PGxCorpus: a Manually Annotated Corpus for Pharmacogenomics

Joël Legrand[1*], Romain Gogdemir[1], Cédric Bousquet[2],
Kevin Dalleau[1], Marie-Dominique Devignes[1], William Digan[3,4],
Chia-Ju Lee[5], Ndeye-Coumba Ndiaye[6], Nadine Petitpain[7],
Patrice Ringot[1], Malika Smaïl-Tabbone[1],
Yannick Toussaint[1], Adrien Coulet[1,8]

[1] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
[2] Sorbonne Université, INSERM, Université Paris 13, LIMICS, Paris, France
[3] Hôpital Européen Georges Pompidou, AP-HP, Université Paris Descartes, Université Sorbonne Paris Cité, Paris, France
[4] INSERM UMR 1138 Equipe 22, Université Paris Descartes, Université Sorbonne Paris Cité, Paris, France
[5] Department of Biomedical Informatics and Medical Education, University of Washington, Seattle, Washington
[6] INSERM U1256 - NGERE, Université de Lorraine, Nancy, France
[7] Centre Régional de Pharmacovigilance, CHRU of Nancy, Nancy, France
[8] Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, California

[*] **corresponding author:** joel.legrand@inria.fr

Abstract

Pharmacogenomics (PGx) studies how individual gene variations impact drug response phenotypes, which makes knowledge related to PGx a key component towards precision medicine. A significant part of the state-of-the-art knowledge in PGx is accumulated in scientific publications, where it is hardly usable to humans or software. Natural language processing techniques have been developed and are indeed employed for guiding experts curating this amount of knowledge. But, existing works are limited by the absence of high quality annotated corpora focusing on the domain. This absence restricts in particular the use of supervised machine learning approaches. This article introduces PGxCorpus, a manually annotated corpus, designed for the automatic extraction of PGx relationships from text. It comprises 945 sentences from 911 PubMed abstracts, annotated with PGx entities of interest (mainly genes variations, gene, drugs and phenotypes), and relationships between those. We present in this article the method used to annotate consistently texts, and a baseline experiment that illustrates how this resource may be leveraged to synthesize and summarize PGx knowledge.

Keywords: natural language processing, NLP, pharmacogenomics, corpus, manual annotation, entity recognition, relationship extraction

Background & Summary

Pharmacogenomics (or PGx) studies how individual gene variations impact drug response phenotypes [54]. This is of particular interest for the implementation of precision medicine, i.e. a medicine tailoring treatments (e.g. chosen drugs and dosages) to every patient, in order to reduce the risk of adverse effects and optimize benefits. Indeed, examples of PGx knowledge have already translated into clinical guidelines and practices [4,13], recommending the consideration of individual genotypes when prescribing some particular drugs. For example, patients with the allele *57:01 of the HLA gene are at high risk to present a hypersensitivity reaction if treated with abacavir, an anti-retroviral, thus should be genotyped for this gene before prescription [34].

Many scientific publications are reporting the impact of gene variants on drug responses, and Medline size (29 million articles) makes it hard for humans or machines to get a full understanding of the state of the art of this domain. NLP (Natural Language Processing) techniques have been consequently developed and employed to structure and synthesize PGx knowledge [9, 16]. Previous works investigated mainly rule-based approaches [6, 10, 42] and unsupervised learning [24, 38], because of the absence of annotated corpora. Supervised learning has also been experimented [5,28,37,43,55], but without a more appropriate corpus, most studies build train and test sets on the basis of PharmGKB, the reference database for PGx [52]. Because it is manually curated, PharmGKB provides a high quality referential for such task. Annotations provided by PharmGKB (i.e. 2 associated entities and the identifier of the PubMed article in support) result from the consideration by human curators of various knowledge sources: article text; tables and figures; and curator's own knowledge of the domain. Consequently PharmGKB annotations result from a high level process that can hardly be compared to an NLP-only approach. In particular, most NLP efforts are restricted to open-access texts only, without considering background knowledge. In this sense, evaluating an extraction system on PharmGKB enables to evaluate how it may guide the curation, but not how it can capture what is actually stated in texts.

In domains close to PGx, corpora have been annotated with biomedical entities, but only few of them include relationships (see Hahn *et al.* [16] for a panorama, plus [29,48]). The most interesting are related to pharmacovigilance or oncology, then focusing on drug–adverse response or drug–drug interactions. To our knowledge, no corpus has been constructed for PGx relationships, which requires a focus on drug response phenotypes and their relations with genomic variations. Developed for pharmacovigilance, **EU-ADR** [49] is a corpus of PubMed abstracts, annotated with *drugs*, *disorders* and targets (*proteins/genes* or *gene variants*). It is composed of three subcorpora, focusing on target-disease, target-drug and drug-disease relationships, each made of 100 abstracts. In the same vein, **ADE-EXT** (Adverse Drug Effect corpus, extended) [14] consists of 2,972 MEDLINE case reports, annotated with *drugs* and *conditions* (e.g. diseases, signs and symptoms) and their relationships. **SNPPhenA** [2] is a corpus of 360 PubMed abstracts, annotated with *single nucleotide polymorphisms* (SNPs), *phenotypes* and their relationships. Domains covered by EU-ADR, ADE-EXT or SNPPhena are related to PGx, but fit only partially with our purpose of PGx relation extraction. In particular EU-ADR and ADE-EXT encompass drug reactions without considering their genetic factor, and SNPPhena does not focus on drug response phenotypes and considers only SNPs whereas other genomic variations are also of importance in PGx. In addition, the size of EU-ADR and SNPPhena are relatively small (only a few hundreds of annotated sentences), which limits the use of supervised learning approaches that require large train sets such as TreeLSTM [47]. These elements motivated us to construct a new corpus, focused on PGx, and large enough to train deep neural network models.

Despite the existence of reference resources, in particular PharmGKB, and of alternative to supervised learning, such as weak supervision or active learning, we believe that high quality training data sets remain an asset for a domain and that the PGx community will benefit from PGxCorpus.

This manuscript presents: the construction of PGxCorpus, in Methods; the corpus itself, in Data Records; and a baseline experiment, in Technical Validation.

Methods

In this section, we detail the steps of the construction of our corpus named PGxCorpus, as presented in Figure 1. This process consists in two main steps: *(1)* the automatic pre-annotation of named entities and *(2)* the manual annotation that encompasses the correction of the pre-annotation and the addition of typed relationships between named entities.

We followed good practices proposed in [30], as well as practical examples provided by EU-ADR, ADE-EXT, SNPPhena and other corpora used in NLP shared tasks such as GENIA [22], SemEval DDI [17]. We particularly considered reports on the MERLOT corpus, which focuse on its annotation guidelines [3, 36] and inter-annotator agreement [12].

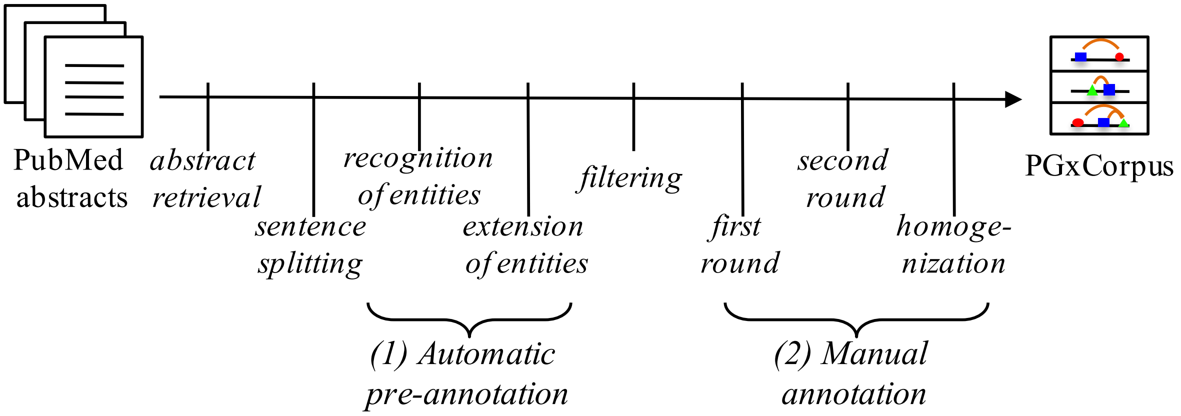

Figure 1. Overview of the construction of PGxCorpus.

Abstract retrieval and sentence splitting

The very first step consists in retrieving abstracts of publications related with PGx from PubMed [33]. This was performed with the tool EDirect [21] queried with:

```
Pharmacogenetics [MeSH Terms] OR
(  ( Therapeutics [MeSH Terms] OR
   Pharmaceutical Preparations[MeSH Terms] OR
   ChemicallyInduced Disorders[MeSH Terms] )                    (query 1)
   AND
   ( Genome Components[MeSH Terms] OR
   Genetic Variation[MeSH Terms] OR
   Genetic Testing[MeSH Terms] )
 )
```

This query aims at retrieving article abstracts concerned with PGx or with at least one treatment and one genetic factor. It has been built by browsing manually the hierarchy of the MeSH vocabulary, which annotates PubMed entries. The use of MeSH terms allows PubMed to retrieve articles using synonyms and descendant terms of those used in the query. The query is voluntarily made general to retrieve a large set of abstracts that may mention PGx relationships.

Every retrieved abstract is subsequently split into its constitutive sentences, using GeniaSS [44].

Automated pre-annotation

To facilitate the manual annotation of PGx relationships, we pre-annotate automatically sentences with various types of entities of interest for PGx. This pre-annotation is composed of two phases: First, PGx *key entities*, i.e. Gene, Mutation, Disease and Chemicals, are recognized and annotated with a state-of-the-art Named Entity Recognition (NER) tool. Second, these annotations are extended when they take part in the description of a PGx *composite entity*, such as a gene expression or a drug response phenotype.

Entity type	Tool	Evaluated on	Performance		
			P	R	F1
Chemicals	Dictionary-based [53]	n/a	n/a	n/a	53.82
Disease	DNorm [26]	NCBI Disease Corpus	82.8	81.9	80.9
Gene	GeneTUKit [20]	n/a	n/a	n/a	82.97
		GNAT-100	43.0	56.7	48.9
Mutation	tmVar [50]	MutationFinder Corpus	98.80	89.62	93.98

Table 1. Performances reported for PubTator. PubTator is the NER tool used during the pre-annotation step of PGxCorpus. P, R and F1 stand for Precision, Recall and F1-score, respectively. n/a denotes we were not able to find information to fill the cell.

Recognition of key PGx entities

Pre-annotation is initiated using PubTator [51], which recognizes the following biomedical entities from PubMed abstracts: chemicals, diseases, genes, mutations and species. PubTator integrates multiple challenge-winning text mining algorithms, listed in Table 1 along with their performances on various benchmark corpora. Disease recognition is performed with DNorm, which uses BANNER [25], a trainable system using Conditional Random Fields (CRF) and a rich feature set for disease recognition. For genes, GeneTUKit uses a combination of machine learning methods (including CRFs) and dictionary-based approaches. For mutations, tmVar also uses a CRF-based model with a set of features including dictionary, linguistic, character, semantic, case pattern and contextual features. PubTator was chosen for three reasons: it offers a wide coverage of the key entities for PGx; it provides an easy-to-use API to recover PubMed abstracts along with entity types and their boundaries; and it includes high performance NER tools.

Extension of the annotations with the PHARE ontology

The second phase of the pre-annotation consists in extending automatically key entity annotations, when possible, with the PHARE (PHArmacogenomic RElationships) ontology [10]. This ontology encompasses frequent terms that, associated in nominal structure with PGx *key entities*, form PGx *composite entities*. These terms were obtained by analyzing dependency graphs of nominal structures in which a key entity syntactically $modifies$ another term, and in turn were structured in the PHARE ontology. In the example provided in Figure 2, the drug name **acenocoumarol** syntactically modifies the term **sensitivity**. According to the PHARE ontology, the term *sensitivity*, when modified by a drug, forms a composite entity belonging to the *DrugSensitivity* class. Since this class is a subclass of the *Phenotype* class, **acenocoumarol sensitivity** may also be typed as a *Phenotype*. Following this principle, annotations of PGx key entities made by PubTator are extended, when possible, to PGx composite entities, then typed with classes of the PHARE ontology. For this matter, the dependency graph of each sentence is constructed with the Stanford Parser [11] and in each graph, the direct vicinity of key entities is explored in the search for terms defined in PHARE.

To homogenize the types of entities in PGxCorpus, we defined a reduced set of entities of interest, listed in Figure 3 and then defined mappings from PubTator entities and PHARE classes on one side to the types allowed in PGxCorpus on the other side. These mappings are reported in Table 2. Note that we decided to use a type Chemical, instead of Drug, first because we rely on PubTator that recognizes chemicals (without distinguishing between those and drugs), second because it allows to include broadly more candidate entities that may be involved in PGx relationships, such as drug metabolites or not yet approved drugs. Also, we decided on a type named Gene_or_protein, broader to Gene, because it is hard to disambiguate between gene and protein names in NLP, and commonly assumed that the task of gene name recognition is indeed a gene-or-protein name recognition [56].

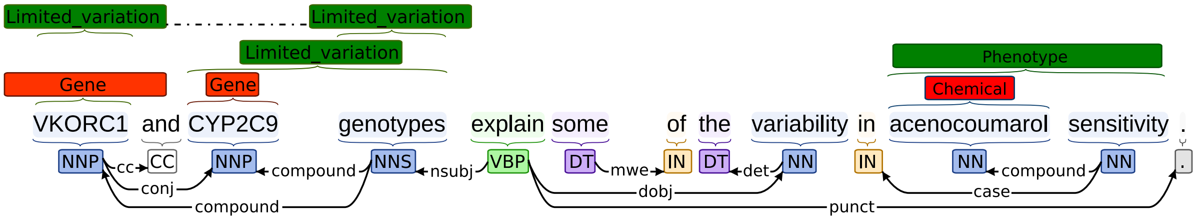

Figure 2. Example of sentence with PGx key and composite entities. The key entities, in red, correspond to entities retrieved by PubTator. Composite entities, in green, were obtained using the PHARE ontology. The syntactic dependency analysis is presented on the bottom of the figure and the entities on top.

Origin	Initial type	Type in PGxCorpus
PubTator	Chemical	Chemical
	Disease	Disease
	Gene	Gene_or_protein
	Mutation	Limited_variation
PHARE	Drug	Chemical
	DrugMetabolite	Chemical
	Gene	Gene_or_protein
	GenomicRegion	Genomic_factor
	GenomicVariation	Genomic_variation
	GeneProduct	Gene_or_protein
	Mutation	Limited_variation
	Phenotype	Phenotype

Table 2. Mapping between PubTator entities types, PHARE classes and PGxCorpus entity types.

Manual annotations

Before the manual annotation itself, malformed sentences (sentence tokenization errors) and sentences that did not contain at least one drug and one genetic factor, according to PubTator or PHARE are filtered out.

Out of the remaining sentences, we randomly select 1,897 of them to be manually annotated. The annotation process is realized by 11 annotators, out of which 5 are considered senior annotators. Annotators are either pharmacists (3), biologists (3) or bioinformaticians (5). Each sentence is annotated in **three phases**: First, it is annotated independently by two annotators (senior or not); Second, their annotations are, in turn, compared and revised by a third, senior annotator; Last, a homogenization phase ends the process.

During the first phase, annotators are provided with sentences and entity pre-annotations. At this stage, they correct pre-annotations, add potential relationships between them, and discard sentences which are ambiguous or not related with PGx domain. Sentences discarded by at least one annotator are not considered for the second phase. During both first and second phases, sentences are randomly assigned to annotators, but we ensure that senior annotators revise only sentences they did not annotate in the first phase.

In order to ensure the consistency of the manual annotations, annotators are provided with **detailed guidelines** [32]. Those describe the type of entities and relationships to annotate (reported here in Figures 3 and 4), relationship attributes (affirmed, negated, hypothetical), the main rules to follow, along with examples. Entity and relationship types are organized in simple hierarchies. Some of the relationship types are directly related to PGx (denoted with △ in Figure 4), whereas some have a broader scope (i.e. isEquivalentTo and treats). This document also provides an how-to-use guide for the annotation tool and answers frequently-asked questions. The first version of the guidelines has been written before the first phase of the annotation. Additional examples and clarifications were added regularly during the first phase of the annotation. Guidelines were subject to an important revision between the two first annotation phases, to clarify how to annotate ambiguous cases, which have been raised by annotators themselves or by the evaluation of agreement score between annotators (see Section Inter-annotator agreement).

The final phase of **homogenization** ends the corpus construction process to reduce heterogeneity remained in the annotations after the second phase. Two expert annotators review together sentences in two times: the first time is a complete pass on all annotated sentences to identify sources of heterogeneity. The second time consists in *(a)* listing sentences associated with each source of heterogeneity using programmatic scripts and keywords, *(b)* reaching a consensus for their annotation, and *(c)* accordingly modifying the annotations. Sources of heterogeneity identified at this stage include: the annotation of drug combinations, of dose-related phenotypes, of mutation-related cancer types (e.g. p53-positive breast cancer), of behavior-related phenotypes (e.g. drug abuse, drug dependence), of genomic factors (e.g. exons, promoters, regulatory regions), of treated conditions (e.g. transplantations or post-surgery treatments), uncommon type of relationships. Concerning the latter, annotations made with uncommon types (i.e. 'metabolizes' and 'transports') are turned into their upper-level type of annotations (i.e. 'influences'). For some heterogeneity sources, guidelines were specific, but sometimes disregarded by annotators; for others, they were caused by unexpected cases, absents from the guidelines.

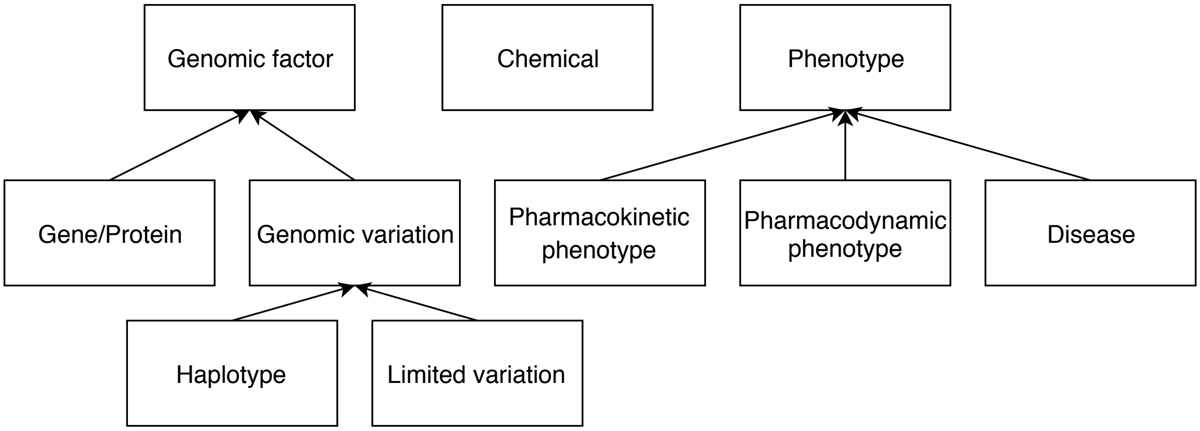

Figure 3. Types of entities annotated in PGxCorpus and their hierarchy.

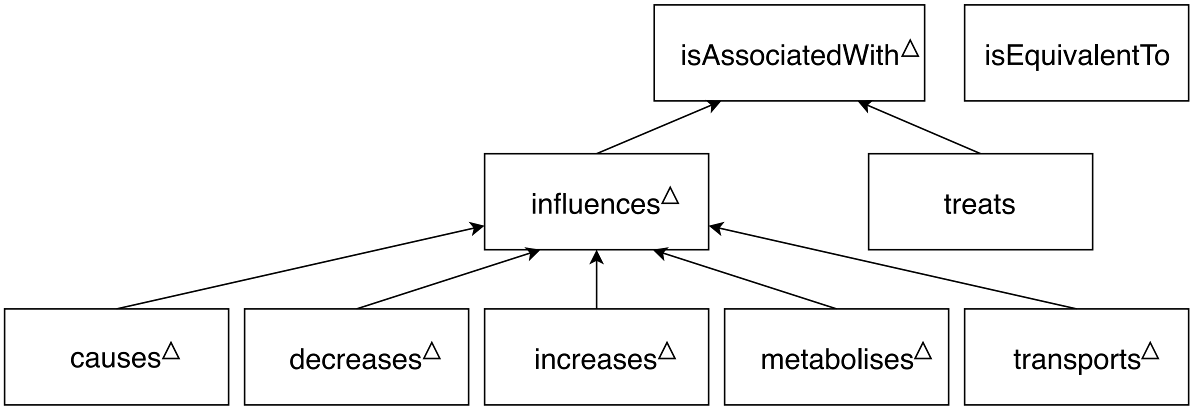

Figure 4. Types of relationships annotated in PGxCorpus and their hierarchy. Types directly related to PGx are marked with △, wheras isEquivalentTo and treats have a broader scope.

PubTator entity	Number recognized
Chemical	90,816
Disease	125,487
Gene	196,460
Mutation	25,417

Table 3. Type and number of entities recognized by PubTator in the pre-annotation.

Code availability

A Git repository of the whole project is accessible at `https://github.com/practikpharma/PGxCorpus/`. It includes the annotation guidelines, the corpus itself and the programmatic code of the baseline experiments presented in Technical Validation.

Data Records

Data availability

PGxCorpus is available in the BioNLP shared task file format [39] at three locations:

- **figshare**, an open access data repository, at the following address: `https://figshare.com/s/9d315cec6bb629d04210`
- A **BRAT server** [46], enabling a friendly online visualization of the annotations: `https://pgxcorpus.loria.fr/`
- A **Git** repository of the whole project that also includes the annotation guidelines and programmatic code of the baseline experiments presented in Technical Validation `https://github.com/practikpharma/PGxCorpus/`.

Statistics on the preparation of PGxCorpus

PubMed has been queried with our initial query (query 1) in July 2017, to retrieve 86,520 distinct abstracts, split out in 657,538 sentences. Statistics of pre-annotations obtained with PubTator and PHARE on these sentences are provided in Table 3 and 4, respectively. After filtering malformed sentences and sentences that do not contain at least one genomic factor and one drug, we obtain 176,704 sentences, out of which we randomly pick 1,897 sentences that are subsequently manually annotated. This number of sentences is chosen in regards of constraints of the distribution of the annotation task. These sentences come from 1,813 distinct abstracts.

PHARE entity	Discontiguous	All
Chemical	430	87,764
Disease	0	29,589
Gene_or_protein	4,690	10,1326
Genomic_variation	8,698	13,601
Phenotype	10,935	16,770

Table 4. Number of entities pre-annotated after extending PubTator annotation with the PHARE ontology. Because discontiguous entities are excluded from our baseline experiments (see Section Technical Validation), their number is specified.

The first phase of manual annotation, by 11 annotators, took roughly four months. The mean number of sentences annotated by an annotator is of 344.73 (standard deviation=126.33) sentences for this phase.

The second phase, by 5 senior annotators, took four other months. Each senior annotator revised 258.6 (sd=0.54) sentences. Annotations were made on a voluntary basis, which explains the relatively long length of this process.

Statistics on PGxCorpus

PGxCorpus encompasses 945 sentences, from 911 distinct PubMed abstracts, annotated with 6,761 PGx entities and 2,875 relationships between them. Detailed statistics on the type of entities and relationships annotated are provided in Table 5 and 6, respectively. Note that we distinguish two types of particular entities: nested and discontiguous ones. Nested entities are entities that encompass fully or partially at least one other entity in their offset. In Figure 2, the phenotype "acenocoumarol sensitivity" is an example of nested entity since it encompasses the "acenocoumarol" drug. Discontiguous entities are entities which offset is discontiguous, such as "VKORC1 genotypes" in Figure 2.

Note also that because of their rareness, annotations made with types 'metabolizes' or 'transports' were subsequently generalized as 'influences'. All the corpus abstracts were published between 1952 and 2017.

PGxCorpus entity	Nested	Discont.	Both	Total
Chemical	192	2	12	1,718
Genomic_factor	68	7	3	99
↳ Gene_or_protein	20	3	0	1,708
↳ Genomic_variation	37	3	0	54
↳ Limited_variation	537	98	47	919
↳ Haplotype	112	4	6	137
Phenotype	330	60	27	699
↳ Disease	143	14	18	635
↳ Pharmacodynamic_phenotype	390	60	25	632
↳ Pharmacokinetic_phenotype	109	14	6	160
Total	1,938	265	144	6,761

Table 5. Numbers of entities annotated in PGxCorpus, by type. Because discontiguous entities (Discont.) and nested entities are considered particularly in our baseline experiments, their numbers are reported. "Both" refers to entities both discontiguous and nested.

isAssociatedWith	733
↳ influences	937
↳causes	168
↳decreases	263
↳increases	243
↳ treats	238
isEquivalentTo	293
Total	2,875

Table 6. Numbers of relations annotated in PGxCorpus, by type. Because of their relatively rareness, annotations made with 'metabolizes' or 'transports' types have been subsequently turned in as 'influences' annotations in the corpus. All counts are disjoint.

Technical Validation

In this section we present an inter-annotator agreement analysis and the results of a baseline experiment of relation extraction using PGxCorpus as training data of a neural network model.

Inter-annotator agreement

Metrics

The annotation task considered for this corpus is particularly complex: it involves 10 entity types, 9 relation types and 3 relation attributes; in addition, entities may be discontiguous or nested. Given this complexity, metrics to control the variability of the annotations have been evaluated, in particular at the end of the first phase of the manual annotation, when each sentence has been annotated independently by two annotators. We evaluate an agreement score that evaluates how much annotators agreed with each others using the F1-score, following [15, 19]. In this case, the agreement or F1-score, is measured using one annotator as a reference and the other as a prediction. Note that inverting the reference and the prediction only inverts the precision and the recall but has no effect on the F1-score itself. We preferred the F1-score instead of other conventional measures, such as the kappa coefficient [7] because of the complexity of our annotation task. Kappa coefficient is designed to evaluate inter-annotator agreements while taking into account the probability that the agreement might be due to random guesses. It is adapted when annotators select a category, out of a set, to annotate already marked-up entities. Then, larger the set is, the less probable an agreement occurs by chance. In our case, the annotators need not only to select a category, but also to identify the boundaries of these potential entities. In this setting, the probability of a by-chance agreement within the kappa coefficient is low and unadapted. The F1-score is defined as the harmonic mean of the precision and recall, i.e. F1-score $= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$.

Entity agreement

Agreement on the entity annotations is determined in four ways, in regards with two parameters: *(a)* using *exact* or *partial* match; *(b)* considering the entity hierarchy or not.

(a) An *exact match* occurs when two annotators agree on both the entity type and their boundaries. A *partial* match is more flexible since it occurs when two annotators agree on the entity type, but annotation boundaries only overlap. Note that an annotation from the first annotator may overlap with multiple annotations from the second annotator, and vice versa. Considering every overlapping entities as a match would artificially increase the recall and the precision because only one can indeed reflect an agreement between the two annotators. We ensure in this case that an entity from the first annotator is matched with at most one entity from the second annotator using the Hopcroft-Karp algorithm [18]. In this case, the problem is seen as a maximum matching problem in a bipartite graph, where each set of annotations, one for each annotator, represents a sub-graph. The edges between the two sets represent possible overlaps between one annotation from the first annotator and one from the second.

(b) We also consider a more flexible setting where the agreement takes into account the upper hierarchies of entities and relationships, as defined in Figures 3 and 4. For instance, if a first annotator annotates an entity as *Pharmacokinetic phenotype (PK)* and a second as *Pharmacodynamic phenotype (PD)*, we consider they agreed to some extent, since both are subtype of *Phenotype*. In this setting, it can be considered that an entity (or relationship) is indeed annotated with several types: the one specified by an annotator and its parents in the hierarchy. In practice, if we consider the first annotator as the reference and the second as the prediction, we can distinguish three cases: (1) the prediction is more specific than the reference. In this case, common annotations shared by reference and prediction are counted as *true positives*, while annotations of the prediction that are too specific are *false positives*. For instance if the reference is *Phenotype* and the prediction is *PD*; we count one *false positive* in the evaluation of *PD* predictions, but the additional *Phenotype* annotation, inferred from the hierarchy, enables to count one *true positive* for *Phenotype* predictions. (2) The prediction is less specific than the reference. In this case, common annotations shared by reference and prediction are counted as *true positives*, while classes from the reference that are missed by the prediction are *false negative*. For instance if the reference is *PD* and the prediction is *Phenotype*, we count one *true positive* for *Phenotype* prediction, but one *false negative* in the prediction of *PD*. (3) The reference and the prediction do not have a direct hierarchy relationships, but a common ancestor (like *PD* and *PK*). In this case classes that are shared by the prediction and reference (i.e. the common ancestors) are counted as *true positive*, but too specific predictions as *false positives* and missed predictions as *false negatives*. For instance if

the reference is *PD* and the prediction is *PK*, we count one *true positive* for the prediction of *Phenotype* (i.e. the only common ancestor), one *false positive* for the prediction of *PK* and one *false negative* for the prediction of *PD*.

Table 7 presents the inter-annotator entity agreement scores, obtained for the first phase of the manual annotation, depending on settings *(a)* and *(b)*. We observe that for relatively simple entities such as chemicals, genes, haplotypes or diseases the F1-score, even on the strictest constraints (exact match, no hierarchy), overpasses 70. We observe also that for more complex entities such as phenotypes, annotators tend to agree on the presence of an entity, but not on its offset. This motivates us to update the annotation guidelines between the two annotation phases, to particularly clarify on how to decide on entity offsets. When considering the hierarchy, the performances for the leaves of the hierarchy should not be affected. However, a slight drop is observed due to the use of the Hopcroft-Karp algorithm. Indeed, when using the hierarchy more potential matches can be observed between prediction and reference annotations generating more edges in the associated bipartite graph. The Hopcrof-Karp algorithm then removes some of the correct matches between leaves, causing a slight drop in the recall.

Entity matching: (exact or partial)	exact	exact	partial	partial
Considering hierarchy: (yes or no)	no	yes	no	yes
Chemical	76.8	76.8	82.1	82.1
Genomic_factor	38.6	72.6	38.8	85.7
↳ Gene_or_protein	85.3	85.3	90.0	89.4
↳ Genomic_variation	32.9	49.3	53.0	76.8
↳ Limited_variation	50.8	50.8	69.0	66.2
↳ Haplotype	76.2	76.2	77.2	76.1
Phenotype	30.5	51.0	53.9	72.6
↳ Disease	71.3	71.0	80.9	79.1
↳ Pharmacokinetic_phenotype	48.2	48.2	57.0	57.0
↳ Pharmacodynamic_phenotype	31.7	31.7	47.0	47.0
Macro average	57.4	63.8	68.7	76.1

Table 7. Inter-annotator agreement (F1-score) for entity annotations. Four different settings, enabling more or less flexibility are presented. The agreement score is computed after the first phase of manual annotation.

Relation agreement

Regarding the inter-annotator agreement on relation annotations, we consider the same two settings, plus an additional one: *(a)* using *exact* or *partial* match, which applies in this case to the two entities involved in the relation; *(b)* the consideration of the hierarchy, which applies in this case to both the hierarchy of entities and relations (see Figure 3 and 4); *(c)* the direction of the relation is considered or not. Resulting agreements are presented in Table 8.

Although the agreement on the relations is low, note that a relation can be considered correct only if an initial agreement on the two entities in relation has been reached.

Baseline experiments

In this section, we report on baseline experiments with PGxCorpus, which evaluates quantitatively its usefulness for extracting PGx entities and relations from text. The task evaluated here is composed of a first step of named entity recognition (NER) and a second one of relation extraction (RE). The NER is achieved with a variant of a Convolutional Neural Network (CNN) model, whereas the RE is processed with a multichannel CNN (MCCNN). Source code of the experiments is available at `https://github.com/practikpharma/PGxCorpus/`.

Entity matching: **Considering hierarchies:** **Considering direction:**	exact none yes	exact both yes	partial none yes	partial both yes	partial both no
isAssociatedWith	12.6	14.3	13.2	33.3	33.3
↳ influences	12.8	12.8	17.7	29.3	29.8
↳ causes	35.8	35.2	37.6	37.2	39.6
↳ decreases	25.8	26.8	33.6	36.7	36.7
↳ increases	14.5	15.6	27.4	30.2	30.2
↳ metabolizes	59.0	59.0	61.5	61.5	61.5
↳ transports	83.1	83.1	83.1	83.1	83.1
↳ treats	33.2	34.7	36.3	37.3	37.3
isEquivalentTo	39.6	40.2	40.7	41.3	62.5
Macro average	47.3	47.1	50.3	53.8	57.0

Table 8. Inter-annotator agreement (F1-score) for the annotation of relations. Five different settings are presented.

In a related work [35], we used a preliminary, partial and naive set of annotations, for testing the feasibility of extracting relations and incorporating them in a knowledge network. This included only 307 sentences (out of 945), annotated with a simplified schema of only 4 entity types and 2 relation types. The associated model for RE was simplistic, since it aimed at proofing feasibility only. The baseline experiment reported here considers all sentences of PGxCorpus and has been done with more advanced annotation schema and models.

Sentence representation with word embeddings

Both our models for NER and RE are fed with *word embeddings* (*i.e.*, continuous vectors) of dimension d_w, along with extra *entity embeddings* of size d_e. RE is fed with an additional *nested entity embeddings* of size d_n.

Regarding word embeddings, given a sentence of N words, $w_1, w_2, \ldots, w_N$, each word $w_i \in \mathcal{W}$ is embedded in a d_w-dimensional vector space by applying a lookup-table operation: $LT_W(w_i) = W_{w_i}$, where the matrix $W \in R^{d_w \times |\mathcal{W}|}$ represents the parameters to be trained in this lookup-table layer. The dictionary $\mathcal{W}$ is composed of all the words of the corpus. Each column $W_{w_i} \in R^{d_w}$ corresponds to the embedding vector of the w_i word in our dictionary $\mathcal{W}$.

Beside word embeddings, two additional embeddings, named entity embeddings, are used to feed our models. (1) One entity embeddings enables to represent what type of entity a word composes. (2) One represents if the word starts, continues or ends the description of an entity. Both use a standard encoding of tags with Begin Intermediate Other End and Single (BIOES)-prefixes [41]. These two first entity embeddings are constructed slightly differently for NER and RE, since in the first, it encompasses tags for entities pre-annotated with PubTator and tags for entities annotated with PGxCorpus types, whereas in the latter, it considers tags for entity types of the corpus, plus special tags that marks pairs of entities between which a relationship may stand.

For the RE model only, a *nested entity embedding* of size d_n is added to word and entity embeddings to represent entity types that may be included in nested entities involved in relations. For each word a *nested entity embedding* is added for each entity type. Given an entity type, this embedding can take one of two values: (a) *absent* if the word is not part of one of the two entities potentially related, or if it is part of one, but no entity of the given type is included in the entity of interest; (b) *present* if the word is part of one of the 2 entities and this one includes another entity of the given type.

Finally, word, entity and nested entity embeddings are concatenated to form the input corresponding to a given word. Let's denote x_i the concatenated input corresponding to the i^{th} word.

Named entity recognition

The core of the CNN model used for NER is described in [8]. We adapted it, along with experiment settings, to fit with the particularity of PGxCorpus that is to encompass about one third of *discontiguous* or *nested entities* $(2,059$ discontiguous or nested / 6,761 entities, see Table 5).

Recognizing discontiguous entities is a complex and open problem in NLP and this baseline experiment does not aim at tackling it. For this reason, we discarded in the sentences, annotations of discontiguous entities from both our train and test sets (265/ 6,761 entities). Nested entities are considered in our experiment by applying the NER model recursively, as many times as there are nesting levels. Entities discovered during one iteration of the model are considered as input of the next iteration. Given the example of Figure 2, a first iteration will recognize the three entities "VKORC1", "CYP2C9" and "acenocoumarol". Then, the second iteration will consider them as an input to recognize "CYP2C9 genotypes" and "acenocoumarol sensitivity". "VKORC1 genotypes" is discontiguous and consequently discarded from the experiment.

Formally, given an input sequence $x_1, \dots, x_N$, a classical sliding window approach is followed by applying a two-layer neural network (NN) on each possible window of size k. We denote $\mathcal{P}$ the set of BIOES-prefixed tags. Given the i^{th} window, the NN computes a vector of scores $s_i = [s^1, \dots, s^{|\mathcal{P}|}]$, where s^t is the score of the BIOES-prefixed tag $t \in \mathcal{P}$, associated with the input x_i. Scores of the window i are given by the following formula:

$$s_i = W_1\ h(\ W_2\ [x_{i-(\frac{k-1}{2})}, \dots, x_i, \dots, x_{i-(\frac{k+1}{2})}]\ ),$$

where the matrices $W_1 \in R^{d_h \times k|\mathcal{W}|}$ and $W_2 \in R^{|\mathcal{P}| \times d_h}$ are the trained parameters of the NN, and h is a pointwise non-linear function such as the hyperbolic tangent, d_h is the number of hidden units and k the size of the window. Inputs with indices exceeding the input boundaries, i.e. when $i - (\frac{k-1}{2}) < 1$ or $i - (\frac{k+1}{2}) > N$, are mapped to a special padding vector, which is also learned.

Scores of each window are finally given to a lattice module that allows to aggregate the BIOES-prefixed tags from our tagger module in a coherent manner, to recover the predicted labels. For more details about this layer, please see [8].

Relation extraction

The model used for RE is a multichannel CNN (MCCNN) described in [40], where it has been successfully applied to the task of extraction of drug-drug and protein-protein interactions. It takes an input sentence and two recognized entities, computes a fixed size representation by composing input word embeddings. This representation is given to a scorer, which computes a score for each possible type of relationships. Sentences with more than two entities are considered by the model iteratively for each possible pair of entities for which a relation may stand, in both directions since relations may be oriented.

The MCCNN applies a CNN of variable kernel size to each input channels of word embeddings. In other words, it considers different embedding channels i.e. different versions of the word embeddings associated with each word, allowing to capture different aspects of input words. Formally, given an input sequence of word representations (i.e. concatenation of word and entity embedding) $x_1, \dots, x_N$, applying a kernel to the i^{th} window of size k is done using the following formula:

$$C_i = h(\sum_{j=1}^{N-k+1} W[x_i, \dots, x_{i+k-1}]^j + b)$$

where $[.]^j$ denotes the concatenation of inputs from channel j, $W \in \mathcal{R}^{(d_w+d_e) \times d_h}$ and $b \in \mathcal{R}^{d_h}$ are the parameters, d_h is the size of the hidden layer, h is a pointwise non-linear function such as the hyperbolic tangent and $N-k+1$ is the number of input channels. For each kernel, a fixed size representation $r^* \in \mathcal{R}^{d_h}$ is then obtained by applying a max-pooling over time (here, the "time" means the position in the sentence):

$$r^* = \max\left[C_1, \dots, C_{N-k+1}\right] .$$

We denote K the number of kernels with different sizes. A sentence representation $r \in \mathcal{R}^{d_s}$ (with $d_s = K * d_h$) is finally obtained by concatenating the output corresponding to the K kernels $r = [r_1^*, \ldots, r_K^*]$.

The sentence representation is finally passed to a single layer NN, which outputs a score for each possible relation type:

$$s(r) = W^{(s)} r + b^{(s)} \ ,$$

where $W^{(s)} \in \mathcal{R}^{d_s \times |S|}$ and $b^{(s)} \in \mathcal{R}^{|S|}$ are the trained parameters of the scorer, $|S|$ is the number of possible relation types. The scores are interpreted as probabilities using a softmax layer [1].

Experimental settings

Word embeddings were pre-trained using the method described in [27] on about 3.4 million PubMed abstracts, corresponding to articles published between Jan. 1, 2014 and Dec. 31, 2016. Our models were trained by minimizing the negative log-likelihood over the training data. All parameters –embeddings, weights W and biases b– were iteratively updated via backpropagation. We used a *hard tanh* function as activation function f. Hyper-parameters were tuned using a 10-fold cross-validation by selecting the values leading to the best averaged performance, and fixed for the rest of the experiment.

For NER, the CNN was fed with word embeddings and two types of entity embeddings (one with PubTator tags, used only for the first iteration of the model and one with PGxCorpus tags used in next iterations) of size $d_w = 100$ and $d_e = 20 \times 2$ (20 for each type of tags), respectively. The size of the hidden layer was fixed to $d_h = 200$, the kernel size to $k = 5$ and the learning rate to 0.01.

For RE, the MCCNN was fed with word embeddings and two types of entity embeddings (one with PGxCorpus entity tags; one to identify pairs of entities between which a relation may stand) of size $d_w = 200$ and $d_e = 20 \times 2$, respectively. The size of the nested entity embeddings was set to $d_n = 5 \times |\mathcal{E}|$, where $\mathcal{E}$ is the entity type dictionary.

We used two kernels of size 3 and 5. Following [23], both channels were initialized with pre-trained word embeddings, but gradients were backpropagated only through one of the channels. The size of the hidden layer was fixed to $d_h = 200$ and the learning rate to 0.01.

For both NER and RE, we applied a dropout regularization after the embedding layers [45] with a dropout probability fixed to 0.5. Both models were evaluated using a 10-fold cross validation. Each result of this evaluation is an average of 100 experiments: 10 experiments for each of the 10 folds starting with different random initializations. Random initialization concerns entity embeddings, weights and biases, but not word embeddings not randomly initialized, but pre-trained.

Baseline performances

The objective of these experiments was not to reach the best performances but rather to propose a baseline for future comparisons, as well as to empirically demonstrate the usefulness of PGxCorpus for extracting PGx entities and relations from text.

Named entity recognition

Performances for the named entity recognition experiments, evaluated with a 10-fold cross validation, are reported in Table 9. A main limitation of the NER model is that discontiguous entities were not considered. This may hurt the performance even for contiguous entities since discontiguous entities were considered as negative, even though they might be very similar (from the model point of view) to contiguous entities.

From results reported in Table 9, other observations can be made. First, the best performances were obtained for *Chemical*, *Gene_or_protein* and *Disease* types, for which (1) the number of training samples is high, (2) PubTator annotations are available and (3) the ratio between normal entities and nested and/or discontiguous entities is low (see Table 5). Note that the definition for the *Limited_variation* entity used in our corpus is broader than the *Mutations* recognized by PubTator. PubTator recognizes precises descriptions of variations such as “VKORC1:C>A”, but not general ones such as “a VKORC1 polymorphism”, which we consider. This explains why the performances obtained for *Limited_variation* were lower than those obtained with PubTator (see Table 1). Even though the number of training samples for *Pharmacokinetic_phenotype* and *Haplotype* is low, we obtained reasonable performances. This may be due to a rather homogeneous phrasing and syntax in the mention of these entities. When not considering the hierarchy, *Genomic_variation* and *Genomic_factor* types for which few training samples are available and a high heterogeneity is observed led to poor performances. Lastly we note that, as expected, the standard deviation for classes with only few examples annotated was high or very high (above 19 for *Haplotype* and *Pharmacokinetic_phenotype*). The random distribution of these "rare" examples between train and test sets, in the 10-fold cross validation, had a strong impact on performances, and explains these large standard deviations. Concerning concepts that are leaves of the hierarchy, we observed a slight drop in performances when considering the hierarchy. This is due to the use of the Hopcroft-Karp algorithm as mentioned in the Subsection Entity agreement.

Entity matching: (exact or partial)	exact	exact	partial	partial
Considering hierarchy: (yes or no)	no	yes	no	yes
Chemical	76.07	76.07	82.67	82.67 (7.24)
Genomic_factor	22.86	71.41	27.68	83.19 (5.90)
↳ Gene_or_protein	85.72	85.72	90.58	90.05 (3.89)
↳ Genomic_variation	2.67	49.13	3.83	71.18 (9.55)
↳ Limited_variation	47.08	47.02	72.71	71.57 (9.50)
↳ Haplotype	66.97	66.97	72.47	72.47 (19.34)
Phenotype	31.76	50.80	48.48	69.57 (5.40)
↳ Disease	66.90	66.88	75.68	72.59 (7.30)
↳ Pharmacokinetic_phenotype	29.30	29.30	36.47	36.27 (19.40)
↳ Pharmacodynamic_phenotype	38.54	38.50	58.84	58.18 (10.11)
Macro average	49.15	59.11	59.76	71.93 (5.64)

Table 9. Performances of the task of named entity recognition in terms of F1-score (and its standard deviation in brackets, for the last setting). Balance between precision and recall, as well as details on standard deviations are provided in Supplementary Table S1.

Relation extraction

Performances for the relation extraction (RE) experiments, evaluated with a 10-fold cross validation, are reported in Table 10. The RE model faced several limitations: (1) for a given sentence along with identified entities, the relation predictions were independent. This is obviously too simplistic and the prediction should be made globally. (2) We considered relationships annotated as *negated* or *hypothetical* by annotators just as regular relationships.

Several observations can be made about the RE results in Table 10. First, the fact that the model had to deal with multiple, complex and associated classes made the classification problem difficult and the performances relatively modest. The experiment in which we considered the hierarchy showed that, even if it was difficult to identify a specific type of relation, is was easier for the model to determine whether there was a relation between two entities or not. In other words, many mis-classifications were in fact predictions for types that belong to the same branch of the hierarchy. Like for the NER, types of relation with less examples tended to be associated with poorer performances and higher standard deviations (except for the *isEquivalentTo* relationship, which is very homogeneous). To build upon these observations, and particularly to avoid the impact of isEquivalentTo type that is not specific to PGx, we evaluated how PGxCorpus can be used to train a model for relations specific to PGx (denoted with △ in Table 10), but without consideration of their sub-types. Results of this experiment is provided on the last line of Table 10

Several enhancements could be introduced to improve this baseline model. First, in our implementation, the hierarchy was not considered during the training phase. Accordingly, learning to predict a leaf penalized all the other categories, even those that were considered correct at test time. This explains why the "PGx Relations only" experiment led to better performances than individual classifications with or without hierarchy. On the other hand, considering the hierarchy at training would increase the number of examples for the higher categories of the hierarchy, potentially harming performances for the leaves. A model enabling multiclass labeling and a weighting dependent on the size of the classes should balance this bias.

Considering hierarchies: (yes or no)	no	yes
isAssociatedWith$^{\triangle}$	30.89	51.71 (4.02)
↳ influences$^{\triangle}$	36.55	46.45 (5.17)
↳ causes$^{\triangle}$	41.91	41.91 (13.35)
↳ decreases$^{\triangle}$	29.47	29.47 (9.85)
↳ increases$^{\triangle}$	17.94	17.94 (15.20)
↳ treats	39.97	39.97 (12.60)
isEquivalentTo	79.76	79.76 (7.69)
Macro average	45.67	49.56 (4.51)
PGx relations only(△), no hierarchy	**54.04** (3.31)	

Table 10. Performances of the task of relation extraction in terms of F1-score (and standard deviation). The last line provides results of an experiment for which only one category is considered, merging all the type specific to PGx (marked with △). For leaves, performances are unchanged when considering the hierarchy. Balance between precision and recall, as well as details on standard deviations are provided in Supplementary Table S2.

Building upon PGxCorpus

We proposed an annotated corpus, named PGxCorpus, and an experimental validation of its usefulness for the tasks of NER and RE in pharmacogenomics.

Unlike existing corpora, PGxCorpus encompasses the three main entities involved in PGx relationships (drugs, genomic factors and phenotypes) and provides a fine-grained hierarchical classification for both PGx entities and relationships. By making this corpus freely available, our objective is to enable the training of supervised PGx relation extraction systems and to facilitate the comparison of their performances. Furthermore, the baseline experiment illustrates that PGxCorpus enables studying many challenges inherent with biomedical entities and relationships: discontiguous entities, nested entites, multilabeled relationships, heterogenous distributions, *etc.*). In particular, PGxCorpus offers both a training resource for supervised approaches and a reference to evaluate and compare to in future efforts. Out of pharmacogenomics, such a corpus may more generally serve transfer learning approaches, as illustrated by [31]. For these reasons, we think that tasks of PGx NER and RE, supported by PGxCorpus, are well suited for Bio-NLP Challenges and shared tasks. Consequently, our expectation is that the release of PGxCorpus will stimulate Bio-NLP research.

Usage Notes

PGxCorpus is made available under the Creative Commons Attribution-Non-Commercial 4.0 International Public License. The programmatic code of our baseline experiments is available at `https://github.com/practikpharma/PGxCorpus/tree/master/baseline_experiment`.

Acknowledgements

The authors acknowledge their funding institutions: the French National Research Agency (ANR) supports the PractiKPharma project (ANR-15-CE23-0028) and the University of Lorraine program "Lorraine Université d'Excellence" (15-IDEX-0004); Inria supports the *Snowball* Inria Associate Team.

Author contributions

JL conducted the annotation campaign, designed and conducted baseline experiments, and wrote the manuscript.
RG conducted the annotation campaign.
AC, CB, CJL, JL, KD, MDD, MST, NCN, NP, RG, WD annotated the corpus and reviewed the manuscript.
PR advised on technical aspects of the project and set up the annotation servers.
YT advised on the design of the project and in the writing of the manuscript.
AC designed the study, supervised the annotation campaign and wrote the manuscript.
All authors read and approved the final manuscript.

Competing financial interests

The authors declare no competing financial interests.

References

[1] Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition.* Information science and statistics. Springer, 2007.

[2] Behrouz Bokharaeian, Alberto Díaz Esteban, Nasrin Taghizadeh, Hamidreza Chitsaz, and Ramyar Chavoshinejad. Snpphena: a corpus for extracting ranked associations of single-nucleotide polymorphisms and phenotypes from literature. *J. Biomedical Semantics*, 8(1):14:1–14:13, 2017.

[3] Leonardo Campillos, Louise Deléger, Cyril Grouin, Thierry Hamon, Anne-Laure Ligozat, and Aurélie Névéol. A french clinical corpus with comprehensive semantic annotations: development of the medical entity and relation limsi annotated text corpus (merlot). *Language Resources and Evaluation*, pages 1–31, 2017.

[4] K. E. Caudle, T. E. Klein, J. M. Hoffman, D. J. Muller, M. Whirl-Carrillo, L. Gong, E. M. McDonagh, K. Sangkuhl, C. F. Thorn, M. Schwab, J. A. Agundez, R. R. Freimuth, V. Huser, M. T. Lee, O. F. Iwuchukwu, K. R. Crews, S. A. Scott, M. Wadelius, J. J. Swen, R. F. Tyndale, C. M. Stein, D. Roden, M. V. Relling, M. S. Williams, and S. G. Johnson. Incorporation of pharmacogenomics into routine clinical practice: the Clinical Pharmacogenetics Implementation Consortium (CPIC) guideline development process. *Curr. Drug Metab.*, 15(2):209–217, Feb 2014.

[5] Jeffrey T Chang and Russ B Altman. Extracting and characterizing gene–drug relationships from the literature. *Pharmacogenetics and Genomics*, 14(9):577–586, 2004.

[6] Luoxin Chen, Carol Friedman, and Joseph Finkelstein. Automated metabolic phenotyping of cytochrome polymorphisms using pubmed abstract mining. In *AMIA Annual Symposium Proceedings*, volume 2017, page 535. American Medical Informatics Association, 2017.

[7] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[9] Adrien Coulet, K. Bretonnel Cohen, and Russ B. Altman. The state of the art in text mining and natural language processing for pharmacogenomics. *Journal of Biomedical Informatics*, 45(5):825–826, 2012.

[10] Adrien Coulet, Nigam H. Shah, Yael Garten, Mark Musen, and Russ B. Altman. Using text to build semantic networks for pharmacogenomics. *Journal of Biomedical Informatics*, 43(6):1009 – 1019, 2010.

[11] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006.*, pages 449–454, 2006.

[12] Louise Deléger, Anne-Laure Ligozat, Cyril Grouin, Pierre Zweigenbaum, and Aurélie Névéol. Annotation of specialized corpora using a comprehensive entity and relation scheme. In *LREC*, pages 1267–1274, 2014.

[13] U.S. Food and Drug Administration. Table of pharmacogenomic biomarkers in drug labeling, 2018. *Online*. `http://www.fda.gov/Drugs/ScienceResearch/ResearchAreas/Pharmacogenetics/ucm083378.htm` Accessed: 2018-07-04.

[14] Harsha Gurulingappa, Abdul Mateen-Rajpu, and Luca Toldo. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 3(1):15, 2012.

[15] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics*, 45(5):885–892, 2012.

[16] Udo Hahn, Kevin Bretonnel Cohen, Yael Garten, and Nigam H. Shah. Mining the pharmacogenomics literature - a survey of the state of the art. *Briefings in Bioinformatics*, 13(4):460–494, 2012.

[17] María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions. *Journal of Biomedical Informatics*, 46(5):914–920, 2013.

[18] John E Hopcroft and Richard M Karp. An nˆ5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.

[19] George Hripcsak and Adam S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *JAMIA*, 12(3):296–298, 2005.

[20] Minlie Huang, Jingchen Liu, and Xiaoyan Zhu. Genetukit: a software for document-level gene normalization. *Bioinformatics*, 27(7):1032–1033, 2011.

[21] Jonathan Kans. Entrez direct: E-utilities on the unix command line. In *Entrez Programming Utilities Help*. National Center for Biotechnology Information, Bethesda (MD), USA, 3 edition, 7 2013. Available online at: `https://www.ncbi.nlm.nih.gov/books/NBK179288/`.

[22] Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10, 2008.

[23] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.

[24] Maria Kissa, George Tsatsaronis, and Michael Schroeder. Prediction of drug gene associations via ontological profile similarity with application to drug repositioning. *Methods*, 74:71 – 82, 2015. Text mining of biomedical literature.

[25] Robert Leaman and Graciela Gonzalez. BANNER: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008, Proceedings of the Pacific Symposium, Kohala Coast, Hawaii, USA, 4-8 January 2008*, pages 652–663, 2008.

[26] Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917, 2013.

[27] Rémi Lebret and Ronan Collobert. Word embeddings through hellinger PCA. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 482–490, 2014.

[28] Kyubum Lee, Byounggun Kim, Yonghwa Choi, Sunkyu Kim, Wonho Shin, Sunwon Lee, Sungjoon Park, Seongsoon Kim, Aik Choon Tan, and Jaewoo Kang. Deep learning of mutation-gene-drug relations from the literature. *BMC Bioinformatics*, 19(1):21, Jan 2018.

[29] Kyubum Lee, Sunwon Lee, Sungjoon Park, Sunkyu Kim, Suhkyung Kim, Kwanghun Choi, Aik Choon Tan, and Jaewoo Kang. Bronco: Biomedical entity relation oncology corpus for extracting gene-variant-disease-drug relations. *Database*, 2016:baw043, 2016.

[30] Geoffrey Leech. Adding linguistic annotation. In Martin Wynne, editor, *Developing linguistic corpora: A guide to good practice*, volume 92, pages 17–29. Oxbow Books Oxford, Oxford, 2005.

[31] Joël Legrand, Yannick Toussaint, Chedy Raïssi, and Adrien Coulet. Syntax-based Transfer Learning for the Task of Biomedical Relation Extraction. In *LOUHI 2018 - The Ninth International Workshop on Health Text Mining and Information Analysis*, Proceedings of LOUHI 2018: The Ninth International Workshop on Health Text Mining and Information Analysis, Brussels, Belgium, October 2018.

[32] Joël Legrand, Romain Gogdemir, Nadine Petitpain, and Adrien Coulet. PGxCorpus – Annotation guidelines, 2017. *Online*. `https://github.com/practikpharma/pgxcorpus-guidelines/blob/master/annotation_guidelines.pdf` Accessed: 2018-09-10.

[33] Zhiyong Lu. Pubmed and beyond: a survey of web tools for searching biomedical literature. *Database*, 2011:baq036, 2011.

[34] M. A. Martin, J. M. Hoffman, R. R. Freimuth, T. E. Klein, B. J. Dong, M. Pirmohamed, J. K. Hicks, M. R. Wilkinson, D. W. Haas, and D. L. Kroetz. Clinical Pharmacogenetics Implementation Consortium Guidelines for HLA-B Genotype and Abacavir Dosing: 2014 update. *Clin. Pharmacol. Ther.*, 95(5):499–500, May 2014.

[35] Pierre Monnin, Joël Legrand, Graziella Husson, Patrice Ringot, Andon Tchechmedjiev, Clément Jonquet, Amedeo Napoli, and Adrien Coulet. Pgxo and pgxlod: a reconciliation of pharmacogenomic knowledge of various provenances, enabling further comparison. *bioRxiv preprint*, 2018.

[36] Working Group of the Cabernet Project. Annotation scheme for the merlot french clinical corpus., 2016. *Online*. `https://cabernet.limsi.fr/annotation_guide_for_the_merlot_french_clinical_corpus-Sept2016.pdf` Accessed: 2018-07-04.

[37] S. Pakhomov, B.T. McInnes, J. Lamba, Y. Liu, G.B. Melton, Y. Ghodke, N. Bhise, V. Lamba, and A.K. Birnbaum. Using pharmgkb to train text mining approaches for identifying potential gene targets for pharmacogenomic studies. *Journal of Biomedical Informatics*, 45(5):862 – 869, 2012. Text Mining and Natural Language Processing in Pharmacogenomics.

[38] Bethany Percha and Russ B. Altman. Learning the structure of biomedical relationships from unstructured text. *PLoS Computational Biology*, 11(7), 2015.

[39] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Dan Sullivan, Chunhong Mao, Chunxia Wang, Bruno Sobral, Jun'ichi Tsujii, and Sophia Ananiadou. Overview of the id, epi and rel tasks of bionlp shared task 2011. In *BMC bioinformatics*, volume 13, page S2. BioMed Central, 2012.

[40] Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. Multichannel convolutional neural network for biological relation extraction. *BioMed research international*, 2016, 2016.

[41] Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. *CoRR*, cmp-lg/9505040, 1995.

[42] Bastien Rance, Emily Doughty, Dina Demner-Fushman, Maricel G. Kann, and Olivier Bodenreider. A mutation-centric approach to identifying pharmacogenomic relations in text. *Journal of Biomedical Informatics*, 45(5):835–841, 2012.

[43] Fabio Rinaldi, Gerold Schneider, and Simon Clematide. Relation mining experiments in the pharmacogenomics domain. *Journal of Biomedical Informatics*, 45(5):851 – 861, 2012. Text Mining and Natural Language Processing in Pharmacogenomics.

[44] Rune Sætre, Kazuhiro Yoshida, Akane Yakushiji, Yusuke Miyao, Yuichiro Matsubayashi, and Tomoko Ohta. Akane system: protein-protein interaction pairs in biocreative2 challenge, ppi-ips subtask. In *Proceedings of the second biocreative challenge workshop*, volume 209, page 212. Madrid, 2007.

[45] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[46] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.

[47] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015*, pages 1556–1566, 2015.

[48] P. Thompson, S. Daikou, K. Ueno, R. Batista-Navarro, J. Tsujii, and S. Ananiadou. Annotation and detection of drug effects in text for pharmacovigilance. *Journal of Cheminformatics*, 10:37, 2018.

[49] Erik M. van Mulligen, Annie Fourrier-Réglat, David Gurwitz, Mariam Molokhia, Ainhoa Nieto, Gianluca Trifirò, Jan A. Kors, and Laura Inés Furlong. The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics*, 45(5):879–884, 2012.

[50] Chih-Hsuan Wei, Bethany R Harris, Hung-Yu Kao, and Zhiyong Lu. tmvar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, 29(11):1433–1439, 2013.

[51] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522, 2013.

[52] Michelle Whirl-Carrillo, EM McDonagh, JM Hebert, Li Gong, K Sangkuhl, CF Thorn, RB Altman, and Teri E Klein. Pharmacogenomics knowledge for personalized medicine. *Clinical pharmacology and therapeutics*, 92(4):414, 2012.

[53] Thomas C. Wiegers, Allan Peter Davis, and Carolyn J. Mattingly. Collaborative biocuration—text-mining development task for document prioritization for curation. *Database*, 2012:bas037, 2012.

[54] Hong-Guang Xie and Felix W Frueh. Pharmacogenomics steps toward personalized medicine. *Personalized Medicine*, 2(4):325–337, 2005.

[55] Rong Xu and QuanQiu Wang. A knowledge-driven conditional approach to extract pharmacogenomics specific drug-gene relationships from free text. *Journal of Biomedical Informatics*, 45(5):827 – 834, 2012. Text Mining and Natural Language Processing in Pharmacogenomics.

[56] Alexander Yeh, Alexander Morgan, Marc Colosimo, and Lynette Hirschman. Biocreative task 1a: gene mention finding evaluation. *BMC bioinformatics*, 6(1):S2, 2005.