BMC Medical Informatics and Decision Making

RESEARCH ARTICLE

Open Access

Investigating ADR mechanisms with Explainable AI: a feasibility study with knowledge graph mining

Emmanuel Bresso[1,2], Pierre Monnin[1,3], Cédric Bousquet[4,5], François-Elie Calvier[4], Ndeye-Coumba Ndiaye[6], Nadine Petitpain[7], Malika Smaïl-Tabbone[1] and Adrien Coulet[1,8,9*]

Abstract

Background: Adverse drug reactions (ADRs) are statistically characterized within randomized clinical trials and postmarketing pharmacovigilance, but their molecular mechanism remains unknown in most cases. This is true even for hepatic or skin toxicities, which are classically monitored during drug design. Aside from clinical trials, many elements of knowledge about drug ingredients are available in open-access knowledge graphs, such as their properties, interactions, or involvements in pathways. In addition, drug classifications that label drugs as either causative or not for several ADRs, have been established.

Methods: We propose in this paper to mine knowledge graphs for identifying biomolecular features that may enable automatically reproducing expert classifications that distinguish drugs causative or not for a given type of ADR. In an Explainable AI perspective, we explore simple classification techniques such as Decision Trees and Classification Rules because they provide human-readable models, which explain the classification itself, but may also provide elements of explanation for molecular mechanisms behind ADRs. In summary, (1) we mine a knowledge graph for features; (2) we train classifiers at distinguishing, on the basis of extracted features, drugs associated or not with two commonly monitored ADRs: drug-induced liver injuries (DILI) and severe cutaneous adverse reactions (SCAR); (3) we isolate features that are both efficient in reproducing expert classifications and interpretable by experts (i.e., Gene Ontology terms, drug targets, or pathway names); and (4) we manually evaluate in a mini-study how they may be explanatory.

Results: Extracted features reproduce with a good fidelity classifications of drugs causative or not for DILI and SCAR (Accuracy = *0.74* and *0.81*, respectively). Experts fully agreed that *73*% and *38*% of the most discriminative features are possibly explanatory for DILI and SCAR, respectively; and partially agreed (2/3) for *90*% and *77*% of them.

Conclusion: Knowledge graphs provide sufficiently diverse features to enable simple and explainable models to distinguish between drugs that are causative or not for ADRs. In addition to explaining classifications, most discriminative features appear to be good candidates for investigating ADR mechanisms further.

Keywords: Adverse drug reaction, Molecular mechanism, Mechanism of action, Explanation, Data mining, Machine learning, Knowledge graph, Explainable AI

Background

Molecular mechanisms behind harmful or beneficial effects of drugs are largely unknown. For instance, the molecular mechanism of highly prescribed drugs acetaminophen, lithium, and metformin is not completely

*Correspondence: adrien.coulet@inria.fr
[8] Inria Paris, Paris, France
Full list of author information is available at the end of the article
Emmanuel Bresso and Pierre Monnin have equal contributions

understood. Indeed, drug development process relies mainly on randomized clinical trials and postmarketing pharmacovigilance that evaluate drug efficacy and safety, independently from any mechanistic investigation [1]. However, understanding a drug's mechanism is fruitful: it can guide drug development, improve drug safety, and enable precision medicine, through better dosing or combination of drugs [2]. Aside from this partial ignorance, many elements of knowledge about drug ingredients are available in open-access knowledge graphs, such as their chemical and physical properties, their interactions with biomolecules such as their targets, or their involvements in biological pathways or molecular functions [3]. Knowledge graphs can be broadly defined as graphs of data with the intent to compose knowledge. Here, we consider knowledge graphs represented using Semantic Web technologies, including RDF (Resource Description Framework) and URI (Uniform Resource Identifier) [4, 5]. In such knowledge graphs, nodes represent *entities*, also named *individuals*, of a domain (e.g., acetaminophen), *classes* of individuals (e.g., analgesics), or *literals* (e.g., strings, dates, numbers). Literals are purposely discarded in this study. Nodes are connected by directed edges that are labeled with *predicates* (e.g., `transportedBy`).

We propose in this article to leverage elements of knowledge about drugs that lie in biomedical knowledge graphs to investigate ADR molecular mechanisms. To this aim, we experiment with knowledge graphs as an input to machine learning approaches (i.e., methods of Artificial Intelligence, commonly denoted AI) that are natively explainable. Indeed, Explainable AI usually refers to research on methods that provide explanatory elements to results (i.e., a classification) of sub-symbolic approaches (e.g., ensembles or Deep Neural Networks) [6]. In a broadly manner, we consider symbolic approaches that provide models that are interpretable by humans, and investigate if features of these models may be explanatory for biomolecular processes involved in ADRs. We particularly consider a knowledge graph named PGxLOD, which encompasses and connects drug, pathway, and biomolecule data [7]; and two particular types of ADRs: drug-induced liver injuries (DILI) and severe cutaneous adverse reactions (SCAR). We choose these types of ADRs first because hepatic or skin toxicities are commonly monitored during drug development for their importance in pharmacovigilance [8]. Indeed, drugs cause frequently hepatic and skin events, and those are severe enough to potentially lead to drug withdrawal in Phase IV. Second, it exists good quality *expert classifications* that label sets of drugs as either causative or not for these types of ADRs [9, 10]. First, our work identifies biomolecular features from our knowledge graph that enable an automatic reproduction of expert classifications. In particular, we mine the graph for neighbors of drugs, paths and path patterns (i.e., paths composed of general classes) rooted by drugs and passing by at least one entity of the following types: pathway, gene/protein, Gene Ontology (GO) term or MeSH term. Second, we isolate both predictive and interpretative features hypothesizing that, in addition to be explanatory for the classification, those may also be explanatory for ADR mechanisms. To this second aim, we consider simple, but explanatory classification techniques, i.e., Decision Tree and propositional rule learner over extracted features, because they provide human-readable models in the form of rules. Finally, we ask three human experts if they consider isolated features as possibly explanatory for ADRs.

A first family of related works can be described as *explanatory*, where known Drug-ADR associations, such as those found in SIDER, are used to highlight molecular mechanisms that may be impacted in ADRs. A second family of works is *predictive*, where data about molecular mechanisms (e.g., GO molecular processes or KEGG pathways) are associated with drugs and used as features to predict ADRs [11]. Boland et al. [12] survey existing works for both predicting ADRs and elucidating their mechanisms; they interestingly list data and knowledge resources that may support these efforts.

In the explanatory family, Lee et al. [13] associate SIDER side effects and GO biological processes through drugs, by the combination of a Drug-Side effect and a Drug-Biological process networks. Highlighted relationships are obtained using statistical approaches (i.e., enrichment and *t*-score) and evaluated in regard to co-occurrences in PubMed abstracts. Wallach et al. link drugs to proteins through molecular docking, then to pathways through the KEGG database, and use logistic regression and feature selection approaches to select pathways most probably impacted in side effects [14]. Bresso et al. [15] group frequently associated drug reactions and propose elements of explanations of their grouping using Inductive Logic Programming. Also, Chen et al. [16] propose a computational algorithm to infer Protein-ADR relationships from a network of protein-protein interactions, ADR-ADR similarities and known protein-ADR relations.

In the predictive family, Bean et al. [17] build a network of drugs, targets, indications and ADRs to select features that are good predictors for ADRs in a logistic regression setting. PhLeGrA is an analytic method implementing Hidden Conditional Random Fields to allow the calculation of the probability of drug reactions given a input drug and a knowledge graph of drugs, proteins, pathways and phenotypes [18]. Similarly, Muñoz et al. [19] propose a specific way to extract features from knowledge graphs

for ADR prediction. They show that several multi-label learning models perform well for this task. Our work is similar to some extent, however it uses simpler but explanatory classifiers, and goes a step further by identifying features, subsequently proposed as explanatory elements for ADRs. Indeed, we hypothesize that within the large set of considered features, those that are both highly predictive and associated with a good level of interpretability may suggest to experts plausible elements of explanation. In Dalleau et al. [20], knowledge graph mining serves a completion perspective and aims at inferring links between drugs and genes. All PhLeGrA, Muñoz et al., Dalleau et al., and the present work illustrate the interest of aggregating several LOD (linked open data) sets for knowledge discovery and data mining tasks, as discussed by Ristoski and Paulheim [21].

Shi and Weninger [22] use a similar approach to ours, but from a fact checking perspective. Indeed, for each relation type p, a set $D^k_{(o_u,o_v)}$ of discriminative paths is learned. This set contains anchored predicate paths $o_u \xrightarrow{r_1} \xrightarrow{r_2} \dots \xrightarrow{r_k} o_v$ of length k that describe a statement $o_u \xrightarrow{p} o_v$, where o_u and o_v are respectively the ontology classes associated with nodes u and v. To check whether a triple $s \xrightarrow{p} t$ is true, they use the learned set of discriminative paths for the relation $D^k_{(o_s,o_t)}$. In such sets, only paths with the most discriminative power are kept. Similarly to our approach, they use ontology class generalization but apply it only to start and end nodes s and t of the fact to be checked. This differs from our approach as we apply generalization on each intermediate node (see Methods Section for details). Additionally, their path modelling allows reverse traversal, i.e., $\xrightarrow{p^{-1}}$, and constrains both source and target nodes, while we only constrain source nodes. Previous works also use knowledge graph mining to provide explanations. Those include Explain-a-LOD that enriches statistical data sets with features from DBpedia. It uses correlation between attributes and rule learning to provide hypothesis explaining statistics [23]. Explain-a-LOD relies on FeGeLOD to extract features from the DBPedia knowledge graph [24]. In particular, FeGeLOD extracts two types of features similar to ours: paths of size 1 ($\xrightarrow{r} e$) starting at the entities of interest; paths of size 1 ($\xrightarrow{r} t$) where the original entity (e) is replaced with ontology classes (t) it instantiates. In the same vein, KGPTree [25] extracts paths of the form `root` → `predicate` → `entity` → `predicate` ...→ `entity`, while allowing generalizations on both predicates and entities, whereas we offer generalization on entities only. However, they only allow a generalization to a unique and broad type denoted with a wild card (*), while we allow a granular generalization following ontology class hierarchies. FeGeLOD and KGPTree extract only paths and path patterns, whereas one may want to extract other common structures such as subtrees. Mustard Python library offers such functionalities applying Graph Kernels on RDF graphs, plus additional facilities such as detecting hubs or low frequency patterns [26, 27]. However, it does not allow generalization operations.

The contribution of our work is twofold: first, we show that knowledge graphs provide sufficiently diverse features to enable simple and explainable models to distinguish between drugs that are causative or not, for two types of ADRs commonly monitored; second, we manually evaluate in a mini-study that in this setting, predictive features constitute good candidates for investigating ADR mechanisms further. The following sections present materials and methods, obtained results and a discussion about our methodological choices and results.

Materials and methods

Data sources

PGxLOD

PGxLOD is a linked open data (LOD) knowledge graph built for pharmacogenomics (PGx) and encoded in RDF [7]. It aggregates data mainly about drugs, genetic factors, phenotypes and their interactions from six data sources: PharmGKB, ClinVar, DrugBank, SIDER, DisGeNET and CTD; but also includes references to Gene Atlas, UniProt, GOA and KEGG. In particular, it includes pharmacogenomic relationships, i.e., n-ary relations that represent how a genomic factor may impact a drug response phenotype. These relations are compiled from PharmGKB and the literature. We used PGxLOD version 4 that encompasses 88,132,097 triples. Table 1 presents its main statistics. PGxLOD is available at https://pgxlod.loria.fr.

In our study, PGxLOD enables to associate phenotypic and molecular features to drugs, by the exploration of their neighborhood in the graph.

Table 1 Types and numbers of entities available in the PGxLOD knowledge graph

Concept	Number of instances
`Drug`	63,485
`GeneticFactor`	494,982
`Phenotype`	65,133
`PharmacogenomicRelationship`	50,435
from PharmGKB	3650
from the literature	36,535

Pharmacogenomic relationships of PGxLOD are of two provenances: the PharmGKB expert database and the literature

Table 2 Classes and size of the original DILIRank expert classification

Class	# drugs
Most DILI concern	192
Ambiguous DILI concern	254
Less DILI concern	278
No DILI concern	312
Total	**1036**

Classes group drugs causative or not for drug-induced liver injury (DILI) on the basis of FDA-approved drug labels and a semi-automatic method

Table 3 Classes and size of the original RegiSCAR drug notoriety list

Class	# drugs
Very probable (3)	18
Probable (2)	19
Possible (1)	94
Unlikely (0)	697
Very unlikely (− 1)	46
Total	**874**

Classes group drugs causative or not for severe cutaneous adverse reactions (SCAR)

Drug expert classifications and their preprocessing

We experiment with two expert classifications of sets of drugs labeled as either causative or not for ADRs. The first concerns drugs causative for drug induced liver injury (DILI), and the second drugs causative for severe cutaneous adverse reactions (SCAR).

DILI classification We built our DILI classification from DILIRank, a list of 1036 FDA-approved drugs classified by their risk of causing DILI [9]. DILIRank distinguishes between four classes listed in Table 2. This classification was obtained by (1) the curation of information gathered from FDA-approved drug labels, setting an initial list of 287 drugs; (2) a semi-automatic approach that completes the list up to 1036 drugs, by combining information from hepatotoxicity studies and the literature. We sub-sampled from DILIRank *370 drugs (146 DILI*⊕, *a*nd 224 DILI⊖) that fulfill criteria required for our subsequent analysis: being either in the Most—or No—DILI concern classes, being associated with a SMILES (simplified molecular-input line-entry system) description, and being mapped to PGxLOD. The latter mapping was obtained with PubChemIDs, which are available both in DILIRank and PGxLOD (coming from PharmGKB, DrugBank, and/or KEGG). Drugs satisfying these criteria and classified as Most-DILI concern constitute the DILI⊕ subset, and those classified as No-DILI concern constitute the DILI⊖.

SCAR classification Our SCAR classification relies on a manually built classification called "Drug notoriety list", shared by members of the RegiSCAR project. This list was originally assembled for the evaluation of the ALDEN algorithm, which assesses the chance for a drug to cause Stevens-Johnson Syndrome and Toxic Epidermal Necrolysis, a specific type of SCAR [10, 28]. This classification lists 874 drugs and assigns them to five classes representing various levels of association with SCAR. These classes are listed in Table 3. We sub-sampled from the RegiSCAR drug notoriety list *392 drugs (102 SCAR* ⊕, *and 290 SCAR*⊖) fulfilling two criteria: being mapped to PGxLOD and having a SMILES description. The mapping starts with 874 drug names, which is the only description available in the RegiSCAR list. Drug names are matched with lists of synonyms associated with drugs in PharmGKB and DrugBank. Drugs satisfying these criteria and classified as Very probable, Probable or Possible constitute the SCAR⊕ subset, and those classified as Unlikely or Very unlikely, the SCAR⊖.

Methods

An overview of the steps of the proposed method is provided in Fig. 1.

Graph-based feature construction

Graph canonicalization In knowledge graphs, several nodes may co-exist, while representing the same entity. For example, a unique drug in PGxLOD can be represented by two nodes: one from PharmGKB and another from DrugBank. Each of these two nodes have their own connections to other nodes in the knowledge graph. Therefore, the union of their edges constitutes all the available knowledge about the drug they both represent. As they represent the same drug, they are connected through an `owl:sameAs` edge. In order to avoid traversing such edges, we *canonicalize* the knowledge graph, that is to say, nodes linked by `owl:sameAs` edges are grouped into a unique node as a pre-processing step, before building graph-based features. Such a procedure corresponds, in graph theory, to the contraction of `owl:sameAs` edges.

Paths, path patterns, and neighbors as drug features

From an arbitrary set of drugs D (such as DILI⊕ ∪ DILI⊖) and a canonicalized knowledge graph (such as PGxLOD), a set of graph-based features is built on D. We distinguish three kinds of features: paths, path patterns and neighbor nodes. First, we build *paths* rooted by drugs from D. The neighborhood of each drug $d \in D$ is explored in the graph with a max distance of k, generating sequences of predicates and nodes of max length k. Accordingly, if a path $d \xrightarrow{p_1} n_1 \xrightarrow{p_2} n_2$ is found in the knowledge graph (with $k = 2$), the drug d is associated with the feature $\xrightarrow{p_1} n_1 \xrightarrow{p_2} n_2$ in the output matrix.

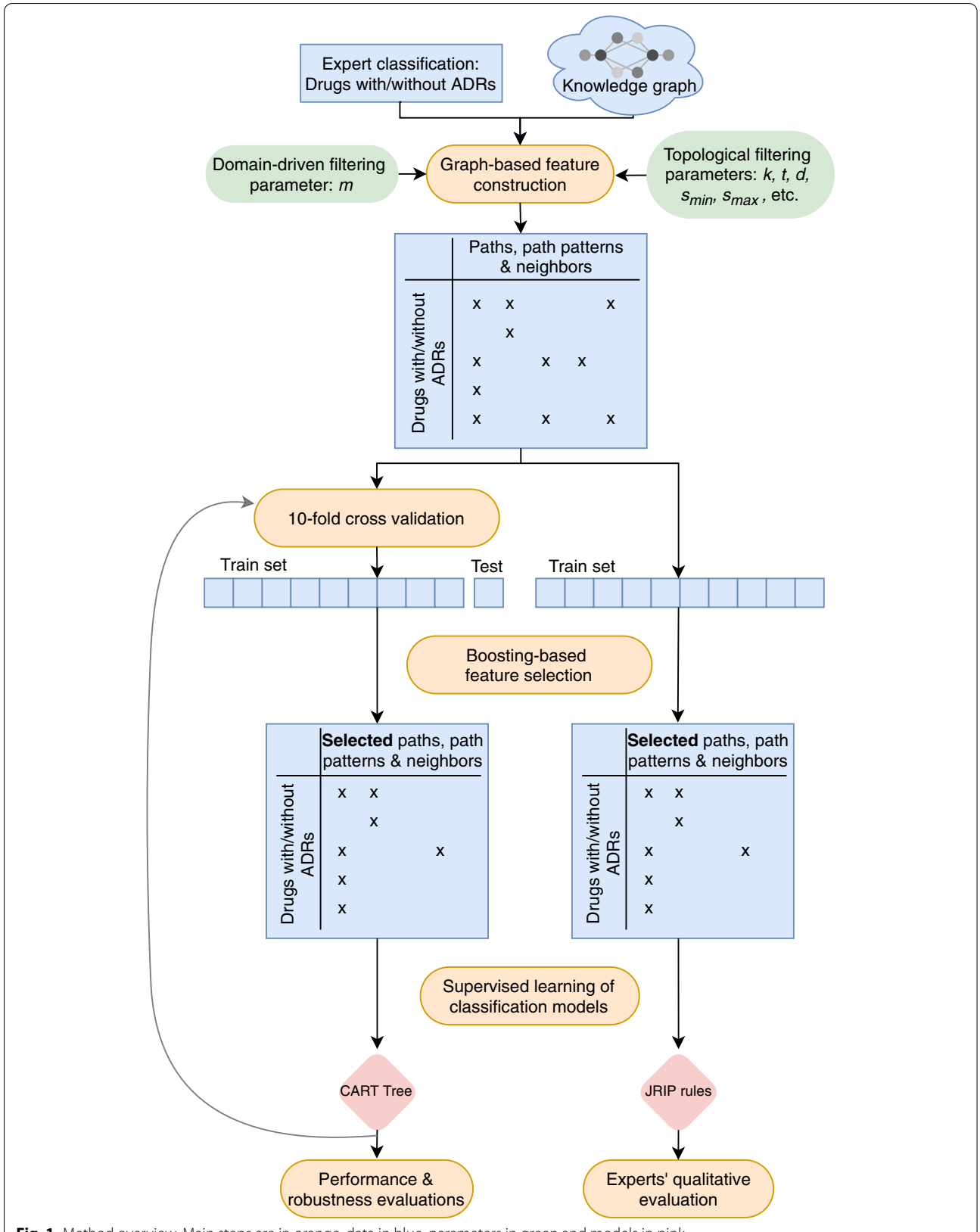

Fig. 1 Method overview. Main steps are in orange, data in blue, parameters in green and models in pink

Second, we build *path patterns* that generalize paths by considering ontology classes instantiated by nodes. The aim of path patterns is to offer more general descriptions, which have more chances to be shared by several entities. For example, if n_1 instantiates C_1 and n_2 instantiates C_2, we add the following path patterns: $\xrightarrow{p_1} C_1 \xrightarrow{p_2} n_2$, $\xrightarrow{p_1} n_1 \xrightarrow{p_2} C_2$, $\xrightarrow{p_1} C_1 \xrightarrow{p_2} C_2$, $\xrightarrow{p_1} \top \xrightarrow{p_2} n_2$, $\xrightarrow{p_1} n_1 \xrightarrow{p_2} \top$, and $\xrightarrow{p_1} C_1 \xrightarrow{p_2} \top$, $\xrightarrow{p_1} \top \xrightarrow{p_2} C_2$, and $\xrightarrow{p_1} \top \xrightarrow{p_2} \top$. It is noteworthy that, to allow a high level of generalization, we always consider the top-level class $\top$ in the generalization procedure. To leverage hierarchies organizing ontology classes, a node n is only generalized by ontology classes at a distance of at most t from itself, following instantiation and subsumption edges.

Third, we list *neighboring nodes*, i.e., any node that can be reached from $d \in D$ within a distance of max k. We do not keep track of the distance between d and its neighbors: if a node n can be reached from a drug d_1 after 2 hops and from a drug d_2 after 3 hops, these two drugs will be associated with the neighbor n in the output matrix. Interestingly, a neighbor n can be represented by the general path pattern $(\xrightarrow{*} *)^{\{0,k-1\}} \xrightarrow{*} n$.

Features can potentially be noisy and numerous, leading to a combinatorial explosion of their number. That is why, we add several constraints to only keep the most interesting features. First, we only keep the most specific paths and path patterns, considering that a node is more specific than the classes it instantiates and a class is more specific than its superclasses. A path or path pattern P_1 is more specific than another path or path pattern P_2 if each node/class in P_1 is more specific than the node/class at the same position in P_2. Thus, if $\xrightarrow{p_1} n_1 \xrightarrow{p_2} n_2$ is associated with the exact same drugs as $\xrightarrow{p_1} C_1 \xrightarrow{p_2} C_2$, then the path pattern is removed and only the path constitutes a feature. Similarly, if $\xrightarrow{p_1} C_1 \xrightarrow{p_2} C_2$ is associated with the exact same drugs as $\xrightarrow{p_1} \top \xrightarrow{p_2} \top$, then we only keep the first one. Second, the exploration is stopped at nodes whose degree is greater than a parameter *deg*. Such nodes are usually called *hubs* [27], and expanding a path ending at a hub may generate a large number of new paths associated with the same drugs. Third, we consider minimal and maximal *supports* of features (denoted $s_{\min}$ and $s_{\max}$) as additional filters. The support of a feature consists in the number of drugs associated with a feature, i.e., neighbors of a same entity or rooting a path or a path pattern. Only features associated with more than $s_{\min}$ drugs and less than $s_{\max}$ drugs are output in the final matrix. Fourth, two blacklists avoid traversing noisy or unwanted edges. Edges are not traversed if their predicate is blacklisted (in $b_{\text{predicates}}$) or if the adjacent node instantiates (directly or indirectly) a blacklisted class (in $b_{\text{exp-types}}$). For example, we blacklist predicates of PROV-O that are used to describe provenance metadata. By blacklisting `rdf:type` in $b_{\text{predicates}}$, we make sure the exploration is performed through entities and not classes of PGxLOD. Relations between classes are only considered when generalizing paths. Aside from noisy features and combinatorial explosion, we use these blacklists to prevent considering features that "obviously" carry the signal we are trying to predict. For example, we blacklist all "side-effects" links from SIDER, which may directly link drugs in D with the side effect we aim at predicting. We also blacklist all classes from MeSH related to SCAR or DILI to avoid taking into account nodes instantiating them. A third blacklist ($b_{\text{gen-types}}$) avoids generalizing nodes in paths by blacklisted classes. This is particularly useful to withdraw general classes (e.g., `Drug`) that increase the number of generated path patterns while not adding useful information.

Besides previous topological constraints, we also perform a domain-driven filtering configured by parameter m. Indeed, in our objective of explaining ADRs, experts highlighted that features that may be explanatory to them mention pathways, genes, GO, and MeSH terms. For this reason, we propose three post-processing atomic filters, only keeping neighbors, paths or path patterns containing at least a pathway ($m = \texttt{p}$), a gene or a GO class ($m = \texttt{g}$), or a MeSH class ($m = \texttt{m}$). Such atomic filters can be combined to form disjunctive filters. For example, the $m = \texttt{pg}$ filter keeps neighbors, paths or path patterns containing at least a pathway or a gene or a GO class.

Table 4 summarizes the parameters that limit the number of features in the output matrix. Additional details about our method of feature construction are provided in [29].

Cross-validation strategy

Once drug features are extracted from the knowledge graph, they are given to a machine learning algorithm that learns a model, which mimics the expert classification. To quantitatively evaluate our approach, we adopt a 10-fold cross-validation strategy, meaning that we repeated the following steps of our learning pipeline (i.e., feature selection, and training) 10 times, holding out each time one tenth of our labeled data for testing. The split in 10 folds is performed once, randomly.

Feature selection with boosting

The concept of boosting relies on the sequential learning of classifiers that successively focus on getting correct the examples that were wrongly classified by previous classifiers, with a weight system: correctly classified examples loose weight, whereas falsely classified ones gain weight [30]. The final classification is built by a weighting of the results obtained from the various classifiers. Ensembles of decision trees can be used in a boosting strategy for

Table 4 Parameters used to limit the number of features

Parameter	Domain	Description
k	$\mathbb{N}^+$	Maximum length of paths and path patterns
t	$\mathbb{N} \cup \{-1\}$	Maximum depth of generalization
deg	$\mathbb{N} \cup \{-1\}$	Maximum node degree to allow expansion
s_{min}	$\mathbb{N}$	Minimum support for features
s_{max}	$\mathbb{N}$	Maximum support for features
Undirected	$\mathbb{B}$	Consider the graph undirected or directed
$b_{\text{predicates}}$	List of URIs	Blacklist of predicates not to traverse
$b_{\text{exp-types}}$	List of URIs	Blacklist of types of entities not to traverse
$b_{\text{gen-types}}$	List of URIs	Blacklist of types not to traverse in generalization
m	{`no- check`, `p`, `g`, `m`, `pg`, `pgm`}	Domain-dependent filter for features with at least a pathway (`p`), a gene (`g`) or a MeSH class (`m`)

Each parameter is associated with a domain for its value

the estimation of feature importance in order to select the most important features for a subsequent learning model [31]. This approach is frequently named "wrapper-based feature selection". We adopt this approach, using the AdaBoost algorithm, with up to 10 decisions trees learned from the whole train set [32]. In case of perfect fit, the learning procedure is stopped early. For the Decision Tree algorithm we used CART from scikit-learn, Gini impurity, all features considered at each split and a *minimal number of examples per leaf* $= 5$. Every train example is used, but classes are artificially balanced: examples are associated with a different weight depending on whether they belong to the $\oplus$ or $\ominus$ class. Features that appear in at least one of the decision trees are selected for the subsequent step of our learning pipeline.

Performance and robustness evaluation

For evaluating our capacity of distinguishing between drugs associated or not with an ADR, we train a last decision tree, using selected features only. Note that because the selection step is repeated at each iteration of the cross validation strategy, selected features may vary from one iteration to another. Algorithm and parameters are the same as those of the selection step, i.e., CART from scikit-learn with Gini, all features considered at each split, *minimal number of examples per leaf* $= 5$, and weighted instances for class balancing.

Performance results are reported in term of Precision, Recall, F1-score (reference class $\oplus$), accuracy and AUC-ROC. Metrics are averaged over the 10 iterations of the cross validation.

A robustness evaluation is also performed to assess the impact of the train set (i.e., expert classifications) on the final classifier. To this aim, first, we reproduced the experiment, but with a shuffled class assignment in the train set. This leads to a train, denoted by $\cdot^{shuffled}$ (where $\cdot$ is either *DILI* or *SCAR*) where drugs are associated randomly to either the class $\oplus$ or $\ominus$. This first sanity evaluation mainly checks the presence of a nonrandom signal in the train. Second, we replace the set of drugs from $\ominus$ by a set of drugs randomly picked out of our knowledge graph. We repeat this random draw five times for each expert classification resulting in 10 train sets denoted $\cdot^{random\ominus_i}$ (where $\cdot$ is either *DILI* or *SCAR* and i is an index taking values from 1 to 5). In each case the draw is made from nodes of the knowledge graph that instantiate (directly or indirectly) `pgxo:Drug`; are identified with a URI from PharmGKB or DrugBank namespaces; are linked by a `x-pubchem` predicate to a PubChem URI (in order to have a SMILES associated with the drug); and are not drugs of the original DILI or SCAR expert classifications. In the case of DILI, this is a draw of 224 nodes out of 5893 in the canonical graph. In the case of SCAR, this is a draw of 290 nodes out of 5921. This second evaluation checks the impact of the selection of negative examples on classifier performances. In addition, we count the number of features that are present in 5, 4, 3 or 2 of the five $\cdot^{random\ominus_i}$ experiments.

Qualitative evaluation by human experts

To go beyond performance evaluation, we produce a classification model made of rules using the RIPPER algorithm, and its Java implementation named JRip [33]. JRip has the advantage over decision trees to provide classification rules that are more concise and by consequence easier to interpret for humans. JRip actually implements a propositional rule learner, produces relatively non redundant rules in comparison to rules that could be learned following branches of a decision tree. However, JRip and CART decision tree usually perform very similarly, since they implement similar pruning strategies and stopping criteria. To evaluate this assessment, we performed a 10-fold cross validation of the JRip approach

and compared performances with CART. JRip generates rules of the form:

$$\left(\bigwedge_{\forall a \in A} a\right) \wedge \left(\bigwedge_{\forall b \in B} \neg b\right) \Rightarrow c$$

where A is a set of attribute-value pairs affirmed, B is a set of attribute-value pairs negated, and c is the minority class of the classification problem. Accordingly, $c = \oplus$ in our study. Note that A or B can be an empty set, but not both at the same time. In other words, JRip rules consist in the conjunction of affirmed and negated features.

As for CART Trees, we start with an initial step of feature selection with AdaBoost, with the same parameters but this time considering all examples of the train set. Following, JRip rules are built considering also all examples of the train set, and features that appear in at least one of the trees built by AdaBoost. To be consistent, with the CART Tree setting, we set to 5 the *minimal number of instances per rule*, which can be compared to the *minimal number of examples per leaf*.

JRip rules are post-processed to facilitate their interpretation by our experts. First, among features that are path patterns, we discard those only involving generic classes such as `Resource` or `Drug`. Indeed, such path patterns turned out to be impossible to interpret. Second, features are translated into a readable format, by resolving ids with associated labels (e.g., `drugbank:BE0003543` is resolved as "Cytochrome P450 1A1") and by interpreting and rewriting paths and path patterns in an understandable form (e.g., $\xrightarrow{\texttt{drugbank\_vocabulary:enzyme}}$ `geneatlas_vocabulary:Resource` is turned in $\xrightarrow{\text{interactsWith}}$ Enzyme). Because of the limited number of features in rules, this translation is made manually, on the basis of descriptions of predicates, classes, and entities found in their original knowledge graph or database.

We asked three experts in pharmacy and pharmacology to review independently each attribute (i.e., feature) of the rules, to evaluate if they may be explanatory for ADRs. Each expert has to answer a voluntarily simple three-way question: "according to your own knowledge or the state of the art, do you think that the feature is explanatory for DILI?" (SCAR, respectively). Possible answers are "yes", "maybe (possible, but not obvious)" and "no (probably not explanatory)". We allow expert to check the literature or any state-of-the-art resource, but up to 15 min, since we consider that more time causes to fall in the "no (probably not explanatory)" option. To guide their decision on each feature, experts are provided with two Web links: one pointing to the list of drugs from the $\oplus$ train set that supports the feature; one to the page of the main entity mentioned in the feature (i.e., the neighbor node, or the final node of the path or path pattern) in an expert database: DrugBank, ChEBI, KEGG, QuickGO, or MeSH browser, depending on the namespace of the node. After expert reviews, answers were normalized, under their supervision, to guarantee all experts interpret the negation of features the same way. For each feature we check if at least one, two or three of the experts think it is or may be explanatory, if the three think it is explanatory and if the three think it is not. In addition, we compute Cohen's kappa coefficient to evaluate the average agreement between experts with two different settings: considering the three different answers as distinct, or considering answers "yes" and "maybe" as a unique positive answer.

Results

Graph-based feature construction

We experimented graph exploration with combinations of the following parameters values $k \in \{1, 2, 3, 4\}$, $t \in \{1, 2, 3\}$, $deg = 500$, $undirected = \texttt{false}$, $s_{\min} = 5$, $s_{\max} = +\infty$ and $m \in \{\texttt{p}, \texttt{g}, \texttt{m}, \texttt{pg}, \texttt{pgm}\}$. $k = 4$ was only tested with $t = 1$ because of memory issues caused by the high number of generated features with greater values of t. However, we report only the best results, which were obtained with $k = 3, t = 3, m = \texttt{pgm}$ for DILI and $m = \texttt{pg}$ for SCAR. These values of m enable to conserve only features that includes an entity that is either a pathway, a gene, or a GO term (for `pg`); or a pathway, a gene, a GO, or MeSH term (for `pgm`).

The construction of graph-based features is obviously limited by the amount of available memory. We enforce $s_{\min}$ to be set to allow the construction of paths and path patterns, in order to avoid combinatorial explosion. Accordingly their number is reported only once we reduced the number of all possible combinations, which we were not able to compute. We used a server with a Xeon E5-2680 v4@2.40GHz CPU, 28 cores/56 threads and 768GB of memory. As an illustration, we obtained the features associated with the DILI expert classification under $k = 3, t = 3$ in approximately 1 h using 62 GB of RAM, and under $k = 4, t = 1$ in 4 days using 380 GB of RAM.

To provide with an idea of the size of the considered neighborhood with regards to all reachable nodes, Table 5 reports statistics about numbers of neighbors, paths and path patterns reachable with different level of filtering. In particular, we report sizes of the full neighborhood of drugs, and of 3 levels of filtering. The first level of filtering consists in prohibiting the expansion of the neighborhood through nodes with a degree higher than 500 ($deg = 500$). In the full neighborhood and first level of filtering, k and t are not constrained since no path or path pattern is computed, but we report max k and t reached

Table 5 Numbers of drug features extractable from the knowledge graph, with different levels of filtering

		DILI	*SCAR*
No filtering ($deg = -1$)	Neighbors	5,488,531	5,488,510
	k	19	20
	t	21	21
Filtering level 1 ($deg = 500$)	Neighbors	2,419,957	2,419,920
	k	23	23
	t	21	21
Filtering level 2 ($deg = 500, s_{\min} = 5, k = 3, t = 3$)	Neighbors	175,652	179,694
	Paths & path patterns	20,145,635	29,011,996
Filtering level 3 ($deg = 500, s_{\min} = 5, k = 3, t\ 3, m_{\text{DILI}} =$ pgm and $m_{\text{SCAR}} =$ pg)			
	Neighbors	4069	1594
	Paths & path patterns	102,674	86,753

The first line corresponds to the full neighborhood of drugs from DILI and SCAR expert classifications. $deg = -1$means that all nodes are considered, regardless of their degree, whereas $deg = 500$ in Filtering level 1 means that nodes with a degree $>$ deg are filtered out. In the two first lines (No filtering and Filtering level 1), k and t are unconstrained, so reported values are maximum k and t observed in the graph. Paths and paths pattern are computed only when deg and $s_{\min}$ (minimum support) are set, to avoid combinatorial explosion. Filtering level 2 and 3 share the following additional parameters: $undirected = \texttt{false}$, $s_{\max} = +\infty$. In Filtering level 3, m is set for additional filtering. Distinct values for m chosen respectively for DILI and SCAR are those associated with the best performances, e.g., $m_{DILI} = \texttt{pgm}$ and $m_{SCAR} = \texttt{pg}$

Table 6 Quantitative evaluation of our classifiers of drugs associated with ADRs or not (DILI or SCAR)

Algorithm	Data set	Precision	Recall	Accuracy	AUC	F1-score
CART	*DILI*	0.68	0.67	0.74	0.73	0.67
	SCAR	0.64	0.68	0.81	0.77	0.65
JRip	*DILI*	0.82	0.71	0.72	0.74	0.75
	SCAR	0.88	0.70	0.71	0.74	0.77

in the neighborhood. We note that k is surprisingly lower in the larger neighborhood, i.e., 19 and 20 versus 23 and 23 with the first level of filtering. This can be explained by the fact that with "hubs" (*nodes with deg* > 500), the full neighborhood can be reached through smaller paths. We also observe that because of this first filtering, certain nodes are not accessible anymore (when every possible path to them pass through a hub), which results in a smaller number of neighbors. The second level of filtering comes on top of the first, and constrains neighbors, paths, and path patterns to have a minimal support set of 5 ($s_{\min} = 5$), a max length of 3 ($k = 3$), and a max depth 3 for generalization of paths into patterns ($t = 3$). The filtering level 3 comes on top of the second, and constrains neighbors, paths, and path patterns to contain a pathway, a gene, a GO term, or a MeSH term for DILI ($m = \texttt{pgm}$), or to contain a pathway, a gene, or a GO term for SCAR ($m = \texttt{pg}$). The filtering level 3 is the one used in the following experiments, because it is computable in our setting, while providing the best performances in our set of experiments.

Table 7 Robustness evaluation of our classifiers

Data set	Accuracy	AUC	F1-score ⊕
$DILI^{shuffled}$	0.52	0.49	0.36
$DILI^{random\ominus_1}$	0.92	0.91	0.89
$DILI^{random\ominus_2}$	0.92	0.91	0.89
$DILI^{random\ominus_3}$	0.93	0.92	0.91
$DILI^{random\ominus_4}$	0.93	0.92	0.90
$DILI^{random\ominus_5}$	0.92	0.91	0.90
$SCAR^{shuffled}$	0.63	0.51	0.26
$SCAR^{random\ominus_1}$	0.93	0.89	0.86
$SCAR^{random\ominus_2}$	0.94	0.90	0.88
$SCAR^{random\ominus_3}$	0.93	0.90	0.86
$SCAR^{random\ominus_4}$	0.92	0.89	0.85
$SCAR^{random\ominus_5}$	0.93	0.89	0.86

$\cdot^{shuffled}$ corresponds to an experiment where class labels (i.e., $\oplus$ or $\ominus$) are randomly affected to drugs. $\cdot^{random\ominus_i}$ correspond to experiments where negative examples (i.e., $\ominus$) are replaced by drugs randomly picked in the knowledge graph. Indices i from 1 to 5 refer to 5 different draws

Quantitative and robustness evaluation

Performances of our CART decision trees to distinguish between drugs associated or not with ADRs are reported in Table 6. With both types of ADRs, we obtained accuracy and AUC higher than 0.70, illustrating the fact that learned classifiers reproduced a large part of expert classifications, on the basis of features from the knowledge graph.

Robustness of classifiers is illustrated by the results provided in Table 7. $DILI^{shuffled}$ and $SCAR^{shuffled}$ are associated with AUC of 0.49 and 0.51, respectively. This illustrates that a random assignment of class labels in the train set, leads to a classifier that randomly assign labels to test examples. This is expected, but illustrates that expert classifications encompass a signal that our classifiers learn and reproduce, to some extent.

Classifiers trained with a random pick of negative examples ($\cdot^{random\ominus_i}$) instead of negatives picked by experts are significantly better for the three performance metrics (t test, $t > 36, p < 1.7 \times 10^{-6}$). This reveals it is harder for our classifier to discriminate between positives and negatives of expert classifications, than it is between positives and randomly picked drugs. This lets us assume that negatives from expert classifications are somehow similar to positives (they may share some properties) and harder to distinguish for the classifier, even if not associated with the studied ADR. When comparing features used in the five random pick of $DILI^{random\ominus_i}$, we observed that, out of a mean of 122 features (sd=8), 6, 12, 27 and 99 were common to respectively 5, 4, 3 and 2 picks. With $SCAR^{random\ominus_i}$, out of 108 features (sd=8), 2, 6, 23 and 81 were common to respectively 5, 4, 3 and 2 picks.

Expert evaluation

JRip produced 6 and 5 rules for DILI and SCAR, respectively. The translation of these rules is available in Additional file 1: Tables S1 and S2. After removing uninformative features, we obtained 11 and 13 distinct features, respectively. Those are provided in Additional file 1: Tables S3 and S4. These features are those reviewed by our three experts in pharmacology (CB, CNC, and NP). Quantitative performances of the JRip algorithm are presented in Table 6 for comparison with CART. On both datasets (DILI and SCAR) differences in performances (Precision, Recall, Accuracy, F-measure) are not statistically significant (t test, $p < 0.05$).

The ratio (and number) of features for which experts reached an agreement, or for which at least one, two or three of the experts think they are or may be explanatory (answers "yes" or "maybe") are provided in Table 8. Detailed results of the manual evaluation are provided in Additional file 2. We observe that no feature generated by JRip is considered as unexplanatory by all three experts. In other words, every feature is thought as possibly explanatory by at least one expert. The ratio of features having a full agreement between experts on the possibility of being explanatory is reduced compared to those having a partial agreement but stays relatively high (0.73 features) for DILI and moderate (0.38 features) for SCAR. Full agreement for features being explanatory (all three experts answer "yes") remains minor, but exists. Kappa's Cohen agreement score is $\kappa_{n=3} = 0.26$ when considering answers "yes", "maybe", and "no" independently, but reaches $\kappa_{n=2} = 0.70$ when the problem is reduced to two classes by aggregating "yes" and "maybe" answers. Note that Table 8 reports in its fifth column the ratio of features that reach full agreement for our three experts when "yes" and "maybe" answers are aggregated. Additional file 2 contains results of the manual evaluation of the features.

Table 8 Ratio of features that either reach a full agreement for being unexplanatory, explanatory or are considered as possibly explanatory to various extents

Data set	With agreement on unexplanatory	Features possibly explanatory for			With agreement on explanatory
		⩾1	⩾2	⩾3 experts	
DILI	0	1 (11)	0.90 (10)	0.73 (8)	0.18 (2)
SCAR	0	1 (13)	0.77 (10)	0.38 (5)	0.08 (1)

Absolute numbers are reported in parentheses. Ratio of unexplanatory features are in the left column, whereas explanatory features are in the right columns. The three middle columns count numbers of features that are, or may be, explanatory according to at least one, two or three experts. Numbers of considered features are 11 and 13 for DILI and SCAR respectively

Examples of features and elements of interpretation

Three features reach an agreement for being explanatory (i.e., three answers "yes" per feature). Those can be interpreted as elements that are well known for being explanatory, or at least associated, with DILI or SCAR mechanisms. As a first illustration, $\xrightarrow{\text{interactsWith}}$ Enzyme $\xrightarrow{\text{cellularComponent}}$ Endoplasmic reticulum reached an agreement for DILI. This is explained by the fact that endoplasmic reticulum is known, in particular in liver tissues, to host primarily cytochrome P450 enzymes, well known for being involved in drug metabolism [34]. As a second illustration, *Cytochrome P450 2B6* reached an agreement for SCAR, whereas genomic variations in the gene coding for this enzyme are known for being associated with SCAR [35]. One might consider fairly that this feature does not bring new explanatory elements, although it can be considered a minimum

that our method highlights well established explanatory elements.

All other features did not reach an agreement, or reach one for "maybe". Each of those is interesting to explore for further interpretation, but for the sack of briefness, we will only discuss two of them here. $\text{First}_{\neg(\xrightarrow{\text{involvedIn}}\text{Pathway}\xrightarrow{\text{associatedWith}}\text{Disease}\xrightarrow{\text{interactsWith}}\text{Calcium signaling pathway})}$ reaches an agreement for potentially being explanatory (i.e., three answers "maybe") for DILI. This path pattern is relatively complex to interpret since it is long ($k = 3$) and negated. Experts searched for literature reporting associations between DILI and Calcium signaling pathway. They found that a relatively old literature (old is seen as lacking confirmation by some experts) were reporting such an association [36]. A more recent bioinformatics article by Chen et al., also reported such an association, but with a finer grain of information, since they report an association with hepatomegaly (a secondary example of DILI), and a negative association with hepatitis (a primary example of DILI) [37]. Accordingly, negative results from this study are consistent with our finding of this latter negated feature. We note that Chen et al. study is computational, as is ours, and that we may also be impacted by similar bias. Second, $\xrightarrow{\text{interactsWith}}\text{Enzyme}\xrightarrow{\text{biologicalProcess}}\text{Electron transport}$ for SCAR, obtained very diverse opinions, with one "no", one "maybe" and one "yes". Even if this disagreement could be perceived as inconclusive, it may also point to a promising candidate for explanation. In this very one case, Electron transport is known for being perturbed in mitochondrion in many types of ADRs, including hepatotoxicity [38]. However, we did not find any study reporting a link with skin toxicities.

Discussion

In our work, simple, but explainable, classifiers (CART Decision Trees and JRip) were preferred to more advanced machine learning methods, even if we are convinced that methods based on deep neural networks, such as graph embedding with Graph Convolutional Networks (GCN), should lead to better performances than those obtained [39, 40]. However, acquiring explanatory elements about decisions made by such models necessitates an additional step of neural network analysis, such as saliency maps [41], which provides information such as the layer or neurons activated by some instances. We consider this information of high interest for data scientists, but such methods require high level interpretation before being understandable by typical domain experts, unfamiliar with neural networks [42]. Consequently, such direction did not seem mature enough to reach our objectives. For instance, heatmaps that are to some extent explanatory for image classification, are still hard to transpose to knowledge graphs [43]. However, it would be of interest to evaluate performances of a GCN on the classification task to measure the gap caused by our choice of simple classifiers. We also hope that our work will motivate studies on explainable subsymbolic approaches.

Our approach is reproducible for other applications. For instance, the same rational could be applied for the investigation of the mechanism of drug beneficial effects. This would necessitate to change our expert classifications for lists of drugs with a same indication ($\oplus$) and drugs without effect for this indication ($\ominus$), which could be obtained from SIDER or DailyMed.

An objective of our work is to illustrate various advantages of mining knowledge graphs, and particularly Semantic Web ones. First, they provide human-readable features, that may subsequently be interpreted by experts. Second, predictive features may come from various connected data sets and be jointly used in a single rule, which would have not been found if data sets were considered isolated. In addition, using Semantic Web standards eases the addition to our graph of novel data, following other `owl:sameAs` links. Third, Semantic Web knowledge graphs are associated with a formal semantics we benefit from at two steps: at the initial canonicalization, and at the generalization of path patterns. In this regards, one may ask if we could benefit from additional reasoning mechanisms, such as generalization over predicates. In our very specific case, predicates are not associated with any hierarchy, so it would not have changed our results, but from a general point of view, path patterns would benefit from this mechanism. Similarly, we could think of a canonicalization, not only with `owl:sameAs` links, but also following properties carrying similar semantics (e.g., `skos:exactMatch`) or by applying matching approaches such as PARIS [44].

When testing with values between 1 and 3, we observed that larger t and k are associated with better performances. To achieve this, we adopted a rational approach for scaling the mining of RDF knowledge graphs, which is presented in [29]. This approach reaches its limits with $k > 3$ and $t > 3$, but we think that additional optimization in the graph mining algorithm is still possible and would enable going further.

We used only binary features as they are easy to consider as explanations. Other strategies (e.g., counting, relative counting [45]) could also have been considered while maybe hindering the descriptive power of such features. To maximize the descriptive power of candidate features and avoid redundancy, one could use specific metrics (e.g., approaches relying on hierarchies [46] and/or extent of classes [47] of ontologies). Such metrics could also be considered within the decision tree

algorithm, to propose to the algorithm additional features (more or less aggregated according to generalization) that may be associated with best split with regards to the Gini index (or others). This would lead to the consideration of formal knowledge directly in the mining algorithm [21]. Also, we proposed, with our parameter *m*, an hardcoded way of selecting features of interest. However, one may think of an interactive selection by user, following the possibilities offered by an ontology.

A usual difficulty in human annotation or human evaluation is the normalization of expert answers. Despite a 1-h training about the task, the interpretation of negated features has been heterogeneous among experts. One pitfall was to think that if the affirmation of a feature is true, its negation is wrong. This is misleading because it is possible that a feature is explanatory for some examples and that its negation is also explanatory for other examples or in another context. To ensure normalization of negated features, we considered a feature as explanatory if its affirmation or its negation is explanatory to the expert. This change has been considered in a normalisation batch of reviews done in cooperation with experts.

Our review by human experts evaluates how many features highlighted by our approach are relevant (similarly to what Precision measures), but does not evaluate how many relevant features we may miss (similarly to what Recall measures). It would be of interest to ask experts what are features such as pathways, drug targets, cellular functions that are known to be associated with DILI and SCAR ADRs to enable a final comparison. However, establishing an exhaustive list from the state of the art would be complex and time consuming for experts. Text mining approaches could be of interest to guide them in this matter.

Conclusion

We illustrate in this work that life science knowledge graphs provide sufficiently diverse features to enable simple and explainable models to distinguish between drugs that are causative, or not, for two severe ADRs. These features take the form of paths, path patterns or simple neighboring nodes from the graph, which have the advantage, when adequately selected, of being human-readable and interpretable by experts. We quantify through a small-sized human evaluation that such features are not only discriminative, thus predictive for the classification, but also appear to be good candidates for providing explanatory elements of ADR mechanisms. In conclusion, this work illustrates that simple models, fed with diverse and explicit knowledge sources such as those connected in the form of linked open data constitute an alternative to complex models, efficient but hard to interpret. A natural perspective is to combine such rich sources of background knowledge with models that are both highly performing (such as GCN) and interpretable.

Abbreviations

ADR:: Adverse drug reaction; AI:: Artificial intelligence; AUC:: Area under the curve, usually under the ROC curve; DILI:: Drug-induced liver injury; GCN:: Graph convolutional networks; GO:: Gene ontology; LOD:: Linked open data; PGx:: Pharmacogenomics; RAM:: Random access memory; RDF:: Resource description framework; SCAR:: Severe cutaneous adverse reactions; SMILES:: Simplified molecular-input line-entry system; URI:: Uniform resource identifier.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12911-021-01518-6.

Additional file 1. Table S1: Positive JRip rules learned from the DILI expert classification. **Table S2**: Positive JRip rules learned from the SCAR expert classification. **Table S3**: Features associated with DILI extracted from JRip rules, and presented to experts for evaluating their explanatory potential. **Table S4**: Features associated with SCAR extracted from JRip rules, and presented to experts for evaluating their explanatory potential.

Additional file 2. This file includes the results of the manual evaluation performed by three experts about the explanatory character of predictive features.

Acknowledgements

Authors acknowledge participants of the BioHackathon Paris 2018 where the seed of this work was planted, and in particular Miguel Boland and Patryk Jarnot. Authors also acknowledge anonymous reviewers of the Podium Abstract Session at MedInfo 2019 for their encouraging feedback on our positional abstract [48].

Authors' contributions

EB, PM, CB, MST and AC designed the study. EB, PM and FC mapped expert classifications to PGxLOD. EB and PM extracted the features, trained and evaluated the classifiers. EB, PM, MST and AC designed the manual evaluation. AC supervised it. CB, CNC and NP performed the manual evaluation of features. PM and AC were major contributors in writing the manuscript. EB and MST participated to the writing. All authors commented on the manuscript. All authors read and approved the final manuscript.

Funding

The authors acknowledge the French National Research Agency (ANR) for funding PractiKPharma (ANR-15-CE23-0028) and FIGHT-HF (15-RHUS-0004) projects. Funding body did not played any roles in the design of the study, in data collection, analysis, interpretation or in writing the manuscript.

Availability of data and materials

PGxLOD is available at https://pgxlod.loria.fr/. DILIRank is available at https://www.fda.gov/science-research/liver-toxicity-knowledge-base-ltkb/drug-induced-liver-injury-rank-dilirank-dataset. RegiSCAR "Drug notoriety list" is available at http://www.regiscar.org/cht/pdf/Drug%20Notoriety%202015.%20revised%20may%202017.xls. Rules and features generated and analysed during this study are included in the Additional file 1 — supplementary_information.pdf. Answers of the manual evaluation are included in the Additional file 2 — manual_evaluation.xlsx.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Université de Lorraine, CNRS, Inria, LORIA, Nancy, France. [2]Centre d'Investigations Cliniques Plurithématique 1433, Inserm 1116, CHRU de Nancy, Université de Lorraine, Nancy, France. [3]Orange, Belfort, France. [4]Service de santé publique et information médicale, CHU de Saint Etienne, Saint Etienne, France. [5]Sorbonne Université, Inserm, Université Paris 13, LIMICS, Paris, France. [6]Université de Lorraine, Inserm U1256 - NGERE, Nancy, France. [7]Centre Régional de Pharmacovigilance, CHRU of Nancy, Nancy, France. [8]Inria Paris, Paris, France. [9]Centre de Recherche des Cordeliers, INSERM, Sorbonne Université, Université de Paris, Paris, France.

Received: 15 December 2020 Accepted: 5 May 2021
Published online: 26 May 2021

References

1. Ciociola AA, Cohen LB, Kulkarni P, Kefalas C, Buchman A, Burke C, Cain T, Connor J, Ehrenpreis ED, Fang J, et al. How drugs are developed and approved by the FDA: current process and future directions. Am J Gastroenterol. 2014;109(5):620–3. https://doi.org/10.1038/ajg.2013.407.
2. Anonymous: Mechanism matters. Nat Med. 2010;16(4):347. https://doi.org/10.1038/nm0410-347
3. Kamdar MR, Fernández JD, Polleres A, Tudorache T, Musen M. Enabling web-scale data integration in biomedicine through linked open data. NPJ Digit Med. 2019. https://doi.org/10.1038/s41746-019-0162-5.
4. Bonatti PA, Decker S, Polleres A, Presutti V. Knowledge graphs: new directions for knowledge representation on the semantic web (Dagstuhl Seminar 18371). Dagstuhl Rep. 2019;8(9):29–111. https://doi.org/10.4230/DagRep.8.9.29.
5. Berners-Lee T, Hendler J, Lassila O. The semantic web. Sci Am. 2001;284(5):34–43.
6. Barredo Arrieta A, Díaz-Rodrìguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, Garcia S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion. 2020;58:82–115. https://doi.org/10.1016/j.inffus.2019.12.012.
7. Monnin P, Legrand J, Husson G, Ringot P, Tchechmedjiev A, Jonquet C, Napoli A, Coulet A. PGxO and PGxLOD: a reconciliation of pharmacogenomic knowledge of various provenances, enabling further comparison. BMC Bioinform. 2019;20(S(4)):139–113916. https://doi.org/10.1186/s12859-019-2693-9.
8. Trifirò G, Pariente A, Coloma PM, Kors JA, Polimeni G, Miremont-Salamé G, Catania MA, Salvo F, David A, Moore N, Caputi AP, Sturkenboom M, Molokhia M, Hippisley-Cox J, Acedo CD, van der Lei J, Fourrier-Reglat A. Data mining on electronic health record databases for signal detection in pharmacovigilance: Which events to monitor? Pharmacoepidemiol Drug Saf. 2009;18(12):1176–84. https://doi.org/10.1002/pds.1836.
9. Chen M, Suzuki A, Thakkar S, Yu K, Hu C, Tong W. DILIrank: the largest reference drug list ranked by the risk for developing drug-induced liver injury in humans. Drug Discov Today. 2016;21(4):648–53. https://doi.org/10.1016/j.drudis.2016.02.015.
10. RegiSCAR project consortium: Drug Notoriety Classification for ALDEN. http://www.regiscar.org/cht/pdf/Drug%20Notoriety%202015.%20revised%20may%202017.xls. Accessed 9 Oct 2020
11. Ho T-B, Le L, Thai DT, Taewijit S. Data-driven approach to detect and predict adverse drug reactions. Curr Pharmaceut Des. 2016;22(23):3498–526. https://doi.org/10.2174/1381612822666160509125047.
12. Boland MR, Jacunski A, Lorberbaum T, Romano JD, Moskovitch R, Tatonetti NP. Systems biology approaches for identifying adverse drug reactions and elucidating their underlying biological mechanisms. WIREs Syst Biol Med. 2016;8(2):104–22. https://doi.org/10.1002/wsbm.1323.
13. Lee S, Lee KH, Song M, Lee D. Building the process-drug-side effect network to discover the relationship between biological processes and side effects. BMC Bioinform. 2011;12(S-2):2. https://doi.org/10.1186/1471-2105-12-S2-S2.
14. Wallach I, Jaitly N, Lilien R. A structure-based approach for mapping adverse drug reactions to the perturbation of underlying biological pathways. PLoS ONE. 2010;5(8):1–11. https://doi.org/10.1371/journal.pone.0012063.
15. Bresso E, Grisoni R, Marchetti G, Karaboga AS, Souchet M, Devignes M, Smaïl-Tabbone M. Integrative relational machine-learning approach for understanding drug side-effect profiles. BMC Bioinform. 2013;14:207. https://doi.org/10.1186/1471-2105-14-207.
16. Chen X, Shi H, Yang F, Yang L, Lv Y, Wang S, Dai E, Sun D, Jiang W. Large-scale identification of adverse drug reaction-related proteins through a random walk model. Sci Rep. 2016;6:36325. https://doi.org/10.1038/srep36325.
17. Bean D, Wu H, Iqbal E, Dzahini O, Ibrahim Z, Broadbent MTM, Stewart R, Dobson R. Knowledge graph prediction of unknown adverse drug reactions and validation in electronic health records. Sci Rep. 2017;66:7.
18. Kamdar MR, Musen MA. PhLeGrA: graph analytics in pharmacology over the web of life sciences linked open data. In: Proceedings of the 26th international conference on World Wide Web, WWW 2017, Perth, Australia, April 3–7, 2017. ACM, 2017. pp. 321–9. https://doi.org/10.1145/3038912.3052692.
19. Muñoz E, Novácek V, Vandenbussche P. Facilitating prediction of adverse drug reactions by using knowledge graphs and multi-label learning models. Brief Bioinform. 2019;20(1):190–202. https://doi.org/10.1093/bib/bbx099.
20. Dalleau K, Marzougui Y, Da Silva S, Ringot P, Ndiaye NC, Coulet A. Learning from biomedical linked data to suggest valid pharmacogenes. J Biomed Semant. 2017;8(1):16. https://doi.org/10.1186/s13326-017-0125-1.
21. Ristoski P, Paulheim H. Semantic web in data mining and knowledge discovery: a comprehensive survey. J Web Semant. 2016;36:1–22. https://doi.org/10.1016/j.websem.2016.01.001.
22. Shi B, Weninger T. Discriminative predicate path mining for fact checking in knowledge graphs. Knowl-Based Syst. 2016;104:123–33. https://doi.org/10.1016/j.knosys.2016.04.015.
23. Paulheim H. Generating possible interpretations for statistics from linked open data. In: Proceedings of the semantic web: research and applications—9th extended semantic web conference, ESWC 2012, Heraklion, Crete, Greece, May 27–31, 2012. Lecture notes in computer science, vol 7295, 2012. pp. 560–74. https://doi.org/10.1007/978-3-642-30284-8_44.
24. Paulheim H, Fürnkranz J. Unsupervised generation of data mining features from linked open data. In: Proceedings of the 2nd international conference on web intelligence, mining and semantics, WIMS'12, Craiova, Romania, June 6–8, 2012. ACM, 2012. pp. 31–13112. https://doi.org/10.1145/2254129.2254168.
25. Vandewiele G, Steenwinckel B, Ongenae F, De Turck F. Inducing a decision tree with discriminative paths to classify entities in a knowledge graph. In: Proceedings of the 4th international workshop on semantics-powered data mining and analytics co-located with the 18th international semantic web conference (ISWC 2019), Aukland, New Zealand, October 27, 2019. CEUR Workshop Proceedings, vol. 2427 2019. http://ceur-ws.org/Vol-2427/SEPDA_2019_paper_3.pdf.
26. de Vries GKD, de Rooij S. A fast and simple graph kernel for RDF. In: Proceedings of the international workshop on data mining on linked data, with linked data mining challenge collocated with the european conference on machine learning and principles and practice of knowledge discovery in databases (ECMLPKDD 2013), Prague, Czech Republic, September 23, 2013. CEUR workshop proceedings, vol. 1082:2013. http://ceur-ws.org/Vol-1082/paper2.pdf.
27. de Vries GKD, de Rooij S. Substructure counting graph kernels for machine learning from RDF data. J Web Semant. 2015;35:71–84. https://doi.org/10.1016/j.websem.2015.08.002.
28. Sassolas B, Haddad C, Mockenhaupt M, Dunant A, Liss Y, Bork K, Haustein U-F, Vieluf D, Roujeau J-C, Le Louet H. Alden, an algorithm for assessment of drug causality in stevens-johnson syndrome and toxic epidermal necrolysis: comparison with case-control analysis. Clin Pharmacol Therap. 2010;88:60–8. https://doi.org/10.1038/clpt.2009.252.
29. Monnin P, Bresso E, Couceiro M, Smaïl-Tabbone M, Napoli A, Coulet A. Tackling scalability issues in mining path patterns from knowledge graphs: a preliminary study. In: 1st International conference "Algebras, Graphs and Ordered Sets" (Algos 2020), Nancy, France; 2020. https://hal.inria.fr/hal-02913224.
30. Kearns M. Thoughts on hypothesis boosting; 1988 **(unpublished)**.

31. Wang R. AdaBoost for feature selection, classification and its relation with SVM, a review. Phys Procedia. 2012;25:800–7. https://doi.org/10.1016/j.phpro.2012.03.160.
32. Schapire RE. A brief introduction to boosting. In: Proceedings of the 16th international joint conference on artificial intelligence—Volume 2 (IJCAI'99). Morgan Kaufmann, San Francisco; 1999. pp. 1401–6.
33. Cohen WW. Fast effective rule induction. In: Prieditis, A., Russell, S. (eds.) Machine learning proceedings 1995. Morgan Kaufmann, San Francisco; 1995. p. 115–23. https://doi.org/10.1016/B978-1-55860-377-6.50023-2.
34. Neve E, Ingelman-Sundberg M. Cytochrome p450 proteins: retention and distribution from the endoplasmic reticulum. Curr Opin Drug Discov Dev. 2010;13(1):78–85.
35. Ciccacci C, Di Fusco D, Marazzi MC, Zimba I, Erba F, Novelli G, Palombi L, Borgiani P, Liotta G. Association between CYP2B6 polymorphisms and nevirapine-induced SJS/TEN: a pharmacogenetics study. Eur J Clin Pharmacol. 2013;69(11):1909–16. https://doi.org/10.1007/s00228-013-1549-x.
36. Jones BE, Czaja MJ. III. Intracellular signaling in response to toxic liver injury. Am J Physiol. 1998;275(5):874–8. https://doi.org/10.1152/ajpgi.1998.275.5.G874.
37. Chen B, Dong X, Jiao D, Wang H, Zhu Q, Ding Y, Wild DJ. Chem2Bio2RDF: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. BMC Bioinform. 2010;11:255. https://doi.org/10.1186/1471-2105-11-255.
38. Vuda M, Kamath A. Drug induced mitochondrial dysfunction: mechanisms and adverse clinical consequences. Mitochondrion. 2016;31:63–74. https://doi.org/10.1016/j.mito.2016.10.005.
39. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. CoRR abs/1609.02907; 2016.
40. Schlichtkrull MS, Kipf TN, Bloem P, van den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: The Semantic Web—15th international conference (ESWC 2018), Heraklion, Crete, Greece, June 3–7, 2018, Proceedings, 2018. p. 593–607. https://doi.org/10.1007/978-3-319-93417-4_38.
41. Mundhenk TN, Chen BY, Friedland G. Efficient saliency maps for Explainable AI. CoRR abs/1911.11293; 2019.
42. Ying Z, Bourgeois D, You J, Zitnik M, Leskovec J. GNNExplainer: generating explanations for graph neural networks. Adv Neural Inf Process Syst. 2019;32:9244–55.
43. Montavon G, Samek W, Müller K-R. Methods for interpreting and understanding deep neural networks. Digi Signal Process. 2018;73:1–15. https://doi.org/10.1016/j.dsp.2017.10.011.
44. Suchanek FM, Abiteboul S, Senellart P. PARIS: probabilistic alignment of relations, instances, and schema. PVLDB. 2011;5(3):157–68. https://doi.org/10.14778/2078331.2078332.
45. Ristoski P, Paulheim H. A comparison of propositionalization strategies for creating features from linked open data. In: Proceedings of the 1st workshop on linked data for knowledge discovery co-located with European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD 2014), Nancy, France, September 19th, 2014. CEUR Workshop Proceedings, 2014; vol. 1232 http://ceur-ws.org/Vol-1232/paper1.pdf.
46. Ristoski P, Paulheim H. Feature selection in hierarchical feature spaces. In: Proceedings of discovery science—17th international conference, DS 2014, Bled, Slovenia, October 8-10, 2014, Lecture notes in computer science, vol. 8777; 2014. pp. 288–300. https://doi.org/10.1007/978-3-319-11812-3_25.
47. d'Amato C, Staab S, Fanizzi N. On the influence of description logics ontologies on conceptual similarity. In: Knowledge engineering: practice and patterns, 16th international conference (EKAW 2008), Acitrezza, Italy, September 29–October 2, 2008. Proceedings. Lecture notes in computer science, vol. 5268; 2008. pp. 48–63. https://doi.org/10.1007/978-3-540-87696-0_7.
48. Calvier F.-É, Monnin P, Boland M, Jarnot P, Bresso E, Smaïl-Tabbone M, Coulet A, Bousquet C. Providing molecular characterization for unexplained adverse drug reactions. Podium Abstract at MedInfo 2019, Lyon, France; 2019. https://hal.inria.fr/hal-02196134.

Publisher's Note