DATABASE
The Journal of Biological Databases and Curation

Database, 2014, 1–10
doi: 10.1093/database/bau073
Original article

Original article

tmBioC: improving interoperability of text-mining tools with BioC

Ritu Khare, Chih-Hsuan Wei, Yuqing Mao, Robert Leaman and Zhiyong Lu*

National Center for Biotechnology Information, National Institutes of Health, 8600 Rockville Pike, Bethesda, MD, USA

*Corresponding author: Tel: +1 301-594-7089; Fax: +1 301-480-2288; Email: zhiyong.lu@nih.gov

Citation details: Khare,R., Wei,C.-H., Mao,Y. *et al. tmBioC*: improving interoperability of text-mining tools with BioC. *Database* (2014) Vol. 2014: article ID bau073; doi:10.1093/database/bau073.

Received 31 January 2014; Revised 30 June 2014; Accepted 1 July 2014

Abstract

The lack of interoperability among biomedical text-mining tools is a major bottleneck in creating more complex applications. Despite the availability of numerous methods and techniques for various text-mining tasks, combining different tools requires substantial efforts and time owing to heterogeneity and variety in data formats. In response, BioC is a recent proposal that offers a minimalistic approach to tool interoperability by stipulating minimal changes to existing tools and applications. BioC is a family of XML formats that define how to present text documents and annotations, and also provides easy-to-use functions to read/write documents in the BioC format. In this study, we introduce our text-mining toolkit, which is designed to perform several challenging and significant tasks in the biomedical domain, and repackage the toolkit into BioC to enhance its interoperability.

Our toolkit consists of six state-of-the-art tools for named-entity recognition, normalization and annotation (*PubTator*) of genes (*GenNorm*), diseases (*DNorm*), mutations (*tmVar*), species (*SR4GN*) and chemicals (*tmChem*). Although developed within the same group, each tool is designed to process input articles and output annotations in a different format. We modify these tools and enable them to read/write data in the proposed BioC format. We find that, using the BioC family of formats and functions, only minimal changes were required to build the newer versions of the tools. The resulting BioC wrapped toolkit, which we have named *tmBioC*, consists of our tools in BioC, an annotated full-text corpus in BioC, and a format detection and conversion tool.

Furthermore, through participation in the 2013 BioCreative IV Interoperability Track, we empirically demonstrate that the tools in *tmBioC* can be more efficiently integrated with each other as well as with external tools: Our experimental results show that using BioC reduces >60% in lines of code for text-mining tool integration. The *tmBioC* toolkit is publicly available at http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/.

Published by Oxford University Press 2014. This work is written by US Government employees and is in the public domain in the US.

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

Database URL: http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/

Introduction

There is an increasing demand of text-mining tools in the biomedical and life sciences domain. Many recent BioNLP challenge tasks (1–5) are focused on extracting structured information from scientific articles and clinical notes. Research groups around the world are developing a variety of stand-alone text-mining tools. Typically, a tool is developed using certain data representation and programming conventions as preferred by the individual research group. To build complex text-mining applications or pipelines, it is often required to combine multiple tools, possibly designed by different groups. The current practice of independent tool development poses a hindrance to tool interoperability and integration. To use a new tool or a new data set, text-mining researchers spend a substantial amount of time developing algorithms for processing the new data format. This heterogeneity in data representation slows down the development of powerful applications and thereby leads to inefficiencies in research and innovation.

There have been quite a few efforts to promote interoperability among text analytics tools and data sets. The BioCreative MetaServer (6) is designed to combine natural language annotations produced by different groups; any group interested in contributing to the central server is required to implement an annotation server based on a predefined three-layered framework consisting of data, communication and application layers. Unstructured Information Management Architecture (UIMA) (7, 8) and General Architecture for Text Engineering (GATE) (9) are two notable proposals that prescribe using a predefined framework to develop text-mining applications to achieve interoperability among independently developed tools. Although type systems such as U-Compare (10) and technical utilities such as UIMAFit (11) have been developed to facilitate easier integration of UIMA-compliant tools, the development of a UIMA- or GATE-compliant application requires the entire tool to be (re-)written into framework-specific constructs and presents a steep learning curve (12). Another approach to accomplish interoperability among annotated corpora is to promote a common output data format. For example, a recent effort in this direction, BioC (13), is based on a minimalist approach in that it offers interoperability by stipulating minimal changes in existing applications or data sets. The goals of BioC are simplicity, reusability, interoperability and wide use. In a nutshell, BioC is a family of XML formats that define how to present text documents and annotations. BioC is different from previously proposed formats, such as IeXML (14, 15), in that BioC also provides tools to read and write documents in the BioC format in multiple programming languages to further minimize the efforts of tool developers.

In this article, we present our efforts on using BioC to repackage the suite of text-mining software and Web-based tools (16–22) developed by the biomedical text-mining group at the National Center for Biotechnology Information (NCBI). Specifically, we wrap five stand-alone biomedical named entity recognition (NER) tools, one Web-based annotation tool and one annotated text corpus, into BioC. In addition, we provide a format converter tool that allows conversions among three different data formats: two tool-specific formats and BioC. By making our toolkit BioC-compatible, we expect to enhance its interoperability and the capacity to develop more sophisticated text-mining applications such as extracting relations between entities (23, 24). We evaluate the BioC-compatible toolkit at two levels: (i) the ability to integrate our tools with each other and (ii) the ability to integrate our tools with tools developed by other groups. This study was conceived and conducted through our participation in the BioCreative IV interoperability task (25, 26).

Method

In this section, we first introduce our toolkit that comprises six tools for concept recognition and annotation (shown in Figure 1), and an annotated text corpus for Gene Ontology (GO) concept recognition. Then, we describe the key steps and challenges in creating a BioC compatible version of the tools and the text corpus.

PubTator

We have developed a Web-based annotation tool called *PubTator* (20, 27, 28) for assisting manual curation. *PubTator* was developed using JavaScript, Perl-CGI, HTML and MSSQL. *PubTator* is synchronized with PubMed® and supports semantic annotation and search of key biomedical entities and their relationships in PubMed articles. Similar to Entrez Programming Utilities (29), *PubTator* also offers programmatic access to its results with a list of parameters specifying the output content and format: (i) a list of PMIDs to be retrieved, (ii) Bioconcepts (‘Gene’, ‘Chemical’, ‘Disease’, ‘Mutation’ and ‘Species’), to be included and (iii) the output format (See Table 1).

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

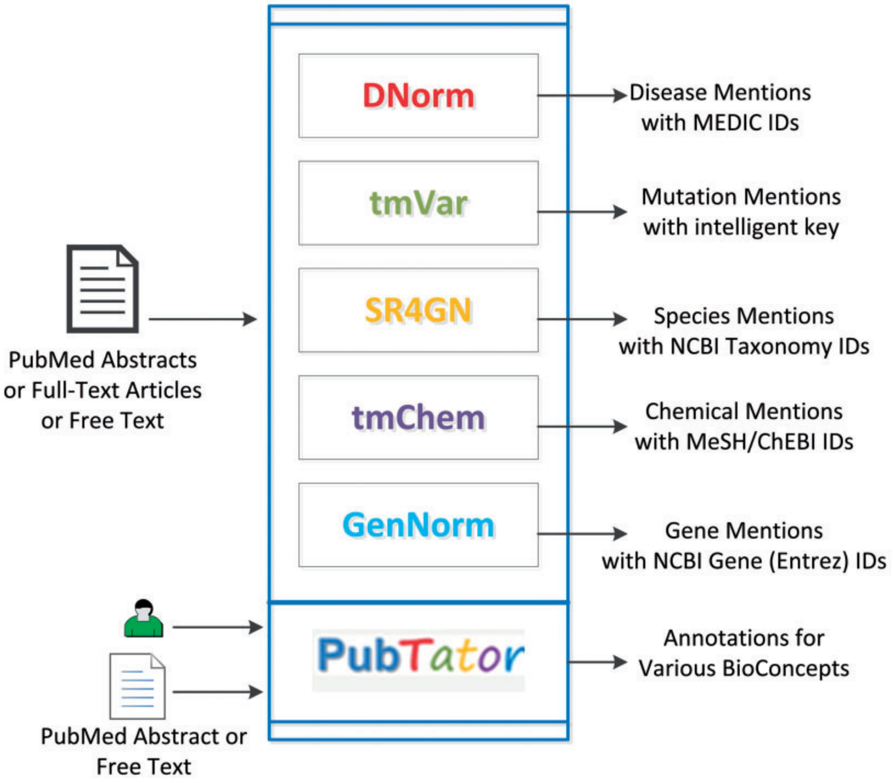

DNorm
tmVar
SR4GN
tmChem
GenNorm
PubTator
Disease Mentions with MEDIC IDs
Mutation Mentions with intelligent key
Species Mentions with NCBI Taxonomy IDs
Chemical Mentions with MeSH/ChEBI IDs
Gene Mentions with NCBI Gene (Entrez) IDs
Annotations for Various BioConcepts
PubMed Abstracts or Full-Text Articles or Free Text
PubMed Abstract or Free Text

Figure 1. Visual summary of our text-mining toolkit.

Table 1. Input/output formats supported by our text-mining toolkit

tmTools	Formats supported (I = Input, O = Output)				
	PMC XML	Free text	Tool-specific format 1[a] (*PubTator*)	Tool-specific format 2[b] (*GenNorm*)	Tool-nonspecific format (BioC)
tmChem	I	I			I/O
DNorm	I	I			I/O
tmVar	I	I	I/O		I/O
SR4GN	I	I		O	I/O
GenNorm	I	I		O	I/O
PubTator	I	I	I/O		I/O

[a] http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/import.example.html.
[b] http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/Format.html#GenNorm.

Concept recognition tools

We have developed several NER tools for automatically recognizing key biomedical concepts, such as chemicals, diseases, genes, mutations and species, from scientific publications (30, 31). Each tool accepts a PubMed or PMC full-text article or free-text as an input and identifies the biomedical entities at both mention and concept levels.

- *DNorm* (1, 19) is an open-source software tool to identify and normalize disease mentions from biomedical texts. *DNorm* is based on the pair-wise learning to rank algorithm and is the first technique to use machine learning for disease normalization. This tool was developed in Java.
- *tmVar* (18) is a machine learning system for mutation recognition to assist biomedical curation. It is based on conditional random fields and identifies many types of mutations and sequence variants in protein, gene, DNA and RNA levels for biomedical curation. This tool was developed in Perl and uses the CRF++ module developed in C++.
- *SR4GN* (16) is a species recognition tool optimized for the gene normalization task. It is a rule-based system that identifies species from full-texts and pairs them with

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

corresponding gene or protein mentions. This tool was developed in Perl.

- *tmChem* (21) is a machine learning-based NER system for chemicals. The system is designed to identify and normalize a wide variety of chemical mentions in literature, including identifiers, brand and trade names and also systematic formats. The system uses conditional random fields with a rich feature set and rule-based post-processing modules for resolving local abbreviations and improving consistency. This tool was developed in Java.
- *GenNorm* (17) is a rule-based tool for gene name recognition, species assignation and species-specific gene normalization. *GenNorm* addresses the challenging issues of orthologous gene ambiguity and intra-species gene ambiguity. This tool was developed in Perl.

The BC4GO corpus

More recently, we developed the *BC4GO* corpus (22) (not shown in the figure), a corpus of 200 full-text articles along with their GO annotations describing genes and gene product attributes across species and databases. As annotations, the corpus presents the evidence sentences along with the gene/protein entities, GO terms and GO evidence codes. The corpus was developed with eight expert biocurators using a Web-based annotation tool. This is the official corpus for the BioCreative IV Track-4 GO Task (32), which tackles the challenge of automatic GO annotation through literature analysis.

Building BioC compatible tools

The BioC family of XML formats and functions comprises the following four items:

(i) The XML Document Type Definition (DTD) that defines the syntax, i.e. how to present text document and annotations (with global offsets) to share common information. The BioC format allows standoff annotations in two ways where users can choose to keep one file or two separate files (for articles and annotations, respectively).

(ii) A key file to describe the semantics, i.e. how to interpret the data (XML elements) in the BioC annotation file. BioC allows many different kinds of annotations to be represented, including title, abstract, sentences, paragraphs, biomedical entities (e.g. gene), etc. A key file is custom-created for a specific type of problem domain.

(iii) C++, Java, Ruby, Python, Go and SWIG libraries that include functions and classes to read and write documents in the BioC format and to hold the documents in memory.

(iv) A format converter to translate an article from the PMC (or PubMed) XML format to the BioC format.

To comply our tools with BioC, we modified the input and output formats of the tools by adding BioC as a new option, and translated the articles and the annotations in our corpus to corresponding BioC files. Table 1 summarizes the formats originally supported by the tools.

PubTator

The original input/output format for *PubTator* was a predefined format, the tool-specific format 1. To make *PubTator* BioC compatible, we added a new format option giving users the option to input and output in the BioC format. For our purposes, we chose to keep the articles and annotations into a single file so that users need to upload a single file when importing annotations into *PubTator*. We defined the **tmBioC.key** file (submitted as Supplemental Data) that describes specific attributes such as bioconcept types, database identifiers, locations and mentions.

Concept recognition tools

The primary effort in converting the various concept recognition tools to BioC was to define an appropriate key file. As the semantics of all these tools were similar to *PubTator* in terms of the type of data, we used the same key file, **tmBioC.key**, which was used for BioC compatible version of *PubTator*. The same key file was used for interpreting the input articles/abstracts and the output articles/abstracts with annotations. Once the key file was finalized, the remaining efforts were to add a new input/output format (i.e. BioC) to the tools.

The mutation recognition tool, *tmVar*, originally accepted the tool-specific format 1, free text and the PMC XML format. The output format was the tool-specific format 1. For *GenNorm* and *SR4GN*, the input formats were free text, PMC XML format and the tool-specific format 2, and the output format was the tool-specific format 2. Translation of these tools into respective BioC versions required additional efforts, as these tools were authored in Perl and the BioC functions were only available in Java and C++ when we conducted this study. Hence, a Perl implementation of BioC functions was implemented for the purpose of this study. Finally, to develop the BioC versions for *tmVar, GenNorm* and *SR4GN*, we added the BioC format as a new option for input/output. The key difference between the tool-specific format 2 and the BioC format is that while BioC recommends specifying global offsets, the

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

tool-specific format 2 calculates offsets for separate sentences. Also, the tool-specific format stipulates separate files not only for articles and annotations but also for different kinds of bioconcepts. Hence, the offset calculation and file writing functions for *GenNorm* and *SR4GN* tools had to be modified accordingly.

The previous output format for *tmChem* was the BioCreative IV CHEMDNER format, which is essentially the same as the tool-specific format 1, representing one NER mention per line. *DNorm* is a relatively new tool and did not previously have a default output format. As both tools are built on top of BANNER (33), input compatibility with BioC only required adding the new data set loading class, provided by the BioC family of tools, in BANNER. Modifying the output required modifying the class containing the main method to output the BioC format.

BC4GO corpus

First, the 200 full-text articles of the *BC4GO* corpus were uploaded to the annotation tool, and eight annotators performed annotation. In the meanwhile, the PMC XML data model formats of those articles were converted to the BioC format using the format converter provided by the BioC tools.

Then, we downloaded the HTML files, and extracted the annotations including annotation texts, genes, GO terms and identified their offsets. There were certain challenges in creating the annotation files that required additional programming efforts. The first challenge was in creating the BioC annotation file using the user annotations downloaded from the Web-based annotation tool. We observed encoding discrepancies in the PMC XML file and the downloaded file. The original article was published in Unicode; the file in PMC XML format was encoded in UTF-8 but was converted to ASCII in the BioC format. Although the annotation results downloaded from the Web were also encoded in ASCII, they were translated using a different Unicode-ASCII converter table. For example, the term ‘neurexin-1α’ (see PMID: 22262843 in corpus) would read ‘neurexin-1alpha’ in the PMC XML file but ‘neurexin-1I+-’in the downloaded annotation file. To maintain consistency between the BioC article and annotation files, we converted the characters using a neighbor matching method as described in (22).

Another challenge was presenting the evidence sentences that contain multiple discontinuous sentences, possibly from different passages in the article. For example, in PMID 19321442, the curator annotated ‘Surprisingly, pep2-SVKI, which blocks both PICK1 and ABP/GRIP PDZ domains, did not completely block the decrease in the rectification index (Figure 2C, pre-OGD = 0.42 6 0.04, post-OGD = 0.37 6 0.03, $n = 6$, $P < 0.05$).’ from the fourth paragraph in the RESULTS section, and ‘We observed a time-dependent reduction in GluR2 surface expression by OGD (Figure 3B), which is consistent with a previous study’ (17) from the sixth paragraph in the same section, as the evidence sentence for GO: 0009986 (cell surface). The two sentences appeared in different passages in the BioC format article, i.e. belonged to different sub-nodes in the XML file. We addressed this challenge by linking these evidence sentences using the same annotation ID, i.e. they are treated as one whole evidence sentence for a GO term.

Finally, for each article we created a corresponding BioC annotation file for the associated GO annotations using the tools provided by BioC. For the gene entity, we provide both the gene mention as appeared in text and its corresponding NCBI Gene identifier. In the BioC released corpus, each article is named by its PubMed identifier, e.g. ‘20130316.xml’. The annotation file associated with the article file shares the same PMID in the file name, e.g. ‘annotation_20130316.xml’. The annotation file includes all annotations of the article; each annotation has a unique ID and is defined by four distinct elements: gene, go-term, go-evidence and type. We define separate key files to describe the full-text articles and the annotation files with GO annotations, namely **pmc_go.key** and **go_annotation.-key**, respectively.

One limitation of the corpus released in BioC is that the BioC annotation file of an article would not contain an evidence sentence that is located in the ‘Acknowledgement’ section of the article (see PMID 18695045) because this section is not provided in the original PMC XML file for the article. Also, in some cases, such as line feeds and hyperlinks, incomplete sentences were created because of the additional space characters in the original PMC XML files. Such cases were manually processed to create consistent BioC annotation files.

Format converters

The BioC family does provide a format converter to translate articles in the PMC XML format to the BioC format. However, there is no converter available to translate the annotation files. Hence, we developed a Perl-based format converter that allows automatic detection and format conversion among three different data formats: tool-specific formats 1 and 2, and BioC. A major issue in implementation of the converter was the discrepancies in file saving conventions adopted by different formats. For a given article with annotations, the tool-specific format 2 generates separate files for the article and for different kinds of bioconcepts such as genes, species, chemicals, etc. In contrast, both tool-specific format 1 and BioC formats prescribe

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

saving sentences and annotated mentions into a single file. Considering these discrepancies, we developed and wrapped multiple small format detectors and converters into a single tool that allows users to translate annotations for a given article into a desired output format.

Results

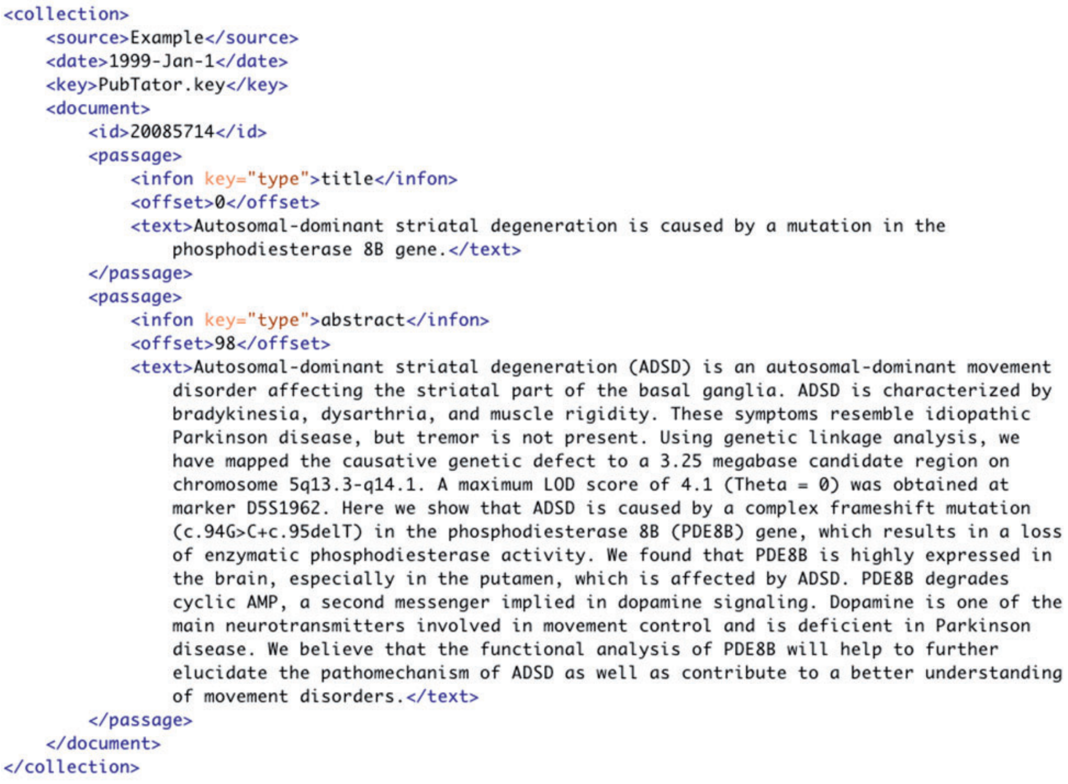
Our upgraded toolkit, which we have named *tmBioC*, comprises the new BioC versions of all tools, the common **tmBioC.key** file, and the format converter. *tmBioC* is made publicly available at http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools. The key file is provided as a Supplementary Data.

To describe the outputs of our concept recognition tools, we use a PubMed abstract (PMID 20085714) that contains mentions of multiple biomedical entities, including genes, mutations, chemicals and diseases, as a running example. A snippet of the BioC input file for this example is shown in Figure 2, and a snippet of BioC output file showing the integrated results from all the tools are displayed in Figure 3. The BioC annotation file contains detailed information about each annotation. For example, the annotation id ‘11’ is represented by the mention ‘c.95de1T’ (text tag) and located in the global offset 679 with a length of 8 characters (location tag). Also, this annotation is a mutation type annotation (infon key = ‘type’) and is normalized to an automatically generated intelligent key, ‘c|DEL|95|T’ (infon key = ‘tmVar’) that captures the fine grained mutation information in the format<Sequence type, mutation type, wild type, mutation position, mutant>. As another example, the annotation id ‘17’ is presented by the mention ‘cyclic AMP’ and located in the global offset 919 with a length of 10 characters; this annotation is a chemical type and is normalized to a concept in the MeSH vocabulary, ‘D000242’.

The BioC version of the *BC4GO* corpus, with 200 BioC article files and 200 BioC annotation files, can be downloaded at the BioCreative IV Track 4 task’s official webpage, http://www.biocreative.org/tasks/biocreative-iv/track-4-GO/. The key files, **pmc_go.key** and **go_annotation.key**, were submitted as part of the BioCreative IV Track 1 submission. These key files are also provided as Supplemental Data. A snippet of the BioC annotation file corresponding to the PubMed article with PMID 23840682 is shown in Figure 4.

Interoperability evaluation

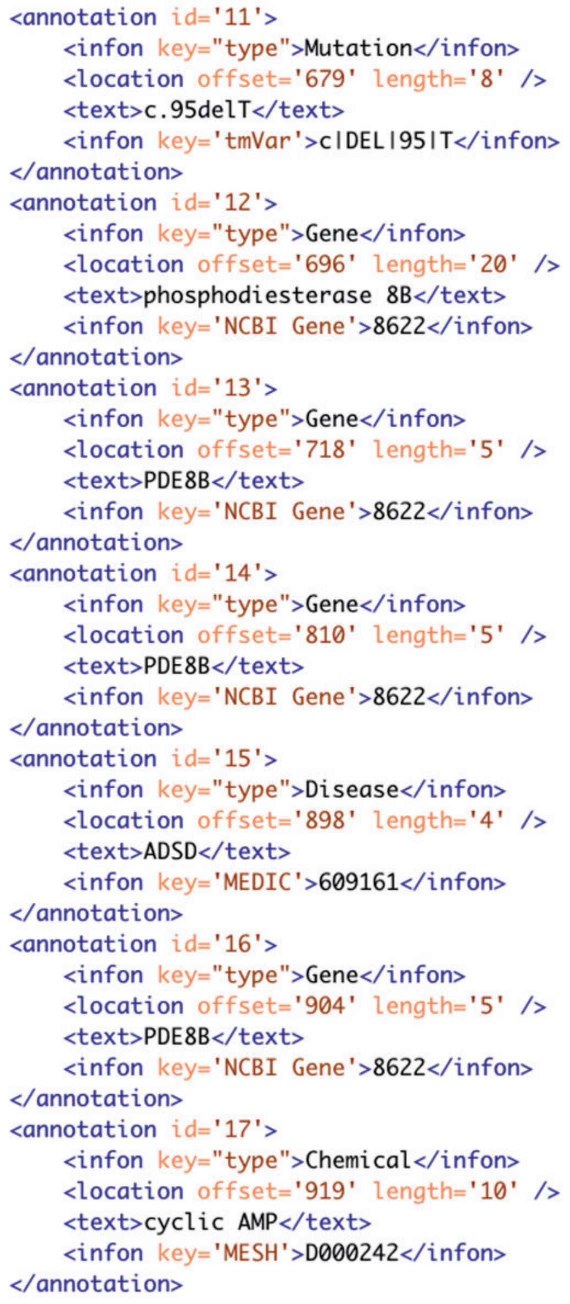
To demonstrate the interoperability of *tmBioC*, we conducted two experiments, with internal and external tools, respectively. For studying the interoperability of *tmBioC*



```
<collection>
    <source>Example</source>
    <date>1999-Jan-1</date>
    <key>PubTator.key</key>
    <document>
        <id>20085714</id>
        <passage>
            <infon key="type">title</infon>
            <offset>0</offset>
            <text>Autosomal-dominant striatal degeneration is caused by a mutation in the
                phosphodiesterase 8B gene.</text>
        </passage>
        <passage>
            <infon key="type">abstract</infon>
            <offset>98</offset>
            <text>Autosomal-dominant striatal degeneration (ADSD) is an autosomal-dominant movement
                disorder affecting the striatal part of the basal ganglia. ADSD is characterized by
                bradykinesia, dysarthria, and muscle rigidity. These symptoms resemble idiopathic
                Parkinson disease, but tremor is not present. Using genetic linkage analysis, we
                have mapped the causative genetic defect to a 3.25 megabase candidate region on
                chromosome 5q13.3-q14.1. A maximum LOD score of 4.1 (Theta = 0) was obtained at
                marker D5S1962. Here we show that ADSD is caused by a complex frameshift mutation
                (c.94G>C+c.95delT) in the phosphodiesterase 8B (PDE8B) gene, which results in a loss
                of enzymatic phosphodiesterase activity. We found that PDE8B is highly expressed in
                the brain, especially in the putamen, which is affected by ADSD. PDE8B degrades
                cyclic AMP, a second messenger implied in dopamine signaling. Dopamine is one of the
                main neurotransmitters involved in movement control and is deficient in Parkinson
                disease. We believe that the functional analysis of PDE8B will help to further
                elucidate the pathomechanism of ADSD as well as contribute to a better understanding
                of movement disorders.</text>
        </passage>
    </document>
</collection>
```

Figure 2. A snippet of the BioC article file for PMID 20085714.

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024



```
<annotation id='11'>
    <infon key="type">Mutation</infon>
    <location offset='679' length='8' />
    <text>c.95delT</text>
    <infon key='tmVar'>c|DEL|95|T</infon>
</annotation>
<annotation id='12'>
    <infon key="type">Gene</infon>
    <location offset='696' length='20' />
    <text>phosphodiesterase 8B</text>
    <infon key='NCBI Gene'>8622</infon>
</annotation>
<annotation id='13'>
    <infon key="type">Gene</infon>
    <location offset='718' length='5' />
    <text>PDE8B</text>
    <infon key='NCBI Gene'>8622</infon>
</annotation>
<annotation id='14'>
    <infon key="type">Gene</infon>
    <location offset='810' length='5' />
    <text>PDE8B</text>
    <infon key='NCBI Gene'>8622</infon>
</annotation>
<annotation id='15'>
    <infon key="type">Disease</infon>
    <location offset='898' length='4' />
    <text>ADSD</text>
    <infon key='MEDIC'>609161</infon>
</annotation>
<annotation id='16'>
    <infon key="type">Gene</infon>
    <location offset='904' length='5' />
    <text>PDE8B</text>
    <infon key='NCBI Gene'>8622</infon>
</annotation>
<annotation id='17'>
    <infon key="type">Chemical</infon>
    <location offset='919' length='10' />
    <text>cyclic AMP</text>
    <infon key='MESH'>D000242</infon>
</annotation>
```

Figure 3. A snippet from the BioC annotation file for PMID 20085714 (integrated result of applying our five concept recognition tools on the abstract).

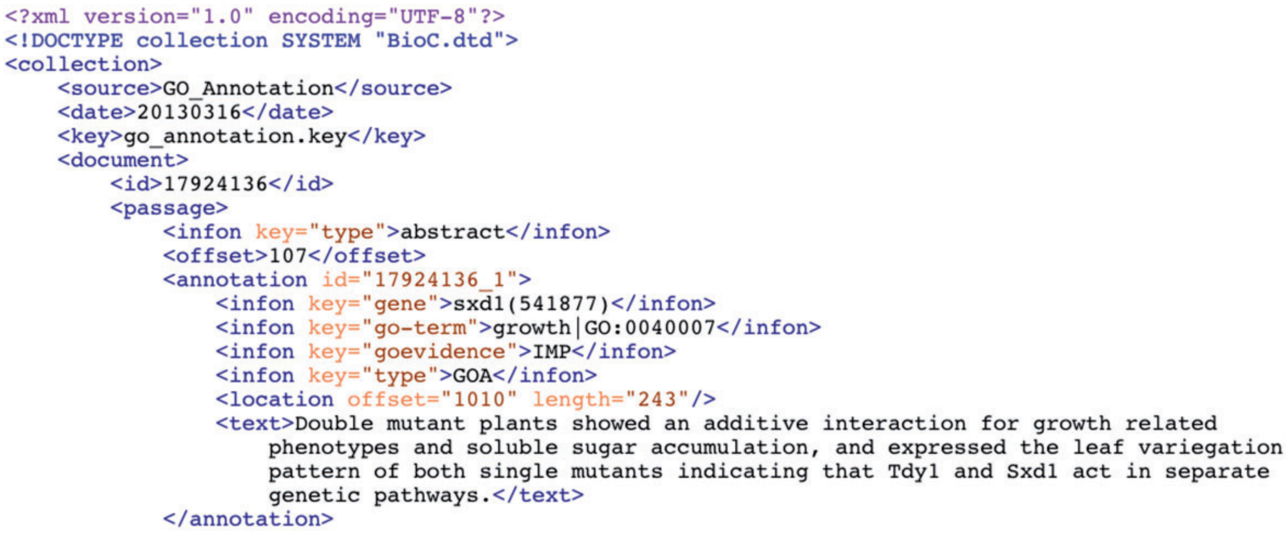
within the toolkit, we integrated *PubTator* with all the five concept recognition tools, *tmVar*, *tmChem*, *GenNorm*, *SR4GN* and *DNorm*, as illustrated in Figure 5. Our previous attempt of integrating the original versions of tools was time-consuming and technically challenging because of the various discrepancies among the tools. As the tools were developed in at least three different programming languages (Java, Perl and C++), they had defined their own data formats, and adopted different offset calculation schemes (global vs. local), as mentioned before. Accordingly, we had to develop five different interfacing modules as illustrated in Figure 5a. In contrast, the integration of the respective BioC versions of all the independently developed tools (Figure 5b) was simply a matter of positioning the tools into a pipeline, as the tools could communicate seamlessly through BioC. Compared with our previous integration effort, using BioC saves >60% of lines of code in programming efforts.

For studying the interoperability of *tmBioC* with externally developed tools, we selected *DNorm*, the disease recognition and normalization tool from our toolkit; an externally developed tool, Abbreviations Plus Pseudo-Precision (Ab3P) (34); and a gold standard resource for disease annotation, the NCBI disease corpus (35). Ab3P is an abbreviation resolution tool that identifies <long form, short form> pairs from biomedical texts. Ab3P incorporates a pattern-matching-based algorithm implemented in C++ and has recently been equipped to process and output the BioC format. Abbreviations in biomedical domain are frequently ambiguous, leading to a number of NER errors. Hence, we added Ab3P as a preprocessing step to *DNorm* that replaced the short forms with long forms in the input articles. For this experiment, we trained and tested *DNorm* on the NCBI disease training and test corpora, recently made available in BioC, respectively. We observed 1.3% improvement in abstract-level normalization precision, over the preintegration version of *DNorm*. Similar to the first experiment, we were able to easily integrate and place the three BioC-compatible tools, namely *DNorm*, AB3P and NCBI Disease Corpus, and resources into the pipeline for supervised disease recognition and normalization system.

Discussion and conclusions

The goal of this study was to improve the interoperability of our NER tools using the recently developed BioC Family of XML formats and classes. Our toolkit consists of several competition winning, high-performing tools for concept recognition and annotation. For example, *GenNorm* obtained the highest performance in the BioCreative III Gene Normalization task (36), and *DNorm* achieved the best results the 2013 ShARe/CLEF shared task for normalizing disease names in clinical notes (1). Also, the *tmVar* tool for mutation recognition delivers >90% F-measure on multiple benchmarking test sets; and the PubMed-like, color-coded interface of *PubTator* makes it a highly usable annotation tool for human biocurators. In addition to accelerating knowledge discovery and assisting manual curation, the toolkit is capable of solving other important and challenging problems in the biomedical

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE collection SYSTEM "BioC.dtd">
<collection>
    <source>GO_Annotation</source>
    <date>20130316</date>
    <key>go_annotation.key</key>
    <document>
        <id>17924136</id>
        <passage>
            <infon key="type">abstract</infon>
            <offset>107</offset>
            <annotation id="17924136_1">
                <infon key="gene">sxd1(541877)</infon>
                <infon key="go-term">growth|GO:0040007</infon>
                <infon key="goevidence">IMP</infon>
                <infon key="type">GOA</infon>
                <location offset="1010" length="243"/>
                <text>Double mutant plants showed an additive interaction for growth related
                    phenotypes and soluble sugar accumulation, and expressed the leaf variegation
                    pattern of both single mutants indicating that Tdy1 and Sxd1 act in separate
                    genetic pathways.</text>
            </annotation>
```

Figure 4. A Snippet from the file annotation_23840682.xml from the BC4GO corpus

domain. For instance, text-mining mutation information is critical for the analysis and interpretation of sequence variations in complex diseases in the post-genomic era. Disease recognition is important for many lines of inquiry, including etiology (e.g. gene–disease relationships) and clinical aspects (e.g. diagnosis, prevention and treatment). Gene and specifies recognition could be useful for protein–protein interaction extraction.

Our experience shows that only minimal changes were required to repackage our tools with BioC and deliver the final product: *tmBioC*. Also, reading and writing to BioC format was fairly straightforward, as the functions and classes are already provided by the BioC authors in two widely used programming languages. For each tool, the primary developers modified their respective tools, and confirmed the simplicity and learnability of the BioC format. The primary challenge was to create the key files for the tools. However, it was a one-time effort, as all the six concept recognition and annotation tools can use a common key file for defining their BioC annotation files. The released **tmBioC.key** file could also evolve as a standard key file for concept recognition and annotation tasks as recommended in (37). Furthermore, we provide an easy-to-use format converter that facilitates automatic conversion among three data formats including BioC. We also provide a full-text corpus of GO annotations in BioC, which can be used to train other NER tools. All our tools are freely available and ready to be reused by a wider community of researchers in text mining, bioinformatics and biocuration communities.

Through this study, we promote the interoperability of our tools, not only with each other, but also with the tools and datasets developed by several other groups worldwide. The tools, although developed in different programming languages and following different data representation schemes, are now capable of sharing their inputs/outputs with each other, without any additional programming efforts. We confirmed this by equipping the BioC version of *PubTator* annotation tool with five other tools from our BioC compatible toolkit. Our tools in BioC can also interact with other state-of-the-art tools to build much more powerful applications. We confirm this by combining one of our tools in BioC with a high-precision abbreviation resolution tool in BioC, and a human-annotated disease corpus in BioC.

Our interoperability experiments only present a glimpse of the powerful applications that could be created by synergistically combining multiple tools. For example, a modular text-mining pipeline of various BioC compatible tools and corpora for NER, normalization, annotation and relationship extraction, could be developed to build sophisticated systems, e.g. an integrative disease-centered system connecting the biological and clinical aspects, providing information from causes (gene–mutation–disease relationship) to treatment (drug–disease relationships) of diseases by mining/annotating unstructured (biomedical literature, clinical notes, etc.) and structured resources (datasets released by organizations and research groups). In the future, we anticipate much broader usage of these tools, as further efforts are invested in publicizing BioC.

Supplementary data

Supplementary data are available at *Database* Online.

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

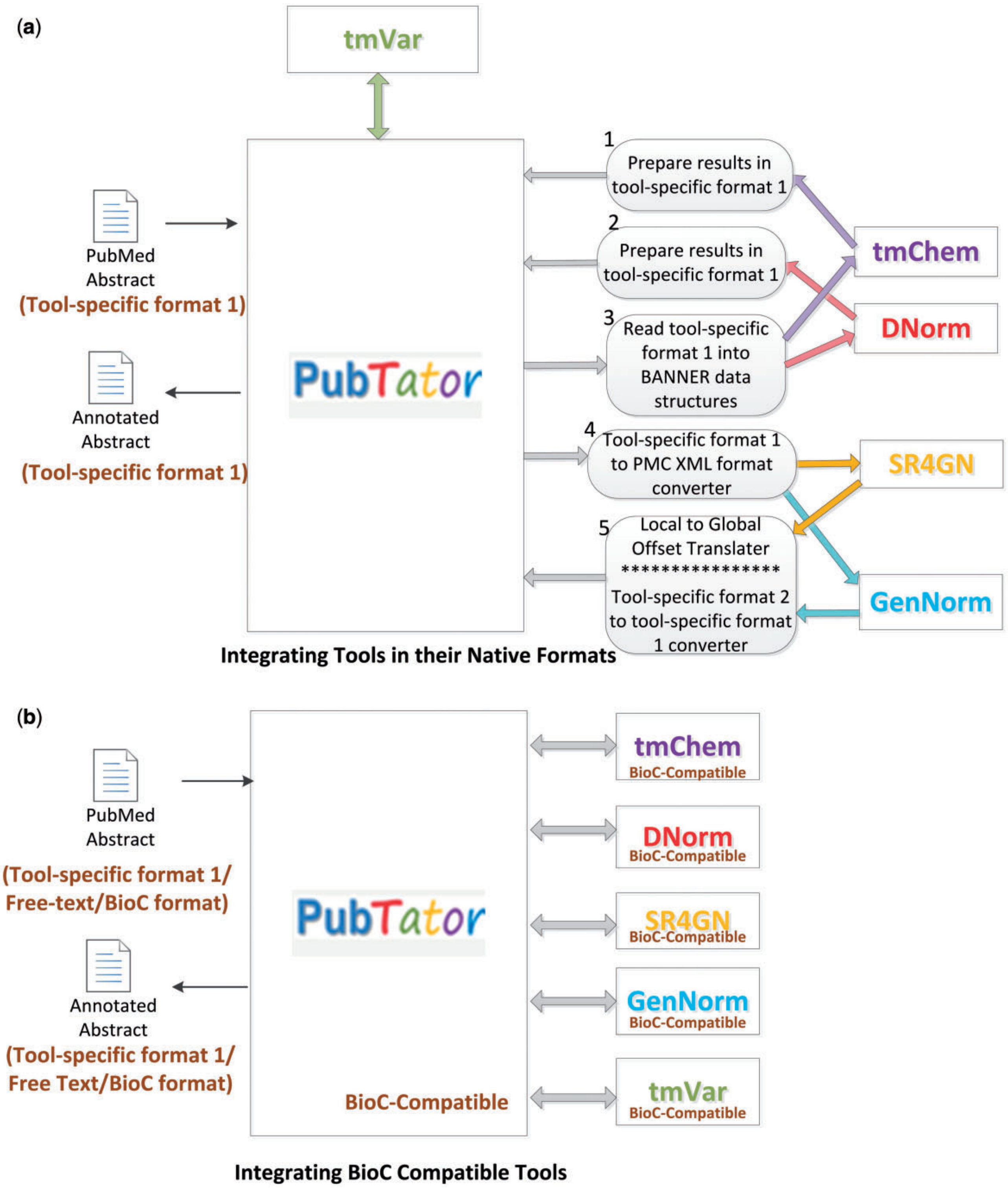

(a)
tmVar
PubMed
Abstract
(Tool-specific format 1)
Annotated
Abstract
(Tool-specific format 1)
PubTator
1
Prepare results in tool-specific format 1
2
Prepare results in tool-specific format 1
3
Read tool-specific format 1 into BANNER data structures
4
Tool-specific format 1 to PMC XML format converter
5
Local to Global Offset Translater
****
Tool-specific format 2 to tool-specific format 1 converter
tmChem
DNorm
SR4GN
GenNorm
Integrating Tools in their Native Formats
(b)
PubMed
Abstract
(Tool-specific format 1/ Free-text/BioC format)
Annotated
Abstract
(Tool-specific format 1/ Free Text/BioC format)
PubTator
BioC-Compatible
tmChem
BioC-Compatible
DNorm
BioC-Compatible
SR4GN
BioC-Compatible
GenNorm
BioC-Compatible
tmVar
BioC-Compatible
Integrating BioC Compatible Tools

Figure 5. The intra-toolkit interoperability experiment. (**a**) Integrating tools in their native formats. (**b**) Integrating BIOC compatible tools.

Acknowledgements

The authors would like to thank Don Comeau, Rezarta Dogan and John Wilbur for their discussion and help with the BioC tools, and in particular, their help on preparing the PMC articles in BioC XML format for the BioCreative IV GO task.

Funding

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

Conflict of interest. None declared.

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024

References

1. Leaman,R., Khare,R. and Lu,Z. (2013) NCBI at 2013 ShARe/CLEF eHealth Shared Task: disorder normalization in clinical notes with DNorm. *Conference and Labs of the Evaluation Forum 2013 Working Notes*. Valencia, Spain.
2. Lu,Z., Kao,H.Y., Wei,C.H. *et al.* (2011) The gene normalization task in BioCreative III. *BMC Bioinformatics*, 12 (Suppl 8), S2.
3. Morgan,A.A., Lu,Z., Wang,X. *et al.* (2008) Overview of BioCreative II gene normalization. *Genome Biol.*, 9 (Suppl 2), S3.
4. Krallinger,M., Vazquez,M., Leitner,F. *et al.* (2011) The protein-protein interaction tasks of BioCreative III: classification/ranking of articles and linking bio-ontology concepts to full text. *BMC Bioinformatics*, 12 (Suppl 8), S3.
5. Mork,J.G., Bodenreider,O., Demner-Fushman,D. *et al.* (2010) Extracting Rx information from clinical narrative. *J. Am. Med. Inform. Assoc.*, 17, 536–539.
6. Leitner,F., Krallinger,M., Rodriguez-Penagos,C. *et al.* (2008) Introducing meta-services for biomedical information extraction. *Genome Biol.*, 9 (Suppl 2), S6.
7. Ferrucci,D. and Lally,A. (2004) UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10, 327–348.
8. Ferrucci,D., Lally,A., Gruhl,D. *et al.* (2006) *Towards an Interoperability Standard for Text and Multi-Modal Analytics*. IBM Research Report, RC24122 (W0611-188)
9. GATE. (2010) *GATE: General Architecture for Text Engineering*. The University of Sheffield. http://gate.ac.uk/.
10. Kano,Y., Baumgartner,W.A. Jr., McCrohon,L. *et al.* (2009) U-Compare: share and compare text mining tools with UIMA. *Bioinformatics*, 25, 1997–1998.
11. Ogren, P., Bethard, S. (2009) UIMAFit. Building Test Suites for (UIMA) Components. Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009). Boulder, Colorado. pp. 1–4
12. Stubbs,A. (2011) MAE and MAI: lightweight annotation and adjudication tools. *Proceedings of the 5th Linguistic Annotation Workshop*. Portland, Oregon, pp. 129–133.
13. Comeau,D.C., Doğan,R.I., Ciccarese,P. *et al.* (2013) BioC: a minimalist approach to Interoperability for biomedical text processing. *Database* (Oxford), 2013: bat064.
14. Rebholz-Schuhmann,D., Kirsch,H. and Nenadic,G. (2006) IeXML: towards an annotation framework for biomedical semantic types enabling interoperability of text processing modules. *Joint BioLINK and Bio-Ontologies SIG Meeting, (ISMB 2006)*. Fortaleza, Brazil.
15. Rebholz-Schuhmann,D., Jimeno Yepes,A.J., Van Mulligen,E.M. *et al.* (2010) CALBC silver standard corpus. *J. Bioinform. Comput. Biol.*, 8, 163–179.
16. Wei,C.H., Kao,H.Y. and Lu,Z. (2012) SR4GN: a species recognition software tool for gene normalization. *PLoS One*, 7, e38460.
17. Wei,C.H. and Kao,H.Y. (2011) Cross-species gene normalization by species inference. *BMC Bioinformatics*, 12 (Suppl 8), S5.
18. Wei,C.H., Harris,B.R., Kao,H.Y. *et al.* (2013) tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, 29, 1433–1439.
19. Leaman,R., Islamaj Dogan,R. and Lu,Z. (2013) DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29, 2909–2917.
20. Wei,C.H., Kao,H.Y. and Lu,Z. (2013) PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Res.*, 41, W518–W522.
21. Leaman,R., Wei,C.H., Lu,Z. (2013) NCBI at the BioCreative IV CHEMDNER Task: recognizing chemical names in PubMed articles with tmChem. *Proceedings of BioCreative IV*. Bethesda, Maryland.
22. Auken,K.V., Schaeffer,M.L., McQuilton,P. *et al.* (2013) Corpus Construction for the BioCreative IV GO Task. *Proceedings of BioCreative IV*. Bethesda, Maryland.
23. Li,J. and Lu,Z. (2012) Systematic identification of pharmacogenomics information from clinical trials. *J. Biomed. Inform.*, 2012: 45, 870–878.
24. Islamaj Dogan,R., Neveol,A. and Lu,Z. (2011) A context-blocks model for identifying clinical relationships in patient records. *BMC Bioinformatics*, 12 (Suppl 3), S3.
25. Khare,R., Wei,C.H., Mao,Y. *et al.* (2013) Improving interoperability of text mining tools with BioC, BioCreative IV Workshop. *BioCreative IV Workshop*, Bethesda, MD. pp. 10–22.
26. Arighi,C.N., Wu,C.H., Cohen,K.B. *et al.* (2014) BioCreative-IV virtual issue. *Database*, 2014, bau039.
27. Wei,C.H., Harris,B.R., Li,D. *et al.* (2012) Accelerating literature curation with text-mining tools: a case study of using PubTator to curate genes in PubMed abstracts. *Database*, 2012, bas041.
28. Wei,C.H., Kao, H.Y. and Lu,Z. (2012) PubTator: a PubMed-like interactive curation system for document triage and literature curation. *Proceedings of BioCreative 2012 Workshop*. Washington DC, USA. pp. 145–150.
29. Entrez Programming Utilities Help. National Center for Biotechnology Information (US), https://www.ncbi.nlm.nih.gov/books/NBK25501/.
30. Islamaj Dogan,R., Murray,G.C., Neveol,A. *et al.* (2009) Understanding PubMed user search behavior through log analysis. *Database*, 2009, bap018.
31. Neveol,A., Islamaj Dogan,R. and Lu,Z. (2011) Semi-automatic semantic annotation of PubMed queries: a study on quality, efficiency, satisfaction. *J. Biomed. Inform.*, 44, 310–318.
32. Mao,Y., Auken,K.V., Li,D. *et al.* (2013) The Gene Ontology Task at BioCreative IV. *Proceedings of the BioCreative IV Workshop*, Bethesda, MD.
33. Leaman,R. and Gonzalez,G. (2008) BANNER: an executable survey of advances in biomedical named entity recognition. *Pac. Symp. Biocomput,* 2008, 652–663
34. Sohn,S., Comeau, D.C., Kim, W. *et al.* (2008) Abbreviation definition identification based on automatic precision estimates. *BMC Bioinformatics*, 9, 402.
35. Dogan,R.I., Leaman,R. and Lu,Z. (2014) NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Inform.*, 47, 1–10.
36. Wei,C.H. and Kao,H.Y. (2010) Inference network method on cross species gene normalization in full-text articles. *Procceding of BioCreative III Workshop*, Bethesda, Maryland. pp. 73–81.
37. Arighi,C.N., Carterette,B., Cohen, K.B. *et al.* (2013) An overview of the BioCreative 2012 Workshop Track III: interactive text mining task. *Database*, 2013, bas056.

Downloaded from https://academic.oup.com/database/article/doi/10.1093/database/bau073/2634750 by guest on 03 May 2024