BIOINFORMATICS

Vol. 26 ECCB 2010, pages i547–i553
doi:10.1093/bioinformatics/btq382

Discovering drug–drug interactions: a text-mining and reasoning approach based on properties of drug metabolism

Luis Tari[1,*], Saadat Anwar[2], Shanshan Liang[2], James Cai[1] and Chitta Baral[2]

[1]Disease and Translational Informatics, Hoffmann-La Roche, Nutley, NJ 07110 and [2]Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA

ABSTRACT

Motivation: Identifying drug–drug interactions (DDIs) is a critical process in drug administration and drug development. Clinical support tools often provide comprehensive lists of DDIs, but they usually lack the supporting scientific evidences and different tools can return inconsistent results. In this article, we propose a novel approach that integrates text mining and automated reasoning to derive DDIs. Through the extraction of various facts of drug metabolism, not only the DDIs that are explicitly mentioned in text can be extracted but also the potential interactions that can be inferred by reasoning.

Results: Our approach was able to find several potential DDIs that are not present in DrugBank. We manually evaluated these interactions based on their supporting evidences, and our analysis revealed that 81.3% of these interactions are determined to be correct. This suggests that our approach can uncover potential DDIs with scientific evidences explaining the mechanism of the interactions.

Contact: luis.tari@roche.com

1 INTRODUCTION

One of the challenges in drug administration and drug development is the need to identify and avoid potential drug–drug interactions (DDIs). When several drugs are being administered, there is a possibility of adverse drug reactions as one drug can increase or decrease the effect of another drug. During the course of drug development, it is vital to identify possible drug interactions with the new drug compound that is being investigated. Revealing the mechanism behind the drug interactions can provide the necessary scientific evidences for further investigation of the drug compound. The ability to find DDIs and their scientific evidences efficiently and economically can have an impact on the current approach in drug administration and drug development.

A common source of finding DDIs is through the use of commercial databases such as the Thomson Micromedex DrugDex System (Micromedex) or online freely available drug databases such as DrugBank (Wishart *et al.*, 2006). While such databases provide extensive lists of DDIs, it is typical that scientific evidences are not provided as part of the information for DDIs. This becomes an issue when a recent study showed that there are ∼25% of disagreements between two drug compendia (Wong *et al.*, 2008). The lack of scientific evidences complicates the process of verifying the discrepancies. An ideal resource not only should provide a comprehensive list of DDIs in a cost-efficient manner but also the mechanism behind the interactions. Text mining of Medline abstracts is a logical choice to identify DDIs, as much of the scientific information is frequently available in biomedical articles. In addition, text mining has the potential of finding large number of DDIs from text in a cost-effective manner.

In this article, we propose a novel approach of discovering DDIs through the integration of 'biological domain knowledge' with 'biological facts' from Medline abstracts and curated sources. Our work is one of the few that utilizes biological domain knowledge in their applications, such as the generation of metabolic networks based on stoichiometric constraints (Mavrovouniotis *et al.*, 1990), hypothesis generation in signaling pathways (Tran *et al.*, 2005) and synthesis of pharmacokinetic pathways (Tari *et al.*, 2010). In this work, the biological domain knowledge includes the mechanism of how drug *A* influences the effect of drug *B* through the induction or inhibition the enzymes that are responsible for the metabolism of *B*. With the domain knowledge, the other component is to find the biological facts to support specific DDIs. Examples of biological facts from Medline abstracts include the identification of which enzymes are induced or inhibited by drugs as well as the enzymes that are responsible for the metabolism of drugs. Curated sources such as UniProt (http://www.uniprot.org) and Gene Ontology (GO) annotations (http://www.geneontology.org) are utilized to identify protein families such as enzymes and transcription factors. Integrating the biological domain knowledge with the biological facts allows us to derive DDIs. This step is achieved through the use of logic representation of the domain knowledge and automated reasoning. It is important to note that our approach differs from typical extraction approach, which focuses on the extraction of interactions that are stated explicitly in the text. In our approach, various properties can be extracted from different publications so that the system utilizes the extracted properties to derive potential DDIs. This enables us to discover potential DDIs that are yet to be annotated in comprehensive resources about DDIs such as DrugBank.

The rest of the article is outlined as follows. We describe the basic properties of DDIs that we encode in our system in Section 2. In Section 3, the processes of acquiring the necessary facts and interactions from existing knowledge bases and Medline abstracts are described. In addition, the reasoning component is illustrated in how the properties of drug metabolism are encoded in order to derive DDIs. In Section 4, we describe the possible scenarios of applying our approach to drug administration and drug design. We demonstrate the feasibility of our approach by illustrating the

*To whom correspondence should be addressed.

correctness of the DDIs derived by our approach in Section 5. We conclude in Section 6.

2 DDIs

Common DDIs involve how drugs are metabolized by the body. Metabolism-based interactions mainly happen in the liver as most of the drugs are metabolized by enzymes that reside in the liver. A drug is expected to be eliminated by the body within a certain amount of time after being taken. In case when the drug elimination process takes longer than expected, toxicity can be accumulated or the pharmacological effect of the drug can be exaggerated. The induction or inhibition of enzymes can affect the drugs directly or indirectly through transcriptional regulation. If the enzymes that are responsible for the metabolism of drug *A* get inhibited or induced by other drugs, then the bioavailability of drug *A* will be higher or lower than expected, rendering it toxic or less effective.

Inhibition of enzymes is a common form of DDIs (Boobis, 2009). Such kind of 'direct inhibition' happens when drug *A* inhibits enzyme *E*, which is responsible for metabolism of drug *B*. Such drug interactions lead to the decrease of the activity level of enzymes, and this in turn may increase the bioavailability for the drug *B*. Alternatively, this may reduce the formation of metabolites of the inhibited enzyme and leads to therapeutic failure of the affected drug. An example of such direct inhibition is the interaction between quinidine and CYP2D6 substrates such as codeine, as quinidine is responsible for the inhibition of CYP2D6. On the other hand, CYP2D6 substrates such as codeine are metabolized by CYP2D6. The inhibition of CYP2D6 by quinidine increases the bioavailability of drugs metabolized by CYP2D6. Such increase can potentially lead to adverse side effects of the affected drug.

Another form of drug interactions is through the induction of enzymes (Boobis, 2009). One form of induction is known as *direct induction* when drug *A* induces enzyme *E*, which is responsible for metabolism of drug *B*. An example of such direct induction is between warfarin and phenobarbital. Such drug interaction occurs due to the fact that warfarin is metabolized by CYP2C9, while CYP2C9 is subject to induction by phenobarbital. This leads to the increase of enzyme activity of CYP2C9, which increases the rate of metabolism of warfarin by CYP2C9. Such increase of metabolism decreases the bioavailability of warfarin. While direct induction is possible, it is not the most common form of drug interactions due to induction. A more common form is through transcription factors that regulate the drug-metabolizing enzymes. An alternative form is *indirect induction* through transcription factors. Such interaction occurs when drug *A* activates transcription factor 'TF', which regulates and induces enzyme *E*, and enzyme *E* is responsible for the metabolism of drug *B*. Such transcription factors are referred as regulators for xenobiotic-metabolizing enzymes. Examples of such regulators are aryl hydrocarbon receptor AhR, pregnane X receptor PXR and constitutive androstane receptor CAR. Figure 1 illustrates how the induction or inhibition of transcription factors and enzymes contribute to drug interactions.

Other than metabolism-based interactions, drug interactions can also occur due to the induction or inhibition of transporters. Transporters are mainly responsible for cellular uptake or cellular efflux of drugs. Transporters play an important role in drug disposition as drugs can only be metabolized after they are transported into the liver cells. However, transporter-based drug interactions are not as well studied as metabolism-based interactions (Boobis, 2009). In this article, we emphasize on the extraction of metabolism-based drug interactions.

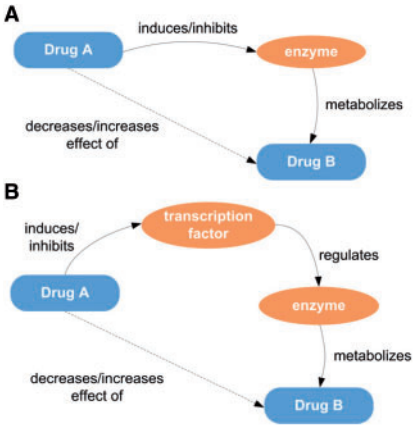

Fig. 1. The effects of drug A on drug B through (**A**) direct induction/inhibition of enzymes; (**B**) indirect induction/inhibition of transcription factors that regulate the drug-metabolizing enzymes.

3 METHODS

Our approach in identifying DDIs can be described in two phases: (i) natural language extraction phase; (ii) reasoning phase. We start with performing an offline one-time parse of the documents and store a structured representation of unstructured text in the form of syntactic structures into a database called the 'parse tree database'. Such structured information includes grammatical structures of sentences known as 'parse trees' and the biological entities involved in the sentences. The stored information of the sentences are used for extraction by means of database queries in the form of 'parse tree query language' (PTQL). PTQL (Tu *et al.*, 2008) is a flexible query language that is designed for extracting various kinds of relations. While these extracted relations provide the necessary knowledge to identify DDIs, the extracted relations alone are not sufficient to derive DDIs. By 'representing the general knowledge about drug metabolism and interactions' in the form of logic rules, the extracted relationships are then applied to the logic rules in order to reveal DDIs in the 'reasoning phase'. An overview of our approach is illustrated in Figure 2. Here we describe the main components of our approach in details.

3.1 Parse tree database and PTQL

A parse tree is composed of a constituent tree and a linkage. A constituent tree is a syntactic tree of a sentence with the nodes represented by part-of-speech tags and leafs corresponding to words in the sentence. A linkage represents the syntactic dependencies (or links) between pairs of words in a sentence. The parse trees are produced by the Link Grammar parser (Sleator and Temperley, 1993), while BANNER (Leaman and Gonzalez, 2008) is used to recognize gene/protein names and MetaMap (Aronson, 1996) for drug names. To disambiguate gene mentions, GNAT (Hakenberg *et al.*, 2008) is applied to identify the official gene symbols for each gene mentions identified by BANNER. The syntactic and semantic information of sentences are stored in our parse tree database and extraction is performed by means of database queries. By storing the syntactic and semantic information, this avoids the need of reprocessing the document collection every time extraction is performed. This also allows on-the-fly extraction that is suitable for extracting various kinds of interactions as needed in the purpose of identifying drug interactions.

Our parse tree database is managed by MySQL containing ~17 million Medline abstracts. Due to the hierarchical nature of the syntactic and

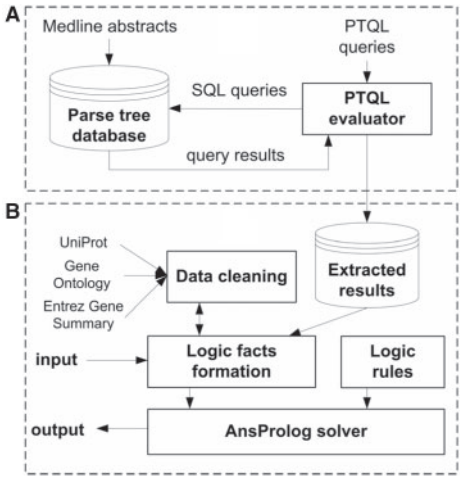

Fig. 2. An overview of our approach in extracting drug interactions. (**A**) In the 'extraction phase', queries are applied to the parse tree database for the extraction of various interactions. (**B**) By utilizing the extracted interactions, the 'reasoning phase' applies the interactions to the logic rules to derive drug–drug interactions.

semantic information of sentences, writing SQL queries for extraction can be complicated with the use of numerous table joins. Our approach is to develop a query language called PTQL that enables generic extraction of relations. A PTQL query is made up of four components: (i) tree patterns; (ii) link conditions; (iii) proximity conditions; and (iv) return expression. The components in a PTQL query are separated by the symbol ':'. We illustrate PTQL queries with examples in the later sections. In the extraction phase, the 'PTQL query evaluator' takes PTQL queries and transforms them into SQL queries. The translated SQL queries are evaluated against the parse tree database so that the returned information is the intended interactions from abstracts.

3.2 Natural language extraction

Our strategy of extraction can be categorized as explicit and implicit extraction of DDIs from text. 'Explicit extraction' refers to the extraction of a DDI that is described within a sentence, while 'implicit extraction' requires the extraction of various properties of drug metabolism that can lead to a DDI. These properties include the identification of protein families and the interactions that are involved in drug metabolism. Such information is stored in the database tables.

3.2.1 Extraction of explicit drug interactions Description of DDIs can sometimes be found in Medline abstracts. Being able to extract such explicit description of interactions between drugs is essential. The following sentences are example sentences from Medline abstracts that describe drug interactions within the individual sentences.

- S1: 'Ciprofloxacin' strongly inhibits 'clozapine' metabolism (PMID: 19067475).
- S2: Enantioselective induction of 'cyclophosphamide' metabolism by 'phenytoin' (PMID: 10423284).

We used two extraction queries that utilize the keywords 'induce', 'induces', 'induction' to extract how one drug decreases the effect of another drug. Similarly, we used the keywords 'inhibit', 'inhibits', 'inhibition' to find the increase of the effect of one drug by another drug. Here we illustrate below one of the queries as PTQL query.

```
//S{//?[Tag='Drug'](d1) =>
    //?[Value IN {'induce', 'induces', 'inhibit',
    'inhibits'}](v) => //?[Tag='Drug'](d2) =>
    //?[Value= 'metabolism'](w)} :: [d1 v d2 w]5 :
    d1.value, v.value, d2.value
```

Table 1. Triplets representing various properties relevant to the extraction of implicit drug interactions and their description

<*actor1*, *relation*, *actor2*>	Description
<*p*, *metabolizes*, *d*>	Drug *d* is metabolized by protein *p*
<*d*, *inhibits*, *p*>	Drug *d* inhibits the activity of protein *p*
<*d*, *induces*, *p*>	Drug *d* induces the activity of protein *p*
<*p*1, *regulates*, *p*2>	Protein *p*1regulates the activity of protein *p*2

The tree pattern of the above query specifies that a drug (denoted as `d1`) has to be followed by (denoted as the operator =>) one of the keywords 'induce', 'induces', 'inhibit', 'inhibits', which is then followed by another drug (denoted as `d2`) with the keyword 'metabolism'. The proximity condition `[d1 v d2 w]5` indicates that these words have to appear within five words of each other in the matching sentences. The return expression `d1.value`, `v.value`, `d2.value` specifies the return of the values for the triplet <*d*1, *v*, *d*2>. The triplet <*d*1, *induce/induces*, *d*2> implies that drug *d*1 *decreases* the effect of drug *d*2 (i.e. <*d*1, *decreases*, *d*2>), while <*d*1, *inhibit/inhibits*, *d*2> for the *increase* of the effect of *d*2 by *d*1(i.e. <*d*1, *increases*,*d*2>). With the PTQL query, sentence S1 matches the criteria of the query and the triplet <ciprofloxacin, increases, clozapine> is formed.

3.2.2 Extraction of implicit drug interactions There are cases when the experimental findings of drug interactions have yet to be published but they can be inferred based on various properties of drug metabolism. It is important to notice that such properties can be originated from different publications. Our approach is to extract the relevant properties and utilize the extracted interactions to infer drug interactions through automated reasoning.

Table 1 illustrates the various kinds of interactions that are necessary to infer drug interactions. Different PTQL queries are used for extraction. As an example, the following PTQL query is used to extract <*protein*, *metabolizes*, *drug*> triplets:

```
//S{/?[Tag='DRUG'](kw2) => //VP{//?[Value IN
   {'metabolised','metabolized'}](kw1) =>
     //?[Tag='GENE'](kw0)}} :::
       kw0.value, kw1.value, kw2.value
```

The tree pattern of the above query specifies that a drug (denoted as `kw2`) is followed by a verb phrase (denoted as `VP`), such that the verb phrase includes one of the keywords 'metabolized' or 'metabolized' and it is followed by a gene mention (denoted as `kw0`). The return expression ensures that the triplets <*protein*, *metabolizes*, *drug*> are returned from the matching sentences. Table 2 illustrates the sample triplets and their supporting evidences for the extraction of various kinds of interactions.

3.3 Knowledge representation and reasoning

As the interactions themselves do not reveal any kind of ordering, the goal is to represent the fundamental behavior and properties of pharmacokinetics so that the representation can be utilized to assign ordering of the interactions through reasoning. Implementation of the reasoning component requires a language that is ideal in specifying what kind of reasoning to be performed rather than how the reasoning is performed. This is analogous to declarative programming language such as SQL, in which the users specify what is intended to be found rather than how the search mechanism of the database system should be performed to answer the queries. AnsProlog (Gelfond and Lifschitz, 1988, 1991) is a declarative language that is useful for reasoning, as well as capable for reasoning with incomplete information. It is important

Table 2. Sample extracted interactions for each kind of relations and their support evidences

PMID	Evidence and extracted interaction
8689812	Lovastatin is metabolized by CYP3A4. <*cyp3a4*, *metabolizes*, *lovastatin*>
8477556	Inhibition by fluoxetine of cytochrome P450 2D6 activity. <*fluoxetine*, *inhibits*, *cyp2d6*>
10678302	Phenytoin induces CYP2C and CYP3A4 isoforms, but not CYP2E1. <*phenytoin*, *induces*, *cyp2c*>, <*phenytoin*, *induces*, *cyp3a4*>
11502872	The CYP2B6 gene is directly regulated by PXR <*pxr*, *regulates*, *cyp2b6*>

Table 3. Logic facts transformed from the extracted interactions in Table 1 after data cleaning

Logic facts and their description
`metabolized(d, p)`: transformed from <*p*, *metabolizes*, *d*> to represent that drug *d* is metabolized by enzyme *p*
`inhibits(d, p)`: transformed from <*d*, *inhibits*, *p*> to represent that drug *d* inhibits the activity of enzyme *p*
`induces(d, p)`: transformed from <*d*, *induces*, *p*> to represent that drug *d* induces the activity of enzyme *p*
`regulates(p1, p2)`: transformed from <*p*1, *regulates*, *p*2> to represent that transcription factor *p*1 regulates the activity of enzyme *p*2

to notice that AnsProlog is a declarative language different from Prolog. While Prolog is a programming language with roots in logic, it includes many non-logical features that are not declarative, making it unsuitable for knowledge representation. Here we give a brief introduction to the syntax of AnsProlog.

An *AnsProlog rule* is of the form:

$$l \text{ :- } l_0, \ldots, l_m, \textbf{not } l_{m+1}, \ldots, \textbf{not } l_n.$$

where l_is are literals and **not** represents negation as failures. The intuitive meaning of the above rule is that if it is known that literals l_0, …, l_m are to be true and if l_{m+1}, …,l_n can assume to be false, then l must be true. A literal is defined as either an atom or an atom preceded by the symbol ¬ that indicates classical negation. If there is no literal l in the head of a rule, then the rule is referred as a *constraint*. An *answer set program* is composed of a set of AnsProlog rules, and the interpretation of an answer set program is called *answer sets*. Readers can refer to Baral (2003) for more details on the syntax and semantics of AnsProlog.

3.3.1 Formation of logic facts To utilize the extracted interactions for reasoning purpose, each of them has to be first translated into their logic forms. The extracted interactions in the form of triplets <*actor1*, *relation*, *actor2*> as in Table 3 are represented as `extr(actor1, relation, actor2)`. For instance, the triplet <*protein*, *metabolizes*, *drug*> is represented as `extr(Protein, metabolizes, Drug)`, where Enzyme and Drug are variables for the domain enzyme and drug. For instance, the triplet <cyp3a4, metabolizes, lovastatin> is represented by the logic facts `extr(cyp3a4, metabolizes, lovastatin)`, `protein(cyp3a4)` and `drug(lovastatin)`.

3.3.2 Data cleaning In the extraction phase, drug-protein interactions such as <*protein*, *metabolizes*, *drug*> and protein–protein interactions such as <*protein*, *regulates*, *protein*> are extracted from the parse tree database. However, it is necessary to include an extra step to ensure that these extracted interactions correspond to the properties of drug metabolism before they can be applied to derive DDIs. For instance, the proteins involved in the triplet <*protein*, *metabolizes*, *drug*> have to be enzymes in order for the drug to be metabolized. Therefore, the extracted relations have to be refined into <*enzyme*, *metabolizes*, *drug*> for the triplet <*protein*, *metabolizes*, *drug*>. Likewise, the triplet <*protein*, *regulates*, *protein*> has to be refined into <*transcription factor*, *regulates*, *enzyme*>. Such refinement for the extracted interactions requires a process called 'data cleaning'. This includes identifying the protein families for the proteins involved in the extracted interactions and extracting negative interactions. Such information is utilized to refine the interactions by means of AnsProlog rules. We first describe the details of identifying protein families using UniProt, the GO and Entrez Gene summary.

A protein *p* is considered as an 'enzyme' if either one of the following holds (in the given order of precedence):

- *p* belongs to the CYP, UGT or SULT gene families, i.e. official gene symbol starts with CYP, UGT or SULT;
- *p* is annotated under UniProt as having keywords 'hydrolase', 'ligase', 'lyase' or 'transferase';
- *p* is annotated under the GO term 'metabolic process' (GO: 0008152);
- the Entrez Gene summary (provided by RefSeq) of *p* contains the key phrase 'drug metabolism' or the regular expression 'enzyme*' or 'catalyz*'.

A protein *p* is considered as a 'transcription factor' if (in the given order of precedence):

- *p* is annotated under UniProt as having keywords 'transcription', 'transcription-regulator' or 'activator';
- *p* is annotated under the GO term 'transcription factor activity' (GO: 0003700);
- the Entrez Gene summary of *p* contains the keyword 'transcription factor'.

Enzymes and transcription factors are represented in the form of `enzyme(Protein)` and `transcription_factor(Protein)`, where `Protein` is a variable for the domain 'protein'. For instance, enzyme CYP3A4 is represented as `enzyme(cyp3a4)` and `protein(cyp3a4)`.

In the extraction of interactions, there are times that negative interactions are extracted. Consider the following sample sentence:

S3. … oxybutynin is predominantly metabolized by CYP3A4 and CYP3A5 but not by CYP2D6 (PMID: 9584328).

By considering the negation words such as 'not' in the extraction queries, the negative interaction <CYP2D6, not_metabolizes, oxybutynin> is extracted.

With the identification of protein families and extraction of negative interactions, we perform data cleaning on the extracted interactions by using AnsProlog logic rules to refine the interactions. We used the following AnsProlog rule to illustrate the idea of data cleaning.

```
metabolized(D, P) :- extr(P, metabolizes, D),
    drug (D), enzyme(P),
    not extr(P, not_metabolizes, D).
```

The above AnsProlog logic rule refines the <*protein*, *metabolizes*, *drug*> relations in the form of `extr(P, metabolizes, D)` into <*enzyme*, *metabolizes*, *drug*> in the form of `metabolized(D, P)`, provided that *D* is a drug and *P* is known to be an enzyme, which are represented as `drug(D)` and `enzyme(P)`. Another condition for such refinement is that it is not known to have a negative interaction for this particular interaction among the extracted interactions. This condition is enforced by `not extr(P, not_metabolizes, D)`. Similar data cleaning logic rules are written to

achieve the logic facts from their corresponding extracted interactions, as illustrated in Table 3.

Using sentence S3 as an example, the logic rule turns the extracted interactions `extr(cyp3a4, metabolizes, oxybutynin)` and `extr(cyp3a5, metabolizes, oxybutynin)` into the logic facts `metabolized(oxybutynin, cyp3a4)`, `metabolized(oxybutynin, cyp3a5)`, provided that logic facts `enzyme(cyp3a4)` and `enzyme(cyp3a5)` are stored in the database of extracted results. On the other hand, `extr(cyp2d6, metabolizes, oxybutynin)` cannot be turned into `metabolized(oxybutynin, cyp2d6)` if `extr(cyp2d6, not_metabolizes, oxybutynin)` is among the interactions.

3.3.3 Representing knowledge on drug metabolism We encode the rules that represent various properties of drug metabolism and how the properties lead to DDIs. Rule 1 encodes the effect of the activity of a protein through the induction by a drug. On the other hand, Rule 2 describes the effect of the activity of a protein that is regulated by a transcription factor, which in turns induced by a drug. Rules 1 and 2 describe the necessary steps involved in direct and indirect DDIs. The decrease of the effects of a drug by another drug is determined by Rule 3. Similar rules are written for changes in the activity of proteins due to inhibition by a drug.

- Rule 1: Changes in the activity of protein *P* due to induction by a drug *Dr*

```
affects(Dr, level(P, high)) :-
 induces(Dr, P), drug(Dr), protein(P).
```

- Rule 2: Changes in the activity of protein *P* due to induction of the regulating transcription factor *TF* by drug *Dr*

```
affects(Dr, level(P, high)) :-
  affects(Dr, level(TF, high)), protein(P),
  regulates(TF, P), drug(Dr),
  transcription_factor(TF).
```

- Rule 3: Drug interactions through induction of enzymes

```
result(Dr1, decreases, Dr2) :-
  affects(Dr1, level(P, high)), enzyme(P),
  metabolized(Dr2, P), drug(Dr1), drug(Dr2).
```

3.3.4 Reasoning Given a set of drugs *D* of interest, the goal is to find interactions among *D*. This involves the extraction of various facts as listed in Table 1. These facts are represented in the form of logic facts as described in Section 3.3.1. The reasoning phase involves the use of data cleaning as described in Section 3.3.2 together with the rules describing the properties involved in drug metabolism as described in Section 3.3.3. An AnsProlog solver called clingo (Gebser *et al.*, 2009) is then utilized to compute the answer sets that infer the DDIs among *D*.

4 SCENARIOS

We illustrate our system with scenarios for drug administration and drug development.

4.1 Drug administration

We illustrate how our system can be used to oversee drug administration. Suppose a patient is about to be prescribed with two drugs phenytoin and gefitinib, a medical doctor or pharmacist needs to know if these two drugs can trigger any interactions.

Among the extracted interactions, the triplets <phenytoin, induces, cyp3a4> and <cyp3a4, metabolizes, gefitinib> are supported by the following evidences.

- Phenytoin induces CYP2C and CYP3A4 isoforms, but not CYP2E1 (PMID: 10678302).
- Gefitinib is extensively metabolized in the liver by cytochrome P450 3A4 enzyme (PMID: 14977817).

With the fact `enzyme(cyp3a4)`, the data cleaning rule is triggered and produces the following logic facts:

- `metabolized(gefitinib, cyp3a4)`
- `induces(phenytoin, cyp3a4)`

The logic fact `induces(phenytoin, cyp3a4)` triggers one of the logic rules about the effects of the expression level of an enzyme. This results in the following logic fact.

- `affects(phenytoin, level(cyp3a4, high))`

The logic facts in turn trigger one of the drug interaction rules, and the system indicates that the drug phenytoin decreases the effect of gefitinib with the following logic fact.

- `result(phenytoin, decreases, gefitinib)`

The resulting logic fact prompts the medical doctor or pharmacist that phenytoin and gefitinib can have potential drug interaction. It is interesting to note that this phenytoin–gefitinib interaction is also annotated in DrugBank, but our approach provides the scientific evidences to explain the mechanism behind the interaction.

4.2 Drug development

Identifying potential drug interactions with the new chemical compound in hand is critical in the drug development process. Suppose we know that the new chemical compound we are investigating is known to be an inhibitor for an enzyme. With this information, we want to identify what known drugs can be affected by this new compound. Suppose our new compound (denoted as `new_drug`) is a CYP3A4 inhibitor, we represent such information in the form of logic facts as the input to our system.

- `inhibits(new_drug, cyp3a4)`

This fact triggers one of the logic rules about the effects of the enzyme activity. In this case, the expression level of CYP3A4 becomes low due to the influence of the CYP3A4 inhibitor. This results in the following logic fact.

- `affects(new_drug, level(cyp3a4, low))`

Among the extracted interactions, terfenadine is found to be one of the drugs that are metabolized extensively by CYP3A4. This leads to the logic fact `extr(terfenadine, metabolized, cyp3a4)`. Together with the fact `enzyme(cyp3a4)` and no other logic facts in the database of extracted interactions indicates that terfernadine is not metabolized by CYP3A4, these trigger one of the data cleaning rules and the following logic fact is formed.

- `metabolized(terfernadine, cyp3a4)`

The newly formed facts `affects(new_drug, level(cyp3a4,low))` and `metabolized(terfernadine, cyp3a4)` in turn trigger one of the drug interaction rules, and the system indicates that the new CYP3A4 inhibitor may increase the effect of terfenadine with the following logic fact.

Table 4. Correctness of the DDIs for the extraction of explicit DDIs, implicit DDIs as a result of direct inhibition/induction and indirect inhibition/induction of enzymes

Relations	Correctness based on supporting evidences, % (*n*)	Overlap with drugbank, % (*n*)
Explicit DDIs	77.7 (132/170)	11.8 (20/170)[a]
Implicit direct DDIs	81.3 (256/315)	2.60 (108/4154)[a]
Implicit indirect DDIs	100 (30/30)	1.5 (15/979)[a]

[a]Represent the number of interactions that match with 'DrugBank gold standard'. The unmatched interactions are verified manually based on their supporting evidences in the second column.

Table 5. Performance of the extracted interactions from 13K Medline abstracts with number of true positives (denoted as number of TP) and false negatives (denoted as number of FN)

Relations	Precision (number of TP), % (*n*)	Recall (number of FN), % (*n*)	*F*-measure, %
<*protein*, *metabolizes*, *drug*>	93.1 (54)	26.7 (148)	41.5
<*drug*, *induces*, *protein*>	61.8 (42)	30.7 (95)	41.0
<*drug*, *inhibits*, *protein*>	58.6 (99)	48.5 (105)	53.1
<*protein*, *regulates*, *protein*>	68.7 (46)	100.0 (0)	81.4
negation	84.4 (38)	–	–

- `result(new_drug, increases, terfenadine)`

This scenario shows that our system can not only alert the investigators for DDIs, but also the mechanism behind the interactions so that the clinical trials can be adjusted accordingly.

5 RESULTS

We used DrugBank to assess the performance of our approach in finding DDIs. We selected 265 drugs and found the interactions among these drugs. This results in a gold standard of 494 DDIs with description stating that one drug increases or decreases the effect of another drug. We call these set of drug interactions as the 'DrugBank gold standard'.

We first evaluated the performance of the extraction of explicit DDIs. The results are summarized in Table 4. An explicit drug interaction is extracted from a single sentence in a Medline abstract. Our extraction queries for explicit drug interactions are specific to the 265 drugs that we selected as the DrugBank gold standard, and the extraction queries result in 170 drug interactions. We found that 132 of the extracted drug interactions (i.e. a precision of 77.7%) are indeed correct based on their originating sentences, and 20 of them (i.e. an overlap of 11.8%) are annotated in the DrugBank gold standard. The results indicate that there is a potentially large number of published drug interactions that are not captured by DrugBank. On the other hand, the small overlap between our extracted drug interactions and the interactions in the DrugBank gold standard also indicates that there is room for improvement in extracting drug interactions.

With the extraction of implicit DDIs, we expect to achieve a higher number of inferred interactions and overlap with the DrugBank gold standard. The extraction of implicit DDIs is divided into two parts: (i) direct inhibition or induction of enzymes; (ii) indirect inhibition or induction of enzymes. The extraction of implicit DDIs of (i) and (ii) from 17 million Medline abstracts result in 4154 and 979 interactions. Among the extracted interactions in (i), 108 interactions coincide with the interactions in the DrugBank gold standard. For the ones that are not found in the DrugBank gold standard, we manually evaluated the interactions based on their supporting evidences. In particular, we evaluated 315 interactions that are supported by at least two evidences for <*drug*, *induces/inhibits*, *protein*> relations and four evidences for <*protein*, *metabolizes*, *drug*> relations. We realized that 256 or 81.3% of the interactions are extracted with the correct evidences to support the DDIs. Among the 256 correct interactions, CYP3A4 is involved in 171 of them, which is reasonable as CYP3A4 is known to be involved in the metabolism of a majority of drugs. Most of the falsely inferred DDIs are due to the extraction of incorrect <*drug*, *induces*, *protein*> and <*drug*, *inhibits*, *protein*> relations. On the other hand, we selected interactions for (ii) that are supported by at least two evidences for each of the underlying relations. We realized that these 30 interactions are supported by correct evidences, and 21 of the interactions involved the change of effect of CYP3A4 substrates through the induction of CAR (NR1I3) and the regulation of CYP3A4 by CAR.

As our approach in finding DDIs heavily relies on the extraction of relevant interactions, we first evaluated the correctness of the interactions extracted from 13K Medline abstracts. We first created a gold standard for each of the interaction specified in Table 1. As an example, in the case of creating the gold standard for <*protein*, *metabolizes*, *drug*> relations, we extracted sentences with co-occurrences of drug and gene mentions together with one of the keywords 'metabolized', 'metabolize', 'metabolized', 'metabolize', 'substrate' in the individual sentences. We then examined which of the sentences indeed indicate <*protein*, *metabolizes*, *drug*> relations. This results in 372 evidence sentences that we use as a gold standard for evaluating <*protein*, *metabolizes*, *drug*> relations. It is important to note that in the creation of gold standard, we omit possible cross-sentence relations. In addition, we relied on the named entity recognizers to identify sentences with co-occurrences of drug and gene mentions.

The extraction performance for each individual interactions are illustrated in Table 5. In the case of <protein, regulates, protein> relations, our strategy is to extract co-occurrences of CYP enzymes and one of the transcription factors known for regulating xenobiotic enzymes (i.e. AhR, NR1I2, NR1I3). This restricted co-occurrence query allows us to find high-quality relations for <*transcription factors*, *regulates*, *enzymes*>. The negation queries are for the extraction of negative relations for the various interactions listed in Table 1. Keywords such as 'no', 'not' are included in these queries. We only evaluated the precision of the extraction of negative interactions by analyzing each of the extracted relation and its supporting evidence without creating a gold standard.

6 DISCUSSION

We described a novel approach in combining text mining and automated reasoning techniques to find DDIs from Medline

abstracts. We demonstrated that our extraction approach has the distinct capability of performing generic extraction. Unlike EDGAR (Rindflesch *et al.*, 2000) and Pharmspresso (Garten and Altman, 2009) that are capable of extracting a variety of drug-gene relationships, our approach utilizes a variety of extracted drug-gene and protein–protein interactions to infer DDIs that are not explicitly stated in text through automated reasoning. By representing the biological phenomenon of DDIs in a logical representation, our work relies on automated reasoning to infer DDIs rather than the heuristic network approaches used in Garten *et al*. (2010) and Masataka (2008) to infer drug–gene relationships and DDIs. Our results show that our approach can identify promising DDIs and can be treated as a complement to existing databases such as DrugBank but with the addition of scientific evidences to support our inferred interactions. The correctness of the extracted DDIs shows that our approach is readily applicable to real-world scenarios.

In this article, we focused on the enzyme-based DDIs. We will expand our approach to include transporter-based DDIs. Other biological factors can influence the likelihood of DDIs. This includes the degree of metabolizing enzymes on drugs, such as whether a drug is strongly or weakly metabolized by an enzyme. In addition, genetic information should also be considered in identifying DDIs. Improving our extraction for various interactions such as expanding the choices of keywords, can likely increase the percentage of overlap with the interactions annotated in DrugBank.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their comments.

Funding: Grants National Science Foundation (NSF) Office of Cyberinfrastructure (OCI) grant number 0950440, NSF grant number 0412000 (to S.A., S.S.L. and C.B.); Science Foundation Arizona (SFAZ) Competitive Advantage Award (CAA) grant number 0289-08 (to S.A., S.S.L. and C.B.).

Conflict of Interest: none declared.

REFERENCES

Aronson,A.R. (1996) *MetaMap: Mapping Text to the UMLS Metathesaurus*. NLM, NIH, DHHS.

Baral,C. (2003) *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, New York, NY.

Boobis,A. *et al*. (2009) Drug interactions. *Drug Metab. Rev.*, **41**, 486–527.

Garten,Y. and Altman,R.B. (2009) Pharmspresso: a text mining tool for extraction of pharmacogenomic concepts and relationships from full text. *BMC Bioinformatics*, **10** (Suppl. 2), S6.

Garten,Y. *et al*. (2010) Improving the prediction of pharmacogenes using text-derived gene-drug relationships. *PSB'2010*, **15**, 305–314.

Gebser,M. *et al*. (2009) Constraint answer set solving. In *Proceedings of the Twenty-fifth ICLP'09*, Pasadena, California, USA, pp. 235–249.

Gelfond,M. and Lifschitz,V. (1988) The stable model semantics for logic programs. In *Proceedings of the 5th ICLP*, Seattle, Washington, pp. 1070–1080.

Gelfond,M. and Lifschitz,V. (1991) Classical negation in logic programs and disjunctive databases. *New Gen. Comput.*, **9**, 365–387.

Hakenberg,J. *et al*. (2008) Inter-species normalization of gene mentions with GNAT. *Bioinformatics*, **24**, i126.

Leaman,R. and Gonzalez,G. (2008) BANNER: an executable survey of advances in biomedical named entity recognition. In *Proceedings of the PSB'13*, Hawaii, USA, pp. 652–663.

Masataka,T. (2008) Network analysis of adverse drug interactions. *Genome Inform.* **20**, 252–259.

Mavrovouniotis,M.L. *et al*. (1990) Computer-aided synthesis of biochemical pathways. *Biotechnol. Bioeng.*, **36**, 1119–1132.

Micromedex Healthcare Series [intranet database]. Version 5.1. Greenwood Village, Colo: Thomson Reuters (Healthcare) Inc.

Rindflesch,T.C. *et al*. (2000) EDGAR: Extraction of Drugs, Genes And Relations from the Biomedical Literature. In *Proceedings of the PSB 5*, Hawaii, USA, pp. 517–528.

Sleator,D. and Temperley,D. (1993) Parsing English with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, Tilburg, The Netherlands, pp. 277–292.

Tari,L. *et al*. (2010). Synthesis of pharmacokinetic pathways through knowledge acquisition and automated reasoning. In *Proceedings of the Pacific Symposium on Biocomputing 15*, Hawaii, USA, pp. 465-476.

Tran,N. *et al*. (2005) Knowledge-based framework for hypothesis formation in biochemical networks. *Bioinformatics*, **21** (Suppl. 2), ii213–ii219.

Tu,P.H. *et al*. (2008) Generalized text extraction from molecular biology text using parse tree database querying. Arizona State University, TR-08-004.

Wishart,D.S. *et al*. (2006) DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res.*, **34**, D668–D672.

Wong,C.M. *et al*. (2008) Clinically significant drug–drug interactions between oral anticancer agents and nonanti-cancer agents: Profiling and comparison of two drug compendia. *Ann. Pharmacother*., **42**, 1737–1748.