Ambiguity and Variability of Database and Software Names in Bioinformatics

Geraint Duck
School of Computer Science
The University of Manchester
Manchester, M13 9PL, UK
duckg@cs.man.ac.uk

David Robertson
Faculty of Life Sciences
The University of Manchester
Manchester, M13 9PL, UK
david.robertson@manchester.ac.uk

Robert Stevens
School of Computer Science
The University of Manchester
Manchester, M13 9PL, UK
r.stevens@manchester.ac.uk

Goran Nenadic
School of Computer Science
The University of Manchester
Manchester, M13 9PL, UK
g.nenadic@manchester.ac.uk

Abstract

There are now numerous options available to achieve various tasks in bioinformatics, but, as yet, little progress has been made to capture the common practice by analysing usage and mentions of databases and tools within the literature. In this paper we analyse the variability and ambiguity of database and software name mentions and provide a set of 30 full-text documents manually annotated on the mention level. Our analyses show that identification of mentions of databases and tools is not a task that can be achieved through dictionary matching alone: our baseline dictionary look-up achieved a F-score of just over 50%. This is primarily because of high variability and ambiguity in database and software mentions contained within the literature and due to the extensive number of new resources introduced. We characterise the issues with various mention types and propose potential ways of capturing additional database and software mentions in the literature.

1 Introduction

Bioinformatics and computational biology widely rely on domain database and software creation to support data collection, aggregation and analysis and, as such, have been reported in research papers, typically as part of the methods section. In addition, many papers introduce new databases and tools. However, little progress has been made to capture the common bioinformatics practice on a large-scale by analysing usage and mentions of databases and tools within the literature[1].

Named entity recognition (NER) has seen wide usage in recent years in identifying mentions of entities of different types in the literature. Within the fields of biology and bioinformatics, these have been used to capture species (Gerner et al., 2010), proteins/genes (Hirschman et al., 2005) and chemicals (Kolluru et al., 2011). NER enables automated literature insight (Zweigenbaum et al., 2007) and provides input to other text-mining applications.

Issues of naming inconsistencies, numerous synonyms and acronyms, and an inability to distinguish entity names from common words in a natural language on top of fuzzy definitions of concepts, make NER an increasingly difficult task (Dingare et al., 2005; Leser and Hakenberg, 2005). Still for some tasks, NER tools achieve relatively high precision and recall scores. For example, LINNAEUS could achieve F-scores around the 95% mark for species name recognition and disambiguation on the mention and document levels (Gerner et al., 2010). On the other hand, gene names, for example, are known for ambiguity and variability, resulting in lower reported F-scores. For example, ABNER recorded

[1] Throughout this paper, we will mention numerous databases and tools by name as examples. A full list of references and web-links to all of these can be found on our website.

an F-score of just under 73% for strict-match gene name recognition and 85% with some boundary error toleration (Settles, 2005).

In this paper we aim to analyse the variability and ambiguity of database and software name mentions in the literature. Similarly to numerous databases with gene names and symbols, there are several inventories that list available database and software resources, including the *NAR* databases and webservices special issues (Brazas et al., 2011; Galperin and Cochrane, 2011), ExPASy[2], the Online Bioinformatics Resources Collection (Chen et al., 2007), etc.

Some work has been done on automated extraction of database and software names from the literature. For example, OReFiL (Yamamoto and Takagi, 2007) utilises URLs to recognise new resources within articles. BIRI (BioInformatics Resource Inventory) uses a series of hand crafted regular expressions to automatically capture resource names, their functionality and classification from paper titles and abstracts (de la Calle et al., 2009). BIRI successfully extracted resource names in 94% of cases in a test corpus, which consisted of 392 abstracts that matched a search for "bioinformatics resource" and 8 documents that were manually included to test domain robustness. However, both of these tools biased their evaluation to resource rich text which prevents full understanding of false negative errors.

This paper aims to analyse database and software name mentions in the literature to evaluate the potential difficulties in automated extraction. We focus on database and software names in the computational biology literature and provide a set of 30 full-text documents manually annotated on the mention level. Although we focus here on bioinformatics resources, we note that the challenges encountered in database and software recognition are a generic issue and not unique to this domain (Kovačević et al., 2012).

2 Methods

For the purpose of this study, we define *databases* as any electronic resource such as database, ontology, repository or classification resource which stores records in a structured form, and provides unique identifiers to each record. Examples include *SCOP* (a database of protein structural classification), *UniProt* (a database of protein sequences and functional information), *Gene Ontology* (ontology that describes gene attributes), etc. We adopt Wikipedia's definition of *software*[3]: "a collection of computer programs ... that provides the instructions for telling a computer what to do and how to do it" (e.g., *BLAST*, for automated sequence comparison). We use *program* and *tool* as synonyms for software, and also include mentions of web-services as well as package names (e.g., *R* packages from *BioConductor*). We explicitly exclude database record numbers/identifiers (e.g. *GO:0002474*, *Q8HWB0*), file formats (e.g. *PDF*), programming languages and their libraries (e.g., *Python*, *BioPython*), operating systems (e.g. *Linux*), algorithms (e.g. *MergeSort*), methods (e.g. *ANOVA*, *Random Forests*) and approaches (e.g. *Machine Learning*, *Dynamic Programming*) from this task. When annotating a mention of database or software in text, associated designators of resources (e.g., *database*, *software*) are included only if part of the official name (e.g., *Gene* ***Ontology***).

We manually annotated a gold standard corpus consisting of 25 full-text articles from BMC Bioinformatics and PLoS Computational Biology articles, and five full-text articles from Genome Biology for database and software name mentions which were randomly selected from the PubMed Central (Roberts, 2001) open-access subset. The annotations were performed by a PhD student with background in bioinformatics and text-mining. Table 1 gives an overview of the annotated corpus. There were 224 lexically unique resources mentioned 1,319 times, with an average of 44 resource mentions per document. The document with the most mentions had 227 resource mentions within it. Finally, the vast majority of mentions are from a small set of names (52% of resource names are only mentioned once).

The inter-annotator agreement (IAA) (Kim and Tsujii, 2006) for annotation of database and software names was calculated from five full-text articles randomly selected from the gold standard corpus, which were annotated by another PhD student

[2] http://expasy.org/

[3] https://en.wikipedia.org/wiki/Software

Total Number of Documents	30
Total Database and Software Mentions	1319
Total Unique Resource Mentions	224
Percentage of Database Mentions	36%
Percentage of Unique DB Mentions	26%
Average Mentions per Document	44.0
Average Unique Mentions per Document	8.2
Max Mentions in a Single Document	227
Max Unique Mentions in a Document	33
Resources with only a Single Mention	117

Table 1: Statistics describing the manually annotated "gold standard" corpus.

with bioinformatics and text-mining background. With lenient agreement (annotation offsets overlap), we calculated a F-score of 86% (93%/80% precision/recall), showing that an adequate level of agreement between two annotators can be achieved despite the potential difficulties of this task. As expected, a decrease in IAA is observed if strict agreement (offsets must exactly match) is used instead (every score drops by 6%).

We have then manually analysed the mentions of database and software names for their length, lexical composition and structural patterns, variability and ambiguity (see Results and Discussion).

To assess the complexity of automated identification of database and software mentions, we used of a baseline text-mining pipeline consisting of a tokeniser, sentence splitter, part-of-speech tagger and gazetteer from GATE's ANNIE (Cunningham et al., 15 April 2011). The gazetteer is based on a dictionary of database and software names, which was compiled from several existing sources (see Table 2). Some well-known acronyms and spelling/orthographic variants have also been added, resulting in 4,871 resources with 5,302 variants (of which, there were 4,879 unique name variants). Dictionary matching was performed by LINNAEUS (Gerner et al., 2010) and standard text-mining performance statistics (precision, recall, F-score) were used for evaluation.

3 Results and Discussion

Database and Software Name Composition. The longest database/software names in the annotated corpus contained five tokens (e.g. *Gene Expression Profile Analysis Suite*, *National Microbial Pathogen Database Resource*). However, there are examples in the dictionary containing more than 10 tokens (e.g. *Prediction of Protein Sorting Signals and Localisation Sites in Amino Acid Sequences*).

As an initial strctural analysis, we collected all the part-of-speech tags assigned to each unique database and software name in our annotated corpus. These were then grouped to profile the structure of resource names (Table 3).

We have identified a total of 228 patterns in the annotated corpus. The majority (82%) of database and software names are comprised of either one, two or three singular proper nouns (NNP). An additional 6% are comprised of a single common noun (NN, e.g. *affy*). A roughly equivalent number contain digits (CD, e.g. *S4*, *t2prhd*). A few contain adjectives (JJ, e.g. *internal transcribed spacer 2*) or prepositions/subordinating conjunctions (IN, e.g. *Structural Classification Of Proteins*). Finally, in three cases (*BLASTed*, *SHAKE*, *dot*), a mention of software was tagged as a verb form (VB and VBP). This is also the reason why there are more patterns (228) than mentions (224). The analysis shows that there is some variety in resource naming and recognition of simple noun phrases alone may not be sufficient.

Variability of Resource Names. We note that the variability of resource names at the dictionary level is 1.09 (5,302 variants over 4,871 resources). For the corpus analysis, we manually grouped names that were referring to the same resource in order to analyse name variability. Of the 224 unique names, 45 were variants of the same tool/database, leaving 179 unique resources. These were either acronyms, misspellings or had alternative orthographics to other mentions. In total, 141 resources had only a single name variant within the corpus (79%). 17% of resources had two variants, and the final 4% had three variants. Of the 45 name variants, 15 were acronyms and all of those were defined in text (and so could be automatically expanded with the right tools (Torii et al., 2007)).

Ambiguity of Resource Names. We note that the ambiguity of resource names at the dictionary level is not high (4,879 unique variants for 4,871 resources). Still, ambiguous resource names exist, e.g.

Type	Entries	URL
DB	196	databases.biomedcentral.com
SW	261	www.bioinformatik.de
PK	597	www.bioconductor.org
SW	1038	www.bioinformatics.ca/links_directory/
SW	365	evolution.genetics.washington.edu/phylip/software.html
DB	140	www.ebi.ac.uk/miriam/main/
DB	1337	www.oxfordjournals.org/nar/database/a/
SW	135	www.netsci.org/Resources/Software/Bioinform/index.html
SW	37	www.bioinf.manchester.ac.uk/recombination/programs.shtml
SW	678	en.wikipedia.org/wiki/Wiki/<various>
–	87	Manually generated entries

Table 2: Database and software URLs from which the database and software name dictionary is comprised. DB = databases; SW = software; PK = packages; data correct as of April 12th, 2011.

Pattern	Count	Freq.
NNP	155	68.0%
NNP NNP	20	8.8%
NN	13	5.7%
NNP NNP NNP	12	5.3%
NNP CD	7	3.1%
NNP CD . CD	4	1.8%
NNP NNP NNP NNP NNP	3	1.3%
NNP LS	2	0.9%
NNP NNP NNP NNP	2	0.9%
Other Patterns	10	4.4%

Table 3: Structure of database and software names. NNP = proper noun, NN = singular noun, CD = cardinal number, LS = list item marker (number).

Network (a tool enabling network inference from various biological datasets) and *analysis* (a package for DNA sequence analysis). We therefore analysed the dictionary of database and software names to evaluate dictionary-level ambiguity when compared to the entries in a full English words dictionary derived from a publicly available list[4] and to a known biomedical acronyms dictionary compiled from ADAM (Zhou et al., 2006), consisting of 86,308 and 1,933 terms, respectively. A total of 37 names matched English words (e.g. *cycle*, *estrogen*, *graph*, *water*) and 43 names fully matched known acronyms (e.g. DIP, *distal interphalangeal* or *Database of Interacting Proteins*). Both comparisons were case-sensitive. The number of matches increase to 405 and 54 respectively when case-insensitive matching is used instead.

To evaluate the recognition-level ambiguity within the annotated corpus, we also compared the tagged database and software names to the English words and acronym dictionary. This resulted in three matches to the English dictionary (*ACT, dot, R*), and one to the acronym dictionary (*IPA*) using case-sensitive matching. This equates to roughly 2% of the annotated names. This increases to 27 matches (12%) if case-sensitive matching is used instead.

Dictionary Matching. Table 4 provides the standard text-mining performance statistics for the dictionary matching approach against the gold standard corpus. The F-scores of under 55% highlight the difficulty of this task, both in terms of matching known ambiguous names (low precision), and from the dictionary not being sufficiently comprehensive (low recall). The most common false positives were *cycle*, *genomes* (potential mentions of BioConductor packages) and *GO* (which was frequently matched within GO database identifiers, e.g., *GO:0007089*). The most common false negatives were *Tabasco*, *MethMarker*, *xPedPhase* and *i Linker*. In each case, the name missed (numerous times) was the resource being introduced in that paper. This shows that any database and software NER must be able to capture newly introduced resources to achieve high recall.

[4] http://wordlist.sourceforge.net/

	TP	FP	FN	P	R	F
Lenient	729	633	590	54%	55%	54%
Strict	695	667	624	51%	53%	52%

Table 4: True positive (TP), false positive (FP), false negative (FN), precision (P), recall (R) and F-score (F) for 30 full-text articles using dictionary look-up.

Type	Contribution
Dictionary matches	55.3%
Heads and Hearst patterns	9.7%
Title appearances	0.6%
References and URLs	1.9%
Version information	1.2%
Noun/Verb associations	20.3%
Comparisons	5.8%
Remaining	5.2%

Table 5: Types of textual patterns and clues for identification of database and software names. Tables 6-11 provide examples of each class.

False negative database and software mentions. We have further analysed the missed database and software names (i.e., the names not in the dictionary) for any common textual clues and patterns. Table 5 summaries different clue categories and their percentage contribution to overall recall. In total, using all clues that we have recognised (see below), final recall could be as high as 95%, though utilising all of these pointers could have a detrimental resulting effect on precision.

The first type of clue that seemed most discriminatory was to associate potential names with *head* terms, i.e. terms that are explicit designators of the type of resource. In the most basic case, a resource name could include a head term or be immediately followed by one (see Table 6). Key head terms included *database*, *software*, *tool*, *program*, *simulator*, *system*, *library* and *service*. Additionally, applying standard Hearst patterns (Hearst, 1992) could be used to extract new and unknown names from enumerations that contain some known database and software names (see Table 6). These patterns could help increase total recall by up to 10%.

We note, however, that not all potential heads are fully discriminatory (for example, *module* in *P and D modules* refer to protein modules (document: PMC1664705), rather than programming ones). Due to the high number of module mentions in that paper, considering *module* as an indicative software head could have a detrimental impact on precision.

the stochastic **simulator** *Dizzy* allows ...
The *MethMarker* **software** was ...
... **tools**: *CLUSTALW*, ..., and *MUSCLE*.
... **programs** such as *Simlink*, ..., and *SimPed*.

Table 6: Example clues and phrases appearing with specific heads or in Hearst patterns. Database and software names are in *italics*, the associated clue is in **bold**.

We further explored a pattern within paper titles where the papers were introducing a new resource (Southan and Cameron, 2009). The title would name the new database or software, and then follow it by a brief description (see Table 7 for examples). Seven of 30 papers in our corpus (over 20%) contained such a pattern. Although this would provide a limited improvement to recall on a mention level ($<$ 1%), it could significantly aid document level recall. In addition, it provides a way to discover new tool names for inclusion in a dictionary with a high discriminatory rate.

Another clue is that database and software mentions are frequently followed by either a reference or a web URL (e.g., "*Galaxy* **[18]** and *EpiGRAPH* **[19]**"). This was the main indicator used by OReFiL (Yamamoto and Takagi, 2007). We recognise, however, that web URLs and citations are not in text only for resources and so this is far less reliable than the previous options (e.g., could incorrectly capture "The *learning metrics principle* **[14, 15]**"). We hypothesise that restricting this type of capture to a paper's *Methods* section may reduce the potential impact on precision.

Numerous database and software mentions also contain or are accompanied by version information (see Table 8). While version numbers can be unambiguous (e.g. having *'v'* or *'version'*), they can also be series of numbers, that are not discriminative enough (e.g. "*AMD Athlon* 1.8 GHz processor" (a CPU), or "sites of *Myc* (0.22) and *NF-kappaB* (0.103)" (genes)).

***CoXpress*:** differential co-expression in gene expression data
***TABASCO*:** A single molecule, base-pair resolved gene expression simulator
***SimHap GUI*:** An intuitive graphical user interface for genetic association analysis

Table 7: Example phrases from Title appearances. Database and software names are in *italics*. Notice that in each case, the name is given as the initial part of the paper's full title (preceding the colon).

using *dot* **v1.10** and *Graphviz* **1.13(v16)**.
CLUSTAL W **version 1.83**
Dynalign **4.5**, and *LocARNA* **0.99**

Table 8: Example versioning clues. Database and software names are in *italics*, the associated clue is in **bold**.

the *SimHap GUI* **installation**.
implemented within *PedPhase*
MethMarker therefore **provides**
A typical **screenshot** of *MethMarker*
Cofolga2 has six free **parameters**
MethMarker's user **interface** reflects
MethMarker can directly **import**
xPedPhase thus needs **cubic time**

Table 9: Example expressions that functionally indicate database and software mentions. Database and software names are in *italics*, the associated clue is in **bold**.

The category with the highest potential contribution (over 20%) includes cases where some expression (could be a noun or a verb) in the sentence (not next to the mention) gives an indication that a database or software is being referred to. Such clues can range from the more discriminatory like *website*, *screenshot* and *download*, to medium ones like *RAM*, *implement*, *simulate* and *running time*, to weak ones such as *run*, *generate*, *evaluate* and *obtain* (see Table 9 for examples). However, this type is also the one with the highest degree of variability as many other "things" can, for example, be *run*, *implemented* or *generated*. Despite some of these being relatively weak, we think that they have limited ambiguity at least within the field of bioinformatics, even if this is not true in a different field. To estimate the effect on precision that inclusion of these clues may have, we compared the number of sentences in the gold standard corpus with a specific clue from this category to the number of sentences with both the clue and a database or software name within our corpus. For example, 77% of sentences contain both a resource name and matched a mention of word *website*, 50% for *RAM* and 48% matched both a name and the regular expression "`ran|run(ning|s)?`". Regardless, there could still be merit in these clues if used in combination with each other rather than alone.

A number of clues can be inferred from sentences that make some comparison between two or more database and software names (see Table 10). Many of these examples can be considered as extended Hearst patterns (e.g., "like tool1, tool2 is ...") but we have analysed them separately for a couple of reasons. In particular, there is an unusually high number of terms contained within this class in the gold standard corpus: a vast majority of the examples within this class (73%) all come from a single paper. Following on from this, neither tool being compared in that paper (most frequently *i Linker* with *xPedPhase*) was present in our dictionary. Thus, even if the comparison pattern has been implemented, the method would need at least to know about some of the tools to infer others. As such, although we envisage potential in addressing this type of database and software mention, we cannot extrapolate how much use it could have due to our biased dataset sample.

Finally, there are a series of mentions (around 5%) without any clear clue, or with particularly ambiguous ones (see Table 11 for examples). Potential clues such as *analyse*, *step* and *minimize* seem too generic within the bioinformatics field to be useful. For example, the number of sentences within our corpus that contained both the regular expression "`analyse(d|s)?|analysis`" and contained a mention of a database or piece of software was only about 20%.

Issues with Scope. There is not always a clear distinction between database and software names, methods, approaches, algorithms, programming lan-

the numbers of breakpoint sites by *xPedPhase* were **equal to** the numbers of breakpoints by *i Linker*.
xPedPhase **did better than** i Linker
Cofogla2 with this cutoff PSVM gives a better false positive rate **compared to** *RNAz*
Foldalign was much **slower than** *Cofolga2* except for
Like *Moleculizer***,** *Tabasco* dynamically generates

Table 10: Examples of comparisons between database and software names. Database and software names are in *italics*, the associated clue is in **bold**.

Additionally, *i Linker* has an error correction step that detects unlikely crossover events.
In addition, *Tabasco* should be a good base to further study interactions on DNA
PSPE is not only able to use one of many common models of nucleotide substitution
The results show that *LibSELDI* tends to have a considerable advantage in the low FDR region
The structure of *Tabasco* confers at least four advantages.

Table 11: Example phrases with no clear or discriminative clues. Database and software names are in *italics*.

guages, database records/identifiers, and file formats. The problem occurs because authors often introduce a novel algorithm and associated implementation (e.g. as a service or a stand-alone application), but frequently refer to their contribution only as an algorithm (or method), rather than software. As such, although they are talking about their algorithm throughout the paper, it could be argued that they are referring to their software implementation, especially when talking about benchmark improvements in results (since the algorithm must have been implemented by this point). The fuzzy boundary between them is going to be a challenge for any focused automated system to overcome.

4 Conclusion

In this paper we present an exploration of variability and ambiguity of database and software mentions in the bioinformatics and computational biology literature. Our results suggest that database and software NER is a non-trivial task that requires more than just a dictionary matching approach. It appears to share many of these difficulties with gene name recognition. Due to bioinformatics' focus on resource creation, a dictionary could never be sufficiently comprehensive, making resource recognition potentially as hard as gene recognition (in contrast to species recognition, which is a relatively stable domain). Example names such as *Network* and *analysis* provide ambiguity and verbalised references to software such as *BLASTed* provide issues of variability that need to be overcome.

Our analyses also provided a series of clues that could be picked up by text-mining techniques, which we are currently in the process of developing. As many of these clues are ambiguous on their own, our approach is to combine various evidence (e.g. using voting and threshold) in order to capture database and software names accurately.

We provide the annotated corpus of 30 full-text articles and manually compiled dictionary at: `http://sourceforge.net/projects/bionerds/`.

Acknowledgments

We would like to thank Daniel Jamieson for his help in establishing the inter-annotator agreement. GD is funded by a studentship from the Biotechnology and Biological Sciences Research Council (BBSRC) to RS, GN and DLR. We also thank authors of sites listed in Table 2 for freely providing inventories of database and tool names.

References

Michelle D Brazas, David S Yim, Joseph T Yamada, and B F Francis Ouellette. 2011. The 2011 bioinformatics links directory update: more resources, tools and databases and features to empower the bioinformatics community. *Nucleic acids research*, 39 Suppl 2(suppl_2):W3–7.

Yi-Bu Chen, Ansuman Chattopadhyay, Phillip Bergen, Cynthia Gadd, and Nancy Tannery. 2007. The Online Bioinformatics Resources Collection at the University of Pittsburgh Health Sciences Library System–a one-stop gateway to online bioinformatics databases and software tools. *Nucleic acids research*, 35(Database issue):D780–5.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, Wim Peters, et al. 15 April 2011. *Text Processing with GATE (Version 6)*. University of Sheffield Department of Computer Science.

Guillermo de la Calle, Miguel García-Remesal, Stefano Chiesa, Diana de la Iglesia, and Victor Maojo. 2009. BIRI: a new approach for automatically discovering and indexing available public bioinformatics resources from the literature. *BMC bioinformatics*, 10(1):320.

Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, and Claire Grover. 2005. A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations. *Comparative and functional genomics*, 6(1-2):77–85.

M. Y. Galperin and G. R. Cochrane. 2011. The 2011 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. *Nucleic acids research*, 39(Database issue):D1–D6.

Martin Gerner, Goran Nenadic, and Casey M Bergman. 2010. LINNAEUS: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(1):85.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, pages 539–545, Morristown, NJ, USA. Association for Computational Linguistics.

Lynette Hirschman, Alexander Yeh, Christian Blaschke, and Alfonso Valencia. 2005. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC bioinformatics*, 6 Suppl 1(Suppl 1):S1.

Jin-Dong Kim and Jun'ichi Tsujii. 2006. Corpora and Their Annotation. In Sophia Ananiadou and John McNaught, editors, *Text Mining for Biology and Biomedicine*, chapter 8, pages 179–211. Boston and London: Artech House.

BalaKrishna Kolluru, Lezan Hawizy, Peter Murray-Rust, Junichi Tsujii, and Sophia Ananiadou. 2011. Using workflows to explore and optimise named entity recognition for chemistry. *PLoS ONE*, 6(5):e20181, 05.

Aleksandar Kovačević, Zora Konjović, Branko Milosavljević, and Goran Nenadic. 2012. Mining methodologies from NLP publications: A case study in automatic terminology recognition. *Computer Speech & Language*, 26(2):105–126.

Ulf Leser and Jörg Hakenberg. 2005. What makes a gene name? Named entity recognition in the biomedical literature. *Briefings in Bioinformatics*, 6(4):357–369.

Richard J. Roberts. 2001. PubMed Central: The GenBank of the published literature. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):381–2.

Burr Settles. 2005. ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–2.

Christopher Southan and Graham Cameron. 2009. Database Provider Survey. Technical report, ELIXIR, EMBL-EBI.

Manabu Torii, Zhang-zhi Hu, Min Song, Cathy H Wu, and Hongfang Liu. 2007. A comparison study on algorithms of detecting long forms for short forms in biomedical text. *BMC bioinformatics*, 8 Suppl 9(Suppl 9):S5.

Yasunori Yamamoto and Toshihisa Takagi. 2007. OReFiL: an online resource finder for life sciences. *BMC bioinformatics*, 8(1):287.

Wei Zhou, Vetle I Torvik, and Neil R Smalheiser. 2006. ADAM: another database of abbreviations in MEDLINE. *Bioinformatics*, 22(22):2813–8.

Pierre Zweigenbaum, Dina Demner-Fushman, Hong Yu, and Kevin B Cohen. 2007. Frontiers of biomedical text mining: current progress. *Briefings in bioinformatics*, 8(5):358–75.