. D'Agati VD. The spectrum of focal segmental glomerulosclerosis: new insights. *Curr Opin Nephrol Hypertens* 2008;**17**(3):271–81.
85. Cai JJ, Petrov DA. Relaxed purifying selection and possibly high rate of adaptation in primate lineage-specific genes. *Genome Biol Evol* 2010;**2**:393–409.

86. Zitnik M, Leskovec J. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* 2017;**33**(14): i190–8.
87. Tang J, Qu M, Wang M, *et al.* Line: large-scale information network embedding. In WWW, Florence, Italy: ACM, 2015.
88. Warde-Farley D, Donaldson SL, Comes O, *et al.* The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Res* 2010;**38**(Web-Server-Issue): 214–20.
89. Lim H, Gray P, Xie L, *et al.* Improved genome-scale multi-target virtual screening via a novel collaborative filtering approach to cold-start problem. *Sci Rep* 2016;**6**:38860.
90. Bateman A, Coin LJM, Durbin R, *et al.* The pfam protein families database. *Nucleic Acids Res* 2000;**28**(1):263–6.
91. Malone B, García-Durán A, Niepert M. Knowledge graph completion to predict polypharmacy side effects. In: *DILS*, Hannover, Germany: Springer, 2018.
92. Papalexakis EE, Faloutsos C, Sidiropoulos ND. Tensors for data mining and data fusion: models, applications, and scalable algorithms. *ACM Trans Intell Syst Technol* 2016;**8**:16:1–16:44.
93. Lipschitz WL, Hadidian Z, Kerpcsar A. Bioassay of diuretics. *Pharmacol Exp Ther* 1943, **79**(2):97–110.
94. Pohl JE, Thurston HF, Swales JD. The antidiuretic action of diazoxide. *Clinical Science* 1972, **42**(2):145–52.
95. Verster JC, Volkerts ER. Clinical pharmacology, clinical efficacy, and behavioral toxicity of alprazolam: a review of the literature. *CNS Drug Rev* 2004;**10**(1):45–76.
96. Overington JP, Al-Lazikani B, Hopkins AL. How many drug targets are there? *Nat Rev Drug Discov* 2006;**5**:993–6.
97. Minoda Y, Kharasch ED. Halothane-dependent lipid peroxidation in human liver microsomes is catalyzed by cytochrome P4502A6 (CYP2A6). *Anesthesiology* 2001;**95**(2): 509–14.
98. Rungruangsak-Torrissen K, Carter CG, Sundby A, *et al.* Maintenance ration, protein synthesis capacity, plasma insulin and growth of Atlantic salmon (salmo Salar L.) with genetically different trypsin isozymes. *Fish Physiol Biochem* 1999;**21**:223–33.
99. van der Maaten L. Accelerating t-sne using tree-based algorithms. *J Mach Learn Res* 2014;**15**:3221–45.
100. Cheung T-Y. Graph traversal techniques and the maximum flow problem in distributed computation. *IEEE Trans Softw Eng* 1983;**4**:504–12.
101. Fraigniaud P, Gasieniec L, Kowalski DR, *et al.* Collective tree exploration. *Network* 2006;**48**(3):166–77.
102. Mohamed SK, Novácek V, Nounu A. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics* 2019;**36**(2): 603–10.
103. Mohamed SK, Muñoz E, Novácek V, *et al.* Identifying equivalent relation paths in knowledge graphs. In: *LDK*, Galway, Ireland: Springer, 2017.
104. Lerer A, Ledell W, JS, *et al.* Pytorch-biggraph: a large-scale graph embedding system. In: *The 2nd SysML Conference*, Palo Alto, CA, USA: ACM, 2019.
105. Tuncbag N, Kar G, Keskin O, *et al.* A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. *Brief Bioinform* 2008;**10**(3):217–32.
106. Zhang J, Kurgan LA. Review and comparative assessment of sequence-based predictors of protein-binding residues. *Brief Bioinform* 2018;**19**:821–37.
107. Mohamed SK. Predicting tissue-specific protein functions using multi-part tensor decomposition. *Inform Sci* 2020;**508**:343–57.
108. Bamshad MJ, Ng SB, Bigham AW, *et al.* Exome sequencing as a tool for Mendelian disease gene discovery. *Nat Rev Genet* 2011;**12**:745–55.
109. Zeng X, Ding N, Rodríguez-Patón A, *et al.* Probability-based collaborative filtering model for predicting gene-disease associations. *BMC Med Genomics* 2017;**10**:76. https://doi.org/10.1186/s12920-017-0313-y.
110. Bauer-Mehren A, Bundschus M, Rautschka M, *et al.* Gene-disease network analysis reveals functional modules in mendelian, complex and environmental diseases. *PLoS One* 2011;**6**(6):e20284. doi: 10.1371/journal.pone.0020284. Epub 2011 Jun 14.
111. Muñoz E, Novácek V, Vandenbussche P-Y. Using drug similarities for discovery of possible adverse reactions. In: *AMIA 2016*, American Medical Informatics Association Annual Symposium, Chicago, IL, USA, November 12–16, 2016. Chicago, IL, USA: AMIA, 2016.
112. Krompass D, Baier S, Tresp V. *Type-constrained representation learning in knowledge graphs*. In: *International Semantic Web Conference*, Bethlehem, PA, USA: Springer, 2015.
113. Minervini P, Costabello L, Muñoz E, *et al.* Regularizing knowledge graph embeddings via equivalence and inversion axioms. In: *ECML/PKDD*, Skopje, Macedonia: Springer, 2017.
114. Gusmão AC, Correia AHC, De Bona G, *et al.* Interpreting embedding models of knowledge bases: a pedagogical approach. In: *Proceedings of WHI*, Stockholm, Sweden: CoRR, 2018.
115. The Uniprot Consortium. Uniprot: a hub for protein information. *Nucleic Acids Res* 2015;**43**(D1): 204–12.
116. Färber M, Bartscherer F, Menne C, *et al.* Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web* 2017;**9**:77–129.
117. Pujara J, Augustine E, Getoor L. Sparsity and noise: where knowledge graph embeddings fall short. In: *EMNLP*, Copenhagen, Denmark: ACL, 2017.
118. Kadlec R, Bajgar O, Kleindienst J. Knowledge base completion: Baselines strike back. In: *Rep4NLP@ACL*, pp. 69–74. Vancouver, Canada: Association for Computational Linguistics, 2017.
119. Wei F, Nair V, Menzies T. Why is differential evolution better than grid search for tuning defect predictors? arXiv, abs/1609.02613. 2016.
120. Solis FJ, Wets RJ-B. Minimization by random search techniques. *Math Oper Res* 1981;**6**:19–30.
121. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. In: *NIPS*, Lake Tahoe, Nevada, United States: NIPS, 2012.
122. Weber L, Minervini P, Münchmeyer J, *et al.* Nlprolog: reasoning with weak unification for question answering in natural language. In: *ACL (1)*, pp. 6151–61. Florence, Italy: Association for Computational Linguistics, 2019.
123. Minervini P, Costabello L, Muñoz E, *et al.* Regularizing knowledge graph embeddings via equivalence and inversion axioms. In: *ECML/PKDD (1)*. Lecture Notes in Computer Science, Vol. 10534. Springer, 2017, 668–83.
124. Muñoz E, Minervini P, Nickles M. Embedding cardinality constraints in neural link predictors. In: *SAC*, pp. 2243–50. Limassol, Cyprus: ACM, 2019.