pyMeSHSim: an integrative python package to realize biomedical named entity recognition, normalization and comparison

Zhi-Hui Luo[1], Meng-Wei Shi[1], Zhuang Yang[1], Hong-Yu Zhang[2]*, Zhen-Xia Chen[1]*

[1]College of Life Science and Technology, Huazhong Agricultural University, Wuhan, Hubei 430070, PR China

[2]Hubei Key Laboratory of Agricultural Bioinformatics, College of Informatics, Huazhong Agricultural University, Wuhan 430070, P. R. China.

*To whom correspondence should be addressed.

Running head: MeSH toolkit pyMeSHSim

ABSTRACT

Summary: Computing semantic similarity between two texts, like disease descriptions, has become important for many biomedical text mining applications. Here, we present PyMeSHSim, which is an integrative, lightweight and data-rich MeSH toolkit that recognizes biomedical named entities (bio-NEs) from texts, maps the bio-NEs to the controlled vocabulary MeSH and measures the semantic similarity between the MeSH terms.

Availability: Package's source code and test datasets are available under the GPLv3 license at https://github.com/luozhhub/pyMeSHSim

Contact: zhen-xia.chen@mail.hzau.edu.cn or zhy630@mail.hzau.edu.cn

1 Introduction

Biomedical Named entity (bio-NE) recognition, normalization and comparison are fundamental and important tasks for extracting and utilizing valuable biomedical knowledge from textual data (Nelson, et al., 2015; Tsuyuzaki, et al., 2015; Wang, et al., 2015; Zemojtel, et al., 2014). They are realized by identifying key entities in unstructured text, mapping identified entities to a controlled vocabulary, and measuring the semantic similarity between the controlled vocabulary terms (Leaman, et al., 2015).

Medical Subject Headings (MeSH) is a controlled vocabulary that can be used in bio-NE recognition, normalization and comparison. It consists of three main record types including descriptor records, qualifier records and supplementary concept records (SCRs). MeSH is curated by the National Library of Medicine (NLM) and serves as the index system in PubMed/MEDLINE and other NLM database. NLM has used Medical Text Indexer (MTI) to provide indexing recommendations based on MeSH in the bio-NE recognition for literatures since 2002. As precise literature annotations, MeSH has become more and more popular in the normalization of bio-NE, including disease names, in medical and genetic public databases (Cui, et al., 2018; Pinero, et al., 2017). Also, the structure of MeSH as a directed acyclic graph like Gene Ontology (Harris, et al., 2004) and Disease Ontology (Schriml, et al., 2012) enables the comparison of semantic similarity between two MeSH terms in the graph.

Several MeSH tools have been developed to realize bio-NE recognition, normalization or comparison. As a MeSH tool for bio-NE recognition and normalization, NCBI MeSH has provided a browser online (https://www.ncbi.nlm.nih.gov/mesh) to parse MeSH terms from the input sentence. However, it is neither tolerant to even subtle difference of input phrases from MeSH terms, nor applicable to batch processing. As MeSH tools for bio-NE comparison, meshes (Yu, 2018) and meshSim (Zhou, et al., 2015) have been developed recently to measure MeSH semantic similarity using the R dataset MeSH.Hsa.eg.db (Tsuyuzaki, et al., 2015) as data framework. However, MeSH.Hsa.eg.db has not been updated for three years. Moreover, the lack of SCRs embedded in its MeSH dataset limits the use of both bio-NE comparison tools for rare diseases such as “alzheimer’s disease 7” and “Bardet-Biedl syndrome 11”. Furthermore, there is still a lack of an integrated one-stop MeSH toolkit to realize bio-NE recognition, normalization and comparison.

To solve above problems, an integrative python package pyMeSHSim was developed to realize bio-NE recognition, normalization and comparison. It can directly curate MeSH terms from free biomedical descriptions and measure the semantic similarity between the descriptions based on MeSH. Additionally, to enable batch processing and the application of pyMeSHSim to both common diseases and rare diseases, a lightweight and comprehensive MeSH dataset, including both main headings (MHs) and SCRs, was generated and embedded as the data framework of pyMeSHSim.

2 Implementation

The pyMeSHSim consists of three subpackages (1) the data subpackage that contains a MeSH dataset, (2) the metamapWrap subpackage that recognizes bio-NEs from the text, and

converts bio-NEs into MeSH terms, and (3) the Sim subpackage that measures the distance between MeSH terms (Figure 1).

2.1 Dataset construction

A comprehensive MeSH dataset is fundamental to MeSH tools, while the MeSH dataset used by most popular MeSH tools contains only MeSH MHs but not SCRs. We thus constructed a MeSH dataset that includes both MHs and SCRs, as well as the relationship between MHs and SCRs. We also made the dataset lightweight for users' convenience.

The MeSH dataset was embedded in the data subpackage in bcolz format with a corresponding data interface (Supplementary table 1). It included five tables: (1) table "MainHeadingDetailData" stored all the MH information, including MeSH unique id, tree code, prefer name, category, term semantic type, IC frequency, and UMLS id; (2) table "supplementMainHeading" stored all the UMLS concepts related to MHs; (3) table "RNDetailData" stored the basic information of SCRs; (4) table "RNandRBRel" stored the narrow-and-broad relationship between SCRs and MHs; and (5) table "ParentChildRel" stored the fundamental tree structure. The five tables enabled the conversion of UMLS concepts to MeSH terms, and the measurement of the semantic similarity between MeSH terms (Supplementary table 2).

2.2 Bio-NE recognition and normalization

The bio-NE recognition and normalization of pyMeSHSim were realized by the *metamapWrap* subpackage, which was a wrapper for MetaMap (Aronson and Lang, 2010). MetaMap can curate another broadly used ontology system Unified Medical Language System (UMLS) (Bodenreider, 2004) concepts instead of MeSH terms from text. The subpackage *metamapWrap* curated UMLS concepts from text and then converted UMLS concepts to corresponding MeSH terms via the data subpackage. Users can tune a series of parameters to get more accurate results.

2.3 Bio-NE comparison

The bio-NE comparison of pyMeSHSim was realized by the *Sim* subpackage via measuring the distance between MHs. Narrow records from the SCRs were converted to their broad terms from MHs before the measurement. As the tool meshes, pyMeSHSim offered five representative methods of semantic similarity measures, including four information content (IC) based (Lin's (Lin, 1998), Resnik's (Resnik, 1995), Schlicker's (Schlicker, et al., 2006), and Jiang and Conrath's (Jiang and Conrath, 1997)) and one graph-based (Wang's (Wang, et al., 2007)) algorithms. The IC value of a MeSH term was calculated as the frequency of its occurrence retrieved directly from the index annotation of PubMed publications.

3 Results

We tested the performance of pyMeSHSim with 461 GWAS phenotypes from GWASdb (Li, et al., 2012), which were manually mapped to MeSH terms by Nelson's group using MeSH browser. PyMeSHSim successfully curated MeSH terms from 443 (96%) GWAS phenotypes, and the curated MeSH terms from 276 (60%) GWAS phenotypes were in consistent with

Nelson's group, and 329 (71%) were in the same category as Nelson's group, demonstrating high performance of pyMeSHSim (Supplementary table S2).

Comparing the inconsistent results of the remaining 114 GWAS phenotypes, we found that Nelson's group preferred mapping the phenotypes to disease category (C). For example, "Color, Eye" was returned by MeSH browser with GWAS phenotype "Eye color" as the input. However, "Eye color" was curated as "color vision defects" by Nelson's group, while as "Color, Eye" by pyMeSHSim. Similarly, GWAS phenotypes "Hair color" and "Serum urate" were curated as "hair diseases" and "urinary calculi" by Nelson's group, while as "Color, Hair" and "Acid, Uric" by pyMeSHSim (Supplementary table S2). Therefore, at least a part of the differences between the manual work and pyMeSHSim were from the human bias in the manual work.

4 Conclusion

We developed pyMeSHSim, an integrative, lightweight and data-rich python package for biomedical text mining. To the best of our knowledge, this is the first universal MeSH toolkit for integrated bio-NE recognition, normalization and comparison analysis. PyMeSHSim is expected to be widely used as a powerful tool in bioinformatics, computational biology and biomedical research.

Funding

This work was supported by Huazhong Agricultural University Scientific & Technological Self-innovation Foundation [2016RC011]; and the Fundamental Research Funds for the Central Universities [2662018PY021, 2662017PY115].

Conflict of Interest: none declared.

References

Aronson, A.R. and Lang, F.M. (2010) An overview of MetaMap: historical perspective and recent advances, *Journal of the American Medical Informatics Association : JAMIA*, **17**, 229-236.

Bodenreider, O. (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology, *Nucleic acids research*, **32**, D267-270.

Cui, T.*, et al.* (2018) MNDR v2.0: an updated resource of ncRNA-disease associations in mammals, *Nucleic acids research*, **46**, D371-D374.

Harris, M.A.*, et al.* (2004) The Gene Ontology (GO) database and informatics resource, *Nucleic acids research*, **32**, D258-261.

Jiang, J.J. and Conrath, D.W. (1997) Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, *Computer Science*.

Leaman, R., Khare, R. and Lu, Z. (2015) Challenges in clinical natural language processing for automated disorder normalization, *Journal of biomedical informatics*, **57**, 28-37.

Li, M.J.*, et al.* (2012) GWASdb: a database for human genetic variants identified by genome-wide association studies, *Nucleic acids research*, **40**, D1047-1054.

Lin, D. (1998) An Information-Theoretic Definition of Similarity. *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., pp. 296-304.

Nelson, M.R.*, et al.* (2015) The support of human genetic evidence for approved drug indications, *Nature genetics*, **47**, 856-860.

Pinero, J.*, et al.* (2017) DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants, *Nucleic acids research*, **45**, D833-D839.

Resnik, P. (1995) Using Information Content to Evaluate Semantic Similarity in a Taxonomy, *CoRR*, **abs/cmp-lg/9511007**.

Schlicker, A.*, et al.* (2006) A new measure for functional similarity of gene products based on Gene Ontology, *BMC bioinformatics*, **7**, 302.

Schriml, L.M.*, et al.* (2012) Disease Ontology: a backbone for disease semantic integration, *Nucleic acids research*, **40**, D940-946.

Tsuyuzaki, K.*, et al.* (2015) MeSH ORA framework: R/Bioconductor packages to support MeSH over-representation analysis, *BMC bioinformatics*, **16**, 45.

Wang, H.*, et al.* (2015) Mining drug-disease relationships as a complement to medical genetics-based drug repositioning: Where a recommendation system meets genome-wide association studies, *Clinical pharmacology and therapeutics*, **97**, 451-454.

Wang, J.Z.*, et al.* (2007) A new method to measure the semantic similarity of GO terms, *Bioinformatics*, **23**, 1274-1281.

Yu, G. (2018) Using meshes for MeSH term enrichment and semantic analyses, *Bioinformatics*.

Zemojtel, T.*, et al.* (2014) Effective diagnosis of genetic disease by computational phenotype analysis of the disease-associated genome, *Science translational medicine*, **6**, 252ra123.

Zhou, J.*, et al.* (2015) MeSHSim: An R/Bioconductor package for measuring semantic similarity over MeSH headings and MEDLINE documents, *Journal of bioinformatics and computational biology*, **13**, 1542002.

FIGURES

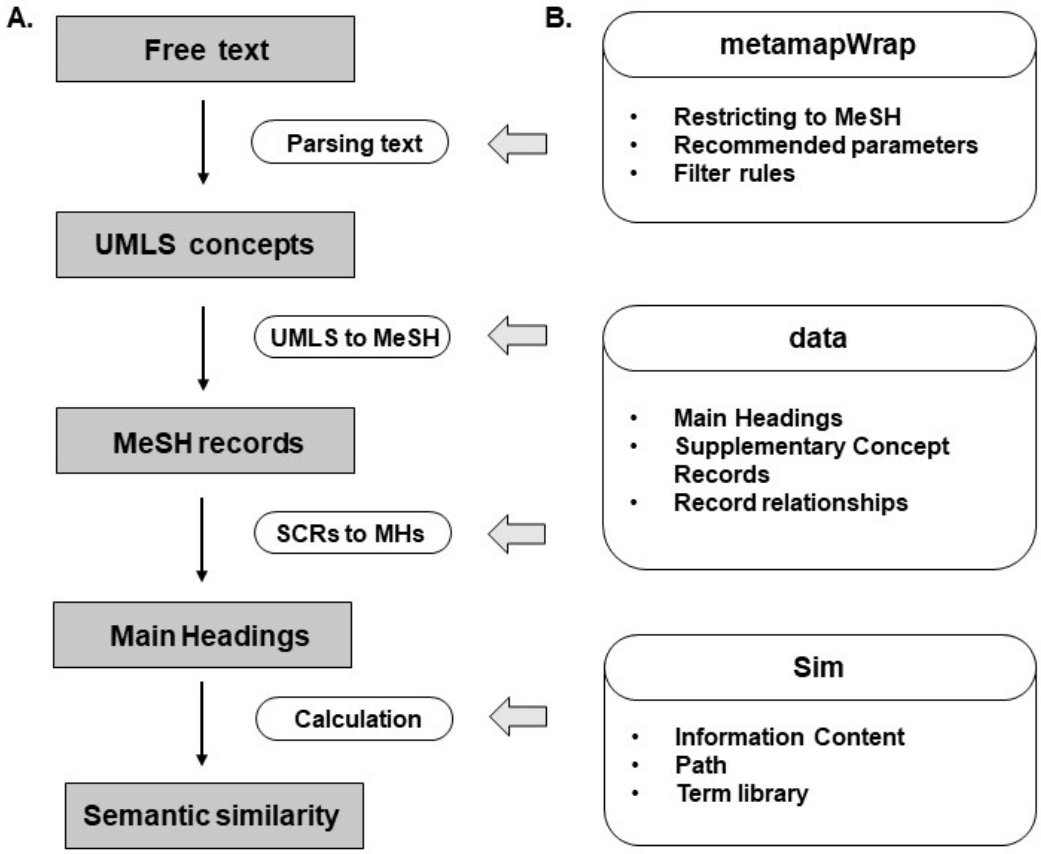

Figure 1. The structure and pipeline of pyMeSHSim. A) The pipeline of comparing two biomedical texts. B) pyMeSHSim is consisted of three subpackages, including metamapWrap, data and Sim. The subpackage metamapWarp curates the MeSH terms from the text. The subpackage data contains selected records and relationships between records in MeSH. The subpackage Sim uses IC-based and graph-based methods to measure MeSH semantic similarity.