TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Bioinformatik

Augmenting Humans.
A Text Mining Approach

Juan Miguel Cejuela Pérez

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Stephan Günnemann

Prüfer der Dissertation:

1. Prof. Dr. Burkhard Rost
2. Prof. Yana Bromberg Ph.D., The State University of New Jersey/USA

Die Dissertation wurde am 27.04.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 02.11.2017 angenommen.

Abstract

Humanity is at a turning point. Accelerated advances in artificial intelligence bring us great benefits but also pose great challenges. One is: are we going to keep our jobs? Factory robots, news written by software, patient treatments decided by machines. On a positive note, in this work we study ways to augment, not substitute, the labor of humans. We observe it from the lenses of *text mining*. We apply it to make sense of the deluge of text data in the fields of genomics. In particular, we look at three peer-reviewed cases that combined the automation of text mining methods with the feedback of experts, ultimately to support database curators in their work.

Firstly, we developed a web-based interface that allows experts to validate and improve the automatic annotations (e.g. gene functions) of a text mining system. We showed that this semi-automatic annotation approach was up to 2-fold faster than manual curation. We demonstrated that the system can assist the curation of biomedical databases in a real setting: multiple employees at FlyBase, the premier repository of the model organism Drosophila melanogaster (a fly species), used the interactive interface to annotate hundreds of full-text scientific publications in a cost-effective manner.

Secondly, we developed a new method to extract from the literature mentions of genetic variations. Our method superseded the results of previous ones, uniquely found 33% of all mentions, and was the only one to discover genetic variations written in natural language. Previous methods primarily only treated simple mentions (e.g. "E6V"), whereas our method was optimized to also understand complex natural language (e.g. "glutamic acid was substituted by valine at residue 6"). This was made possible thanks to the iterative and selective re-training of the automatic system, which was guided by users.

Lastly, we developed a text mining method to extract the native subcellular localization of proteins. Compared to previous solutions, the new method boasted very high accuracy (New=86% vs. Old=51%). We applied the system to mine the latest research; we verified that 65%-85% of the text-mined protein localization annotations were correct and novel (i.e. not known before). Assisted by this method, non-experts (we) were able to discover >100 novel annotations per work day.

List of Publications

The work at hand constitutes a cumulative dissertation based on peer-reviewed publications. Chapters 2, 3, and 4 describe methodologies and results published in (the articles are included in this dissertation):

- **Juan Miguel Cejuela**, Peter McQuilton, Laura Ponting, Steven J. Marygold, Raymund Stefancsik, Gillian H. Millburn, Burkhard Rost, and the FlyBase Consortium. *tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles*. Database (Oxford) 2014;2014(0):bau033.

- **Juan Miguel Cejuela**, Aleksandar Bojchevski, Carsten Uhlig, Rustem Bekmukhametov, Sanjeev Kumar Karn, Shpend Mahmuti, Ashish Baghudana, Ankit Dubey, Venkata P. Satagopam, & Burkhard Rost. *nala: text mining natural language mutation mentions*. Bionformatics 2017.

- **Juan Miguel Cejuela**, Shrikant Vinchurkar, Tatyana Goldberg, Madhukar Sollepura Prabhu Shankar, Ashish Baghudana, Aleksander Bojchevski, Carsten Uhlig, André Ofner, Pandu Raharja-Liu, Lars Juhl Jensen and Burkhard Rost. *LocText: relation extraction of protein localizations to assist database curation*. BMC Bioinformatics 2018; 19.

During the dissertation work, I co-authored several other peer-reviewed publications:

- T. Goldberg, S. Vinchurkar, **J. M. Cejuela**, L. J. Jensen, and B Rost. *Linked annotations: a middle ground for manual curation of biomedical databases and text corpora.* BMC Proceedings 2015, 9(Suppl 5): A4.

- Pyysalo, S., J. Campos, **J. M. Cejuela**, F. Ginter, K. Hakala, C. Li, P. Stenetorp and L. J. Jensen (2015). *Sharing annotations better: RESTful Open Annotation*. In: Proceedings of ACL-IJCNLP 2015 System Demonstrations. Association for Computational (ACL) 91-96.

- H. V. Cook, R. Bērziņš, C. L. Rodríguez, **J. M. Cejuela**, and L. J. Jensen. *Creation and evaluation of a dictionary-based tagger for virus species and proteins*. ACL 2017 BioNLP Workshop, submitted.

- Arighi, C. N., B. Carterette, K. B. Cohen, M. Krallinger, W. J. Wilbur, P. Fey, R. Dodson, L. Cooper, C. E. Van Slyke, W. Dahdul, P. Mabee, D. Li, B. Harris, M. Gillespie, S. Jimenez, P. Roberts, L. Matthews, K. Becker, H. Drabkin, S. Bello, L. Licata, A. Chatr-aryamontri, M. L. Schaeffer, J. Park, M. Haendel, K. Van Auken, Y. Li, J. Chan, H. M. Muller, H. Cui, J. P. Balhoff, J. Chi-Yang Wu, Z. Lu, C. H. Wei, C. O. Tudor, K. Raja, S. Subramani, J. Natarajan, **J. M. Cejuela**, P. Dubey and C. Wu (2013). *An overview of the BioCreative 2012 Workshop Track III: interactive text mining task*. Database (Oxford) 2013: bas056.

- Jiang, Y., T. R. Oron, W. T. Clark, A. R. Bankapur, D. D'Andrea, R. Lepore, C. S. Funk, I. Kahanda, K. M. Verspoor, A. Ben-Hur, C. E. Koo da, D. Penfold-Brown, D. Shasha, N. Youngs, R. Bonneau, A. Lin, S. M. Sahraeian, P. L. Martelli, G. Profiti, R. Casadio, R. Cao, Z. Zhong, J. Cheng, A. Altenhoff, N. Skunca, C. Dessimoz, T. Dogan, K. Hakala, S. Kaewphan, F. Mehryary, T. Salakoski, F. Ginter, H. Fang, B. Smithers, M. Oates, J. Gough, P. Toronen, P. Koskinen, L. Holm, C. T. Chen, W. L. Hsu, K. Bryson, D. Cozzetto, F. Minneci, D. T. Jones, S. Chapman, D. Bkc, I. K. Khan, D. Kihara, D. Ofer, N. Rappoport, A. Stern, E. Cibrian-Uhalte, P. Denny, R. E. Foulger, R. Hieta, D. Legge, R. C. Lovering, M. Magrane, A. N. Melidoni, P. Mutowo-Meullenet, K. Pichler, A. Shypitsyna, B. Li, P. Zakeri, S. ElShal, L. C. Tranchevent, S. Das, N. L. Dawson, D. Lee, J. G. Lees, I. Sillitoe, P. Bhat, T. Nepusz, A. E. Romero, R. Sasidharan, H. Yang, A. Paccanaro, J. Gillis, A. E. Sedeno-Cortes, P. Pavlidis, S. Feng, **J. M. Cejuela**, T. Goldberg, T. Hamp, L. Richter, A. Salamov, T. Gabaldon, M. Marcet-Houben, F. Supek, Q. Gong, W. Ning, Y. Zhou, W. Tian, M. Falda, P. Fontana, E. Lavezzo, S. Toppo, C. Ferrari, M. Giollo, D. Piovesan, S. C. Tosatto, A. Del Pozo, J. M. Fernandez, P. Maietta, A. Valencia, M. L. Tress, A. Benso, S. Di Carlo, G. Politano, A. Savino, H. U. Rehman, M. Re, M. Mesiti, G. Valentini, J. W. Bargsten, A. D. van Dijk, B. Gemovic, S. Glisic, V. Perovic, V. Veljkovic, N. Veljkovic, E. S. D. C. Almeida, R. Z. Vencio, M. Sharan, J. Vogel, L. Kansakar, S. Zhang, S. Vucetic, Z. Wang, M. J. Sternberg, M. N. Wass, R. P. Huntley, M. J. Martin, C. O'Donovan, P. N. Robinson, Y. Moreau, A. Tramontano, P. C. Babbitt, S. E. Brenner, M. Linial, C. A. Orengo, B. Rost, C. S. Greene, S. D. Mooney, I. Friedberg and P. Radivojac. *An expanded evaluation of protein function prediction methods shows an improvement in accuracy*. Genome Biol. 2016 17(1): 184.

Acknowledgments

First, and foremost, I want to thank my *Doktorvater*: Prof. Burkhard Rost. You always trusted me. Early on, you saw and then encouraged my capabilities. You impacted me manyfold: you are a free thinker, humble, cheerful, exacting, sharp. Also, sometimes a little bit difficult. Our quick and witty conversations in your office and during walks sharpened my intelligence.

I want to thank all the very talented and friendly people at the Rostlab group. I start from those who always guarantee that everything runs smoothly. Timothy Karl: you assisted me in all ways possible for hardware and software installations. Of most importance, you are a good person. Inga Weise and Marlena Drabik: thank you for your more than excellent administrative support, and so much love and care. You had to cope, and succeeded, with my sloppiness in bureaucratic affairs. On the same note, Manuela Fischer, Ute Stinzel, and Tamara Schyrzisko helped me greatly with the submission procedures of this dissertation.

The great researchers and scientific visitors at Rostlab taught me so much. My dear officemate Tatyana Goldberg: you showed me a level of drive and dedication that is contagious and seen only in a handful of people. I had the pleasure to participate in or merely observe discussions of extremely clever people: Edda Kloppmann, Maximilian Hecht, Dmitrii Nechaev, Jonas Reeb, Thomas Hopf, Marco Punta, and Marc Offman. My then master's thesis supervisor Andrea Schafferhans also accompanied me with the PhD in the Rostlab journey; her passion for teaching is inspiring. Great colleagues were: Yannick Mahlich, Christian Schaefer, Frank Wallrapp, David Dao, Tobias Hamp, Venkata P. Satagopam, Sebastian Wilzbach, Michael Bernhofer, Maximilian Miller, Jiajun Qiu, Shaila C. Roessle, Valérie Marot. Guy Yachdav guided the first steps of my research and gave me great feedback; especially career-wise. I am especially thankful to Lothar Richter: you also had to cope sometimes with the worst sides of me, and eventually became a great partner. You do a superb and much needed work at Rostlab. I thank Esteban Peguero Sánchez, who instantly became a friend and on top was so thorough and determined in his science. Prof. Yana Bromberg is a mine of pure gold. She is sharp; she is rebellious; she is marvelous. The always smiling Christian Dallago: you are a rock star, always helpful, always overproviding, and exceeding the expectations. My collaborator Prof. Lars Juhl Jensen: you are, simply put, a machine. You opened me to a whole other level of rigor and skill. My collaborator Jin-Dong Kim: you brought me many new adventures, including the co-organization of

the BLAHmuc hackathon in Munich with international visitors, and the work-intensive and also playful weeks in Tokyo and other colorful towns of Japan. Very important, thank you Michael Heinzinger and Tanzeem Haque: who quickly jumped to help me out with German translations!

My work was truly not possible without my students, and ultimately, I had the most fun working with them. I am so lucky for having met such a line-up of absolute legends: Shrikant Vinchurkar, Rustem Bekmukhametov, Carsten Uhlig, Pandu Raharja-Liu, André Ofner. Other students impacted me severely: Ashish Burkhard, Kujtim Rahmani, Madhukar Sollepura Prabhu Shankar, Ankit Dubey, Sanjeev, Shpend Mahmuti, Vasileios Magioglou. I am especially thankful to Aleksander Bojchevski: you worked with me in so many projects and put me on pressure to become as good as you are. Also, I must thank the many students of the datamining labs and bioinformatics practical courses: you were so good and brought me many playful insights.

To all those who share open source and curate public databases: your immense and selfless efforts make my work possible. Thank you for all the great communities: GitHub, UniProtKB, PubMed, Stack Overflow, Wikipedia, *Die Stabi in München*...

I cannot forget my friends. My friend and partner Jorge Campos: you know well our countless nights working around the clock. Thank you for your patience, support, and hard work. Roc Reguant, *el j**** catalán: your pesky reminders and much accountability kept me motivated. Michael Eigster and Atanas Dimitrov: thank you for your support in tha' hustle; keep it real gangsters. Annalisa Tonni: you are simply, cute and lovely. Katharina Popp: you were a dear companion in my life; thank you so much for your love. Huang Xiao: thank you, thank you, thank you my dear Chinese friend. Habtom Kahsay: your unorthodox thinking made me think twice and sometimes challenged my perceptions. In the same vein, I must add Dmitrii Nechaev: you weird nerd. Also, thank you for the proofreading! Pandu Raharja-Liu: you are awesome; but not as much as me. Sebastian Wilzbach: :*. Dorothea Haider: thank you so much for your tireless feedback. Takuya Kajiwara: thank you for the good times in Tokyo! Srushty Chafekar: you, girl. Other friends impacted me indirectly: Klaus Schu, Jan Smarschevski**.** You will soon be my friend: Gary Vaynerchuk; you infected my brain me vastly. Thank you Tim Ferriss. To Alejandro Gata: you touched the fabric of my reality. All the great people at Toastmasters Prostmasters: Thomas Dall, Ineke Vermeulen, Christian Sammut, Mel Kelly, Ranjith Venkatesh, Stefan Gruß, Christopher Magyar, ... Ultimately, I have an infinite list of people to thank to; including all the good and "bad" encounters. My *ænema*, my essence, is defined by *you*.

Finally, my father: *eres la persona que más quiero en este mundo. Me has enseñado a ser una buena persona y lloro sólo de recorder tu amor. Gracias a ti me he podido permitir tantas cosas y en esencia te lo debo todo*.

♥

Table of Contents

List of Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Problem overview: massive data, little knowledge

Everything is accelerating. We carry super computers in our pocket. We fuse into our bodies smart sensors, robotic limbs, 3D-printed organs. Online transactions are global; every click, every search, every rating is stored. YouTube videos, Instragram photos, tweets. The problem with Big Data is not that is *big*. The problem is that is *insatiably growing*. Exponentially (Hilbert and Lopez 2011).

Genomics, once on the information technology wagon, shows the same explosive growth. DNA sequencing does not follow Moore's law; it surpasses it doubling every seven months (Fig. 1.1) (Stephens et al. 2015). Sequence data increased as sequencing costs shrunk. The costs reduction also surpassed Moore's law (Fig. 1.2), from the estimated €2.7 billion spent in the human genome project, that took a decade, to the now a reality $1,000 genome, that takes hours (Hayden 2014; NHGRI 2010).

The problem: we still do not grasp even a tiny fraction of this humongous data. The standard database for protein sequences and functional annotations, UniProtKB (The UniProt Consortium 2017), lists as of time of writing over 80 million proteins. The existence of the majority of proteins was only predicted (74%) or inferred from homology (24%). Less than 1.7% of the proteins were evidenced experimentally (protein or transcript level evidence). Moreover, nearly all proteins only have predicted functional annotations (99.3%; UniProtKB/TrEMBL (Bairoch and Apweiler 1999)) as compared to experimentally-based or manually-verified annotations (0.7%; UniProtKB/Swiss-Prot (Boutet et al. 2016)).

The performance of automatic prediction methods remains largely insufficient. Several international experiments continue to assess methods for the prediction on, e.g. protein structure (CASP (Kryshtafovych et al. 2014)), protein-protein interaction (CAPRI (Lensink and Wodak 2013)), or protein function (CAFA (Jiang et al. 2016)). These methods primarily machine-learn patterns in the protein sequences or structures or infer functional aspects by homology, i.e. transfer the annotations and functionality of better-known, similar proteins.

The text of the literature is another source of data to machine learn, i.e. to *text mine*. In

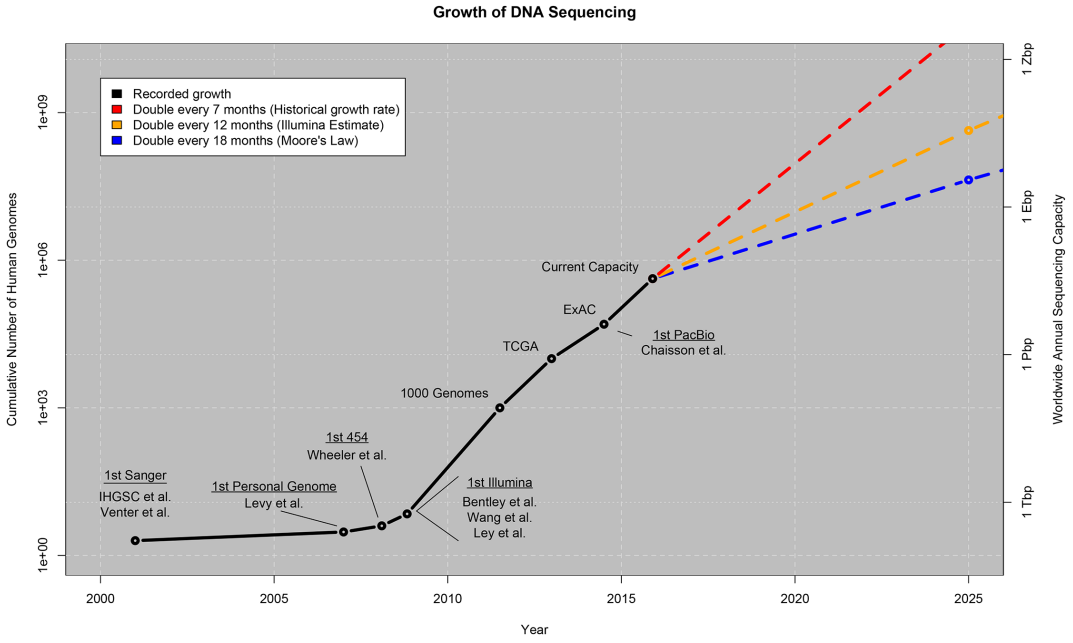


Fig. 1.1. Explosive growth in genomic data. Logarithmic scale of growth in cumulative number of sequenced human genomes (left axis) and in worldwide annual sequencing capacity measured in DNA basepairs (right axis: Tera-basepairs (Tbp), Peta-basepairs (Pbp), Exa-basepairs (Ebp), Zetta-basepairs (Zbps)). Selected milestones are shown, from the first reported human genomes (2001) (IHGSC 2001; Lander et al. 2001), to the first next-generation sequencing technology (2008) (D. R. Bentley et al. 2008), to the Exome Aggregation Consortium (ExAC), that collects over 60,000 human exomes (~2016) (Lek et al. 2016). The historical growth was recorded until 2015 (black). The growth for the following decade is projected, considering three estimators: historical growth rate (red), estimate of Illumina, next-generation sequencing company, (orange), and Moore's law (blue). *Source:* (Stephens et al. 2015).

parallel to genomic data, the body of knowledge in biomedical literature is massive. The go-to place for biology and medicine is PubMed, the search engine maintained by the United States National Library of Medicine. PubMed currently registers 27 million publications, has a growth rate of ~4.5% and now grows with over 1.1 million new articles every year (Fig. 1.3).

By text mining the biomedical literature, we mean extracting from publications, results and descriptions that can help us understand different aspects of molecular knowledge (examples, Table 1.1). One would think a priori that the deposited knowledge in scientific journals was well referenced and mapped (i.e. hyperlinked) to standard ontologies such as UniProtKB or the Gene Ontology (GO) (Ashburner et al. 2000). Sadly, despite past efforts, this is not the case. For example, what does "GC1" (in Table 1.1) mean? Is it a protein, a gene? Is it the protein "Mitochondrial glutamate carrier 1" or the different protein (that shares the same abbreviation) "Epimerase family protein SDR39U1 homolog, chloroplastic" ? Is it a protein from human, mouse, another species? what is the actual meaning of "inhibit cell motility"? We will review these challenges in the next section.

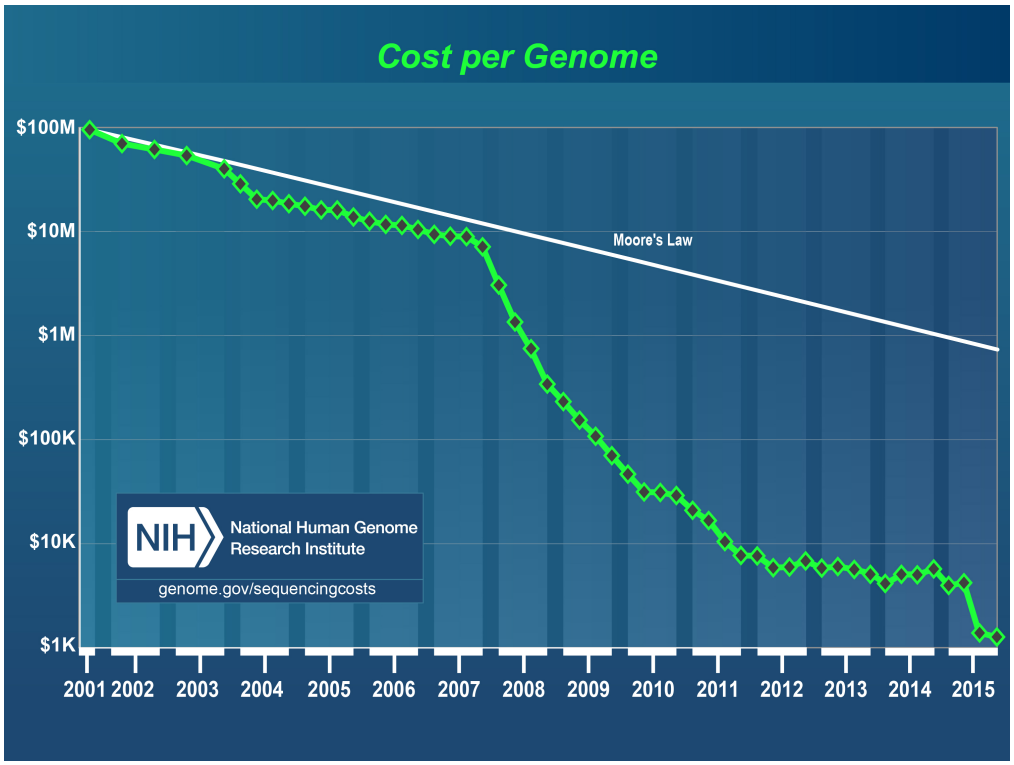


Fig. 1.2. Sequencing costs reduction, faster than Moore's law. Logarithmic scale of the yearly progression of costs in dollars to sequence a single entire human genome. The drastic drop in sequencing costs sparked around 2008, due to emergence of the first next-generation sequencing technologies (D. R. Bentley et al. 2008). *Source:* (Wetterstrand 2017).

Table 1.1. Examples of molecular knowledge deposited in the literature.

Evidence for	Literature Passage
molecular function, biological process	*"**PKA** and **CDK5** can **phosphorylate** specific serines on the intracellular domain of podoplanin (PDPN) to **inhibit cell motility**", PMID 25959509*
genetic mutations, linked diseases	*"**Mis-sense mutation Val----Ile in exon 17** of **amyloid precursor protein** gene in Japanese familial **Alzheimer's disease**", PMID 1678058*
protein cell localization	*"the C-terminal domains of **AtCASP** and **GC1** to **GC6** localized to the **Golgi**", PMID 18182439*

In the context of making sense of the literature, one job is essential: that of the *biocurator* (Salimi and Vita 2006; Burge et al. 2012; Bateman 2010). Biocurators are professional scientists who collect, validate, and maintain biological research information and deposit it, in machine-readable form, into specialized biomedical databases (i.e. *biodatabases*). Examples of databases that employ manual curation labor are the already mentioned UniPro-

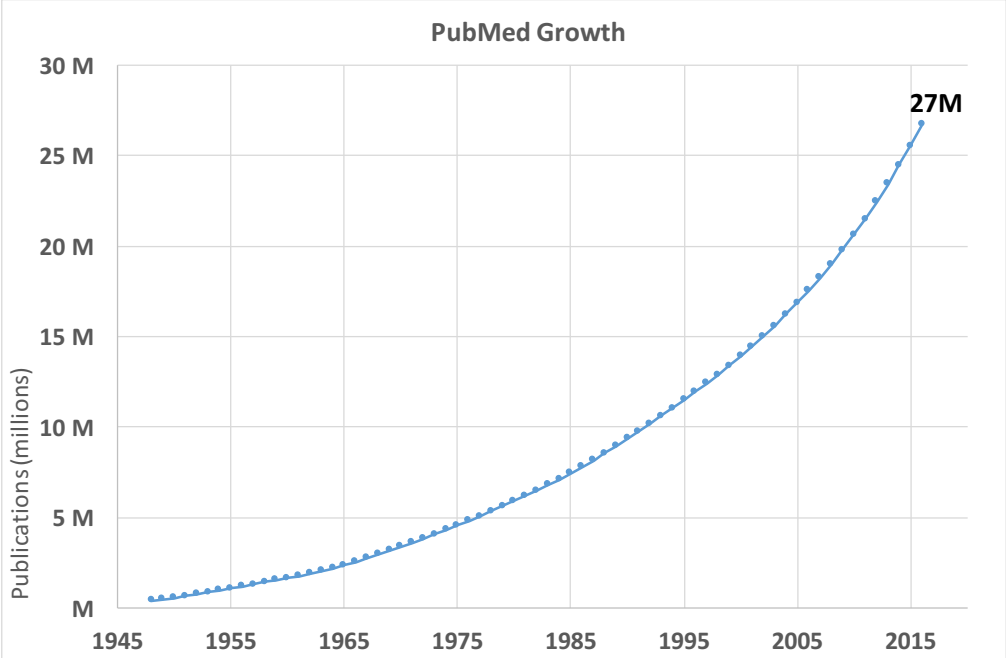


Fig. 1.3. PubMed: rapid growth of the biomedical literature. The number of biomedical publications experiences a growth rate of ~4.5%. Over 1.1 million new publications are now deposited every year in PubMed.

tKB (The UniProt Consortium 2017), and more specifically its manually-annotated section *Swiss-Prot* (Boutet et al. 2016), or the many databases of model organisms, e.g. *FlyBase* for the organism *Drosophila melanogaster* (a fly species) (Gramates et al. 2017), *SGD* for *Saccharomyces cerevisiae* (baker's yeast) (Cherry et al. 2012), or *MGI* for *Mus musculus* (mouse) (Blake et al. 2017). By large, the primary source of biological research stems from the literature. Because of this, biocurators must constantly scan and read newly published articles to spot and organize the latest scientific findings. The particular tasks of a biocurator vary and depend on the target database, however, simplified, three are the main activities: (1) filter documents relevant for curation (a process often called *triage*), e.g. to select only those articles treating a specific organism or a particular disease; (2) identify in the text the discussed biological entities and processes that are of interest, e.g. a newly discovered function of a gene; and (3) convert the information in a way that is unambiguous and machine-readable for final database entry. Still largely, these tasks are done manually, that is, without automation and not at scale (human readers cannot cope with the millions of new articles incessantly being published). In this context, literature-based text mining methods may assist biocurators by suggesting and pre-filling data that later they can confirm or reject. This quality assurance step of the automatic annotations (*human-in-the-loop*) is still essential, for most biodatabases demand a very high level of accuracy in the annotations, and the performance of automatic methods does not match yet. Now, we review the challenges in text mining specifically encountered in the biomedical domain.

1.2 BioNLP: concepts and challenges

The final goal of having all the biomedical literature perfectly semantically indexed is far from being realized. *Natural language processing* (NLP) automatic techniques aim to solve this problem. NLP can sometimes be seen as a methodology or part of the global text mining field. In this work we make no distinction. Also, in recent years there has been more research emphasis on, in contrast to *processing*, the aspect of natural language *understanding* (NLU). In this work we see them as equivalent. *Biomedical text mining* is also often referred as *BioNLP*.

Regardless of terminology, the challenge is always: how to use very large amounts of unstructured text that is understandable by humans, i.e. *natural language,* and turn into *unambiguous* (structured) useful knowledge? What useful knowledge is depends on the application case. One may ask simply how often Michael Jordan is talked about in the news, now or through the years. This involves recognizing the discussed concepts (e.g. "Michael Jordan"), a process called *named-entity recognition* (NER). One may ask more difficult questions, such as what is the network of Jordan's personal connections or the list of books he wrote. This involves relating named entities, a process called *relationship extraction* (RE). Crucial is gathering information that is *unambiguous*, that is, e.g. knowing whether "Jordan" contextually meant Michael Jordan the player and businessman, or Michael I. Jordan the machine learning researcher, or Air Jordan the shoes brand. Identifying entities unambiguously is equivalent to linking (mapping) them to a unique identifier in an external recognized resource (e.g. Wikipedia URLs for personalities, passport ids for citizens, or UniProtKB identifiers for proteins), a process called *named-entity normalization* (NEN), also often called *named-entity linking* (NEL), or *named-entity disambiguation* (NED).

Ambiguity is the pervading characteristic of natural language. It is most exacerbated in the biomedical domain. Tens of different concepts exist: proteins, genes, mutations, function, phenotype, diseases, symptoms, medical procedures, chemical reactions, drugs, organisms, etc. Worst: for many concepts the terminology of names is not standard. Proteins, the machinery of life, specially suffer from name ambiguity. For a start, e.g. with the wording "p53", is not clear whether this refers to a protein name, or its encoding gene, or its mRNA, if not something else completely different (Hatzivassiloglou, Duboue, and Rzhetsky 2001). The distinction between proteins, genes, or mRNA is difficult (even for human readers) and is often ignored; many groups of researchers encompass all three into the single concept of *gene or gene product* (GGP). Then, many other ambiguities lurk:

Synonymy: proteins and most biological entities as referenced by various different names and all must be contemplated, for example for search (Fig. 1.4).

New names constantly coined: new names are created as UniProtKB and other biomedical databases expand. This largely complicates the curation of comprehensive dictionaries (of names) and demands constant updates from tools.

Abbreviations and short symbols: many entities can share a same abbreviation, e.g. "ACE" may stand for at least 20 different expansions, including "angiotensin converting enzyme", "affinity capillary electrophoresis", or "acetylcholinesterase" (Leser and Hakenberg 2005; Adar 2004). (J. T. Chang, Schutze, and Altman 2002) reported that abbreviations with six characters or less collapsed in average with 4.61 different definitions.

Clash with common words: English words may name proteins too, e.g. "white", "And", "cactus", "eagle", "zen", "Pavarotti", "Pokemon" (later retracted for copyright infringement (Simonite 2005)), or even single letters like "A" or "C" may mean names.

Homology-shared names: e.g. from the wording alone "ATM" ("Serine-protein kinase ATM"), it is unknown whether it refers to the protein in human (UniProtKB: ATM_HUMAN), in mouse (UniProtKB: ATM_MOUSE), or another organism.

A myriad of ontologies and databases exist. These are usable as references for biomedical entity disambiguation. Some of the most important are listed in Table 1.2. Most BioNLP applications need to navigate several of these vocabularies to provide useful results, adding to the complexity.

Names & Taxonomy

Protein names	*Recommended name:* **Multifunctional 2-oxoglutarate metabolism enzyme** *Alternative name(s):* • 2-hydroxy-3-oxoadipate synthase (EC:2.2.1.5) ▪ *Short name:* HOA synthase ▪ *Short name:* HOAS • 2-oxoglutarate carboxy-lyase • 2-oxoglutarate decarboxylase • Alpha-ketoglutarate decarboxylase (EC:4.1.1.71) ▪ *Short name:* KG decarboxylase ▪ *Short name:* KGD • Alpha-ketoglutarate-glyoxylate carboligase Including the following 2 domains: • 2-oxoglutarate dehydrogenase E1 component (EC:1.2.4.2) ▪ *Short name:* ODH E1 component *Alternative name(s):* ▪ Alpha-ketoglutarate dehydrogenase E1 component ▪ *Short name:* KDH E1 component • Dihydrolipoyllysine-residue succinyltransferase component of 2-oxoglutarate dehydrogenase complex (EC:2.3.1.61) *Alternative name(s):* ▪ 2-oxoglutarate dehydrogenase complex E2 component ▪ *Short name:* ODH E2 component ▪ *Short name:* OGDC-E2 ▪ Dihydrolipoamide succinyltransferase
Gene names	*Name:***kgd** Synonyms:sucA Ordered Locus Names:MSMEG_5049, MSMEI_4922
Organism	Mycobacterium smegmatis (strain ATCC 700084 / mc(2)155)
Taxonomic identifier	246196 [NCBI]

Fig. 1.4. When names are ambiguous. Example of a UniProtKB protein entry (KGD_MYCS2) with several recommended, alternative, short, or gene names. All names must be contemplated for an efficient search of concepts. Short abbreviations such as "KGD" may be shared by many other different proteins. *Source:* UniProtKB.

Table 1.2. Many biomedical ontologies exist; examples.

Biomedical entity	Ontologies
proteins (or genes)	*UniProtKB* (The UniProt Consortium 2017)
genes (or proteins)	*Entrez Gene* (Sayers et al. 2010; Maglott et al. 2011)
organisms	*NCBI Taxonomy* (Sayers et al. 2010)
protein function, biological processes, and protein subcellular localization	*Gene Ontology (GO)* (Ashburner et al. 2000)
mutation mentions	the *HGVS* nomenclature (Dunnen et al. 2016)
human diseases, or other medical, or pharmaceutical terms	*Disease Ontology (DO)* (Kibbe et al. 2015), *SNOMED CT* (Shahpori and Doig 2010), *UMLS* (Bodenreider 2004), *MeSH* (Nelson n.d.; Major, Kostrewski, and Anderson 1978), *ICD-10* (*ICD-11* version is in development) (WHO 1992; First et al. 2015)
drugs	*RxNorm* (S. Liu et al. 2005) or *DrOn* (Hanna et al. 2013)
chemicals	*ChEBI* (Degtyarenko et al. 2008)
phenotypes	*HPO* (Kohler et al. 2017)
many, many more ...	see *BioPortal* from the *NCBO* (National Center for Biomedical Ontology) (Whetzel et al. 2011; Musen et al. 2012) and *OBO* (Open Biomedical Ontologies) (B. Smith et al. 2007).

A limitation faced by biomedical text mining systems is that a great part of the literature is closed behind walls. PubMed lists biomedical articles but only abstracts (i.e. including titles) are available. Many research results are contained only within the full text of an article (e.g. in the Conclusions section) (J. Lin 2009). *PubMed Central (PMC)*, the subset of PubMed with freely available articles full texts, contains as of today 4.2 million articles, a small number compared to the over 27 million articles in PubMed. In other words, only ~16% of PubMed articles have free full texts. Worse, only a fraction of PubMed Central articles (as of today, 1.5 million) is in the denominated "Open Access Subset", that grants text mining tools free use. That is, only ~6% of PubMed articles are freely and fully available for text mining. It was not until recently (Van Noorden 2014), that the publisher Elsevier opened, although with commercial restrictions, its more than 11 million research papers to text mining. Even then, most studies even nowadays have been limited to abstracts only. A recent search on PubMed revealed that only ~3% of text-mining-related studies mention full text (searches: `http://bit.ly/2nbGyVG` and `http://bit.ly/2oDL7aZ`). Further, the efforts put into development of methodologies and research of systems trained on ab-

stracts only, may be futile for full texts (Cohen et al. 2010; Antonio Jimeno Y. and Karin Verspoor 2014; A. Jimeno Y. and K. Verspoor 2014).

Besides PubMed and PMC, other biomedical sources relevant for text mining exist. The most important are: *Europe PMC* (Europe 2015) is a newer search index that includes PubMed and PMC, and adds some research studies funded by European institutions, also patents, NHS (National Health Service) guidelines, and Agricola records; *PMC Canada* is also built on PMC and adds studies funded by Canadian institutions; *ClinicalTrials.gov* (Gillen et al. 2004) contains clinical reports of pharmaceutical drug trials with patients; *Drugs@FDA* (Schwartz et al. 2016) lists all drugs and descriptions approved by the FDA (US Food & Drug Administration); *USPTO* (United States Patent and Trademark Office) or *EPO* (European Patent Office) list patent applications, of which some concern the pharmaceutical and medical industries. Finally, hospitals *electronic health records (EHRs)* are suitable for text mining too.

All other challenges common in NLP apply to the biomedical domain too. NLP tools are complex systems that depend on many NLP predictor submodules of which depend on one another as in a stack of tasks (Fig. 1.5). Problematically, errors are compounded, i.e. errors in one level of the stack are carried over to the next level, which may turn all following predictions irrelevant. For example, to know the root of the word "bound", one must know first if it is used as a verb and if so its tense too, as to either declare the root as "bound" (i.e. to jump) if in present tense, or as the root "bind" (i.e. to fasten) if in past tense, or otherwise it may be used as a noun, in which case the root would be "bound" (i.e. a jump). Next, we summarize some of the basic NLP tasks.

Language detection: nearly all NLP tools have some domain-specific knowledge or at least greatly benefit from it. First of all, one must know the language (e.g. English or Spanish) to be able to apply any other NLP subsystem. The biomedical literature is vastly written in English. Nonetheless, 17% of PubMed is written in other languages (~4.5 million articles as of now). A ~3% of PubMed is written in German. Some journals or conferences may be only available in other languages and electronic health records are written in the language of the country of origin.

Topic modeling (detection): second of all, as domain-specific knowledge, one must account for linguistic and structure differences in different topics. As already discussed, the biomedical domain is characterized for a highly-specialized jargon. The language of English news is not the same as that of (English) biomedical papers. Equally, biomedical papers, patent applications, clinical trials, or patient records differ. NLP systems trained for general English may not work for biomedical contexts (e.g. *lemmatizers*, next explained, need to know the complete language vocabulary).

Document classification: related to topic modeling, can be in itself a useful biomedical application. For instance, as previously discussed, model organism databases need to and

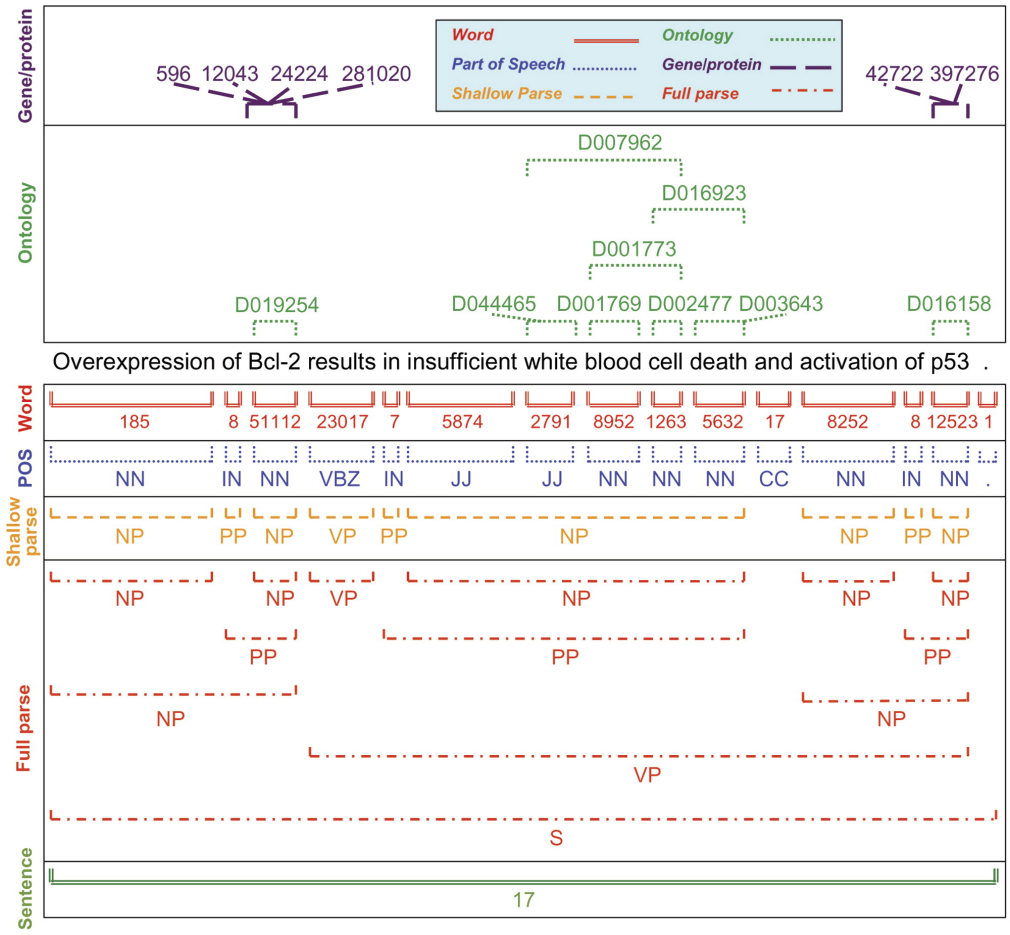


Fig. 1.5. Stack of NLP subsystems; errors cumulate. The full understanding of language requires the accumulation of several NLP tasks. In the figure, from the bottom to the top, common NLP tasks are depicted: *sentence segmentation*, (constituency) *full parsing* (syntax tree), *shallow parsing* (constituents identification such as noun or verb phrases), *part-of-speech* (POS) *tagging*, *word segmentation* or more general *tokenization*, *normalization* first to the MeSH ontology (e.g. "cells" = D002477 or "white blood cells" (leykocytes) = D007962), then *normalization* to Entrez Gene (e.g. "Bcl-2" is normalized to 5 different organisms, 596 for human, 12043 for mouse, 24224 for Norway rat, 281020 for *Bos taurus* (cattle)). Likely an error, the here related "p53" protein is *normalized* to unrelated organisms (42722 for *Drosophila melanogaster*, the fruit fly, and 297276 for pig). *Source:* (Hunter and Cohen 2006), adapted originally from (Nakov et al. n.d.).

filter those publications related to their respective organisms. The task, therefore, is to *cluster* texts into closely related fields.

Sentence segmentation: sentences boundaries must be correctly identified from text. Although seemingly simply detected by periods, some uses of natural language make sentence segmentation a non-so-trivial task: abbreviations or identifiers that use periods, decimal numbers, clauses in parentheses, etc. (Read et al. 2012); all these are frequent in the biomedical domain. In particular, sentence segmentation tools that were trained for general

English may lessen performance on biomedical tasks.

Word segmentation: the space character is a good approximation in English and other Indo-European languages to delimit words. However, contractions like "don't" vs "do not" or "Indoeuropean" vs "Indo-European" vs "Indo European" may have to be accounted too. Other languages such as Chinese, Japanese, or Korean do not have characters for word delimitation, hence complicating the prediction (C.-R. Huang et al. 2007).

Word tokenization: related to but not necessarily equal to word segmentation. Tokenization is the process of dividing text into *tokens*, understood as the basic parsing units that are linguistically meaningful and useful for the methodology or application case at hand. For example, it may be sensible for mutation mention recognition to split the main constituents by special characters and numbers, e.g. "g.123A>G" into the sequence of tokens: "g", ".", "123", "A", ">", "G"; (C.-H. Wei et al. 2013). Tokenization has been shown to be a critical performance component in NLP. As most other NLP submodules depend on this step, tokenization may be either a bottleneck or a leverage. Therefore, given the needs of specialization, many different tokenization strategies exist (He and Kayaalp 2006; Barrett and Weber-Jahnke 2011; Webster and Kit 1992; Dridan and Oepen 2012).

Part-of-speech (POS) tagging: the categorization of words (or tokens) into categories that have similar grammatical properties, such as similar use in the syntax or analogous inflections. Most languages have *nouns* and *verbs* as POS categories, but beyond this, languages display many differences (Kroeger 2005). For example, English has one type of adjective only, whereas Japanese has three (i-adjectives, na-adjectives, and English-like true adjectives). Some languages do not make a strong distinction between adjectives and adverbs (e.g. German). Some languages may not even have nouns and verbs (Broschart 1997); etc.

Stemming and *lemmatization*: are the reduction of words into non-inflected forms, *stems* and *lemmas*, respectively. Stemming does not consider the context of words (disregards grammar), in particular the part of speech, and requires only knowledge of general rules for word inflection. In contrast, lemmatization does take the context into account (considers grammar). For example, lemmatization may correctly identify that "binds" and "bound" are different forms of the same lemma "bind" (to join) or that the lemma for "bad" or "worse" have same lemma "bad", whereas stemming (e.g. with the common Porter 2 stemmer for English), will remove only the inflected appendixes (i.e. "bind" and "bound", and "bad" and "wors", resp.). For regular verbs or common nouns, both methods may provide the same results, e.g. root "work" for the words "works", "working", "worked", etc. Typically, lemmatization is preferred. Notwithstanding, stemming is a faster and simpler process and thereby also often used.

Spelling corrector: errors in spelling can compromise all other NLP subsystems (e.g. lemmatization) and so must be corrected. Generally not a problem in biomedical scientific texts, spelling errors are more frequent, however, in for instance patient records, laboratory, or clinical reports.

Constituency parsing: is the repeated subdivision of a sentence into its sub-parts indicating the relation between these (Fig. 1.5, full parse in red). The result is a hierarchical *syntax parse tree*, starting from a root sentence (S), to more granular phrases, e.g. noun phrase (NP), verb phrase (VP), or prepositional phrase (PP), down to the leaves in the tree, i.e. the individual words labeled with their POS tags.

Shallow parsing (chunking): is the subdivision of a sentence into its main constituents; commonly, noun phrases or verb phrases (Fig. 1.5, shallow parse in yellow). These *chunks* can be derived from a full parse tree or otherwise directly and independently predicted, which normally requires less computation time.

Dependency parsing: is the subdivision of a sentence into a directed hierarchy (typically a tree unless independent clauses exist), expressing the relationships between words, starting from most often a verb as the root (main verb) to words that modify the verb, to subsequent words that modify the previous ones (Fig. 1.6, example).

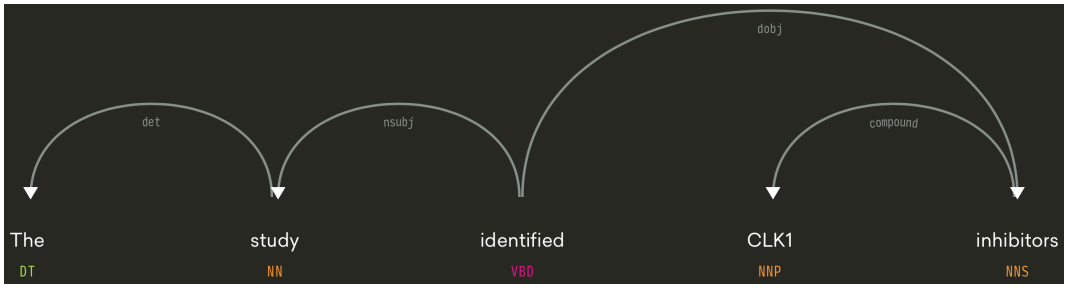


Fig. 1.6. Dependency parsing tree example. Words are interconnected from head words to words that modify those. For instance, here, the verb "identified" is the root, which is modified by "study" (*nsubj*, the nominal subject of the clause) and "inhibitors" (*dobj* or simply *obj*, the direct object of the verb). *Graphic:* `https://demos.explosion.ai/displacy/`

Abbreviation expansion: the resolution of the full expansion of abbreviations, common in the biomedical domain, e.g. "Acs" expanded into the long name "Acetyl-coenzyme A synthetase" (UniProtKB: ACSA_SALTY). Automatic systems must keep track of previously-introduced abbreviations and must consider abbreviations that are never explained in the text at hand, either for being common short names or due to space limitations (as in abstracts).

Coreference resolution: the resolution of expressions that refer to the same thing. Coreferences complicate relationship extraction, for the relations involved in coreferences are only indirectly expressed (Fig. 1.7, example). Coreference resolution is still a hard problem (Choi, Zobel, and K. Verspoor 2016), that has to be solved if we want to transition towards a finer understanding of human language, i.e. to natural language understanding (NLU).

Besides intrinsic NLP tasks, and for completeness in the description of concepts, the reader must be familiar with general terms in machine learning:

Learning: a machine learning model is ultimately an automatic algorithm (hence the *machine* part) that maps inputs to outputs (from different spaces) and is optimized (*learned*)

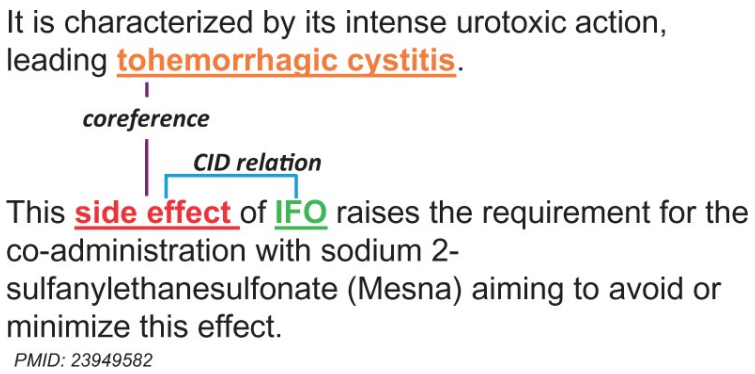


Fig. 1.7. Coreference resolution still hard. In the example, the disease "tohemorrhagic cystitis" (orange) is indirectly associated with the chemical IFO (green) only expressed through the coreferent "side effect" (red). *Source*: (Le et al. 2016). The difficulty of deriving these types of relations complicate our understanding of the biomedical sources.

to do so. For example, named-entity recognition can be reduced to the problem of given some input text, list the text offsets that enclose names of entities, and, most commonly too, associate each enclosed name to a different entity class, e.g. protein or chemical.

Data labeling: in the context of learning, raw data (e.g. unstructured text) is the input and (data) *labels* (e.g. entity text offsets) the output. As should be expected, most data is *unlabeled* with the desired output, hence the need of automatic machine learning methods. Experts and users, however, can manually label some data that can be used for the *training* (optimization) of the machine learning models. For example, as is familiar to the reader, when a user tags a person in a photo, is in fact labeling the photo (the input) by expressing that a particular region (pixels) represent a face or body of a person (the labeled output). Likewise, users can select on a web interface that displays a text article (the input) some words that represent an entity/concept (the output).

Supervised learning: refers to training (optimizing) machine models with completely labeled data (i.e. data manually-labeled and supervised by humans). Here, models must recognize patterns in the input, that lead to the given expected output. For example, a model can perhaps learn that people's faces on a photo are often associated with the pixel patterns of having a somewhat circular region enclosing two smaller white circles and a white line below those. Likewise, a model can perhaps learn that names of entities are often found within noun phrases and surrounded by often repeated words (e.g. "the protein ... functions as a ...").

Unsupervised learning: refers to training machine learning models without any aid of labeled data. Here, models must recognize intrinsic patterns in the input that are statistically significant (e.g. repeated round areas with somewhat orange or black pixels, two sub circles, and a white line) but without really knowing what those patterns represent.

Semi-supervised learning: represents the mixture in which a machine model learns patterns both from labeled and unlabeled data.

Active learning: is a special case of semi-supervised learning, in which a machine can actively query (ask) users (or another information source), for labels in data. Here, a pre-learned model can perhaps find patterns in data for which it does not know or is unsure about its corresponding output labels yet, and so query the user to provide an explanation (learning example) for this case. As data labeling is expensive for users (time consuming), the approach of active learning can reduce this effort, by selectively choosing which data is the most interesting (not understood or seen yet) to learn, and so to focus users on labeling these cases only.

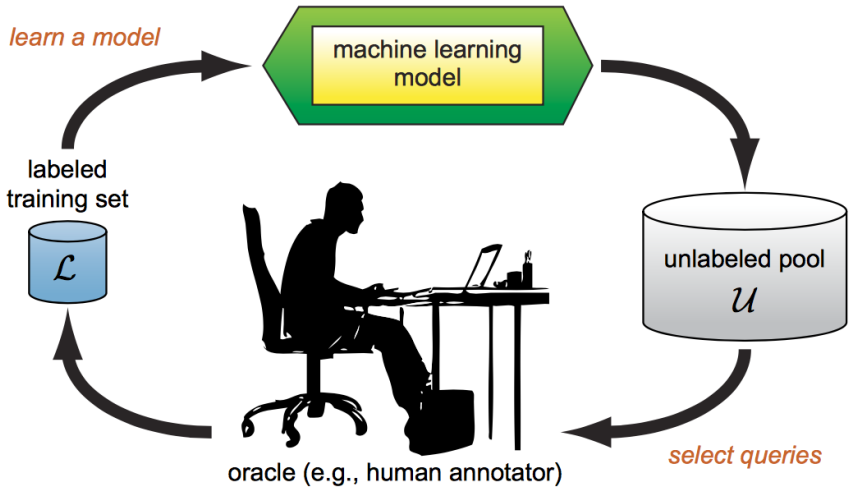


Fig. 1.8. Active learning can reduce labeling efforts. An active learning process begins, often, with an initial set of training labeled data *(L)* annotated by human experts and users (more generally, an oracle). The labeled data is used to learn a model that consequently makes predictions in a set (*pool*) of unseen unlabeled data (*U*); the model then queries the human annotator to label specific cases in which the machine was unsure about (low or intermediate confidence probability in the predictions). The newly labeled data is added to the set *L* of labeled data. The whole process cycles over iterations of manual labeling + automatic querying for new "interesting" (to learn) labels. *Source*: (Settles 2009).

All in all, compounded errors in all the different NLP subtasks, all the many different approaches, plus errors in named-entity recognition (NER), named-entity disambiguation (NED), and relation extraction (RE), explain the large, still unresolved challenges of drawing comprehensive biomedical knowledge from natural language text sources. In fact, natural language understanding is considered to constitute an *AI-complete* problem. That is, an artificial general intelligence (AGI) must solve the problem of language in order to pass the Turing test (Yampolskiy 2013). Notwithstanding, following, we discuss existing methodologies to tackle the problem, the start of the art, and some successes.

1.3 The state of the art & some successes

Various conferences and challenges have assessed the performance and applicability of tools along the years, among the most important: the *BioCreative* challenges (2005-2017) (Hirschman et al. 2005; Arighi et al. 2011; S. Kim et al. 2016; C. H. Wei et al. 2016), the *BioNLP* shared tasks (2009-2017) (J.-D. Kim et al. 2009; Pyysalo et al. 2015), the *BioASQ* challenges (2013-2017) (Tsatsaronis et al. 2015), or the *Biocuration* conferences (2005-2017). Other important NLP-general conferences are *SemEval* (1998-2017) (S. N. Kim et al. 2010), the conferences and workshops of the *ACL* (Association for Computational Linguistics) (1990-2017) (ACL 2016), the *COLING* conferences (1965-2016) (from the International Committee on Computational Linguistics), or the *CoNLL* (Conference on Computational Natural Language Learning) conferences (1997-2017) (CoNLL 1997). All these conferences run until to date, attesting for the difficulty of the problem.

Drawing a conclusive guideline of best methodologies and baseline of best performances is difficult. Many different sub-problems exist and research is active. As things change so fast, the reader is advised to contrast the latest reviews in the mentioned conferences; for instance, the review of named-entity recognition and normalization of diseases and of biomedical relation extraction for chemical-induced disease (C. H. Wei et al. 2016) and of interactive interfaces intended to aid database curators (Q. Wang et al. 2016).

Until recently, the graphical models, *conditional random fields* (CRF) (Lafferty, McCallum, and Pereira 2001) had been used consistently as best-performing methods for named-entity recognition. In the last 4-5 years, however, the so called *deep learning* class of algorithms have taken all fields by storm. The latest best techniques still often combine CRFs with various architectures of *artificial neural networks* (ANNs) or features derived from these. For example, the so called *word embedding* features, i.e. vector representations of words learned unsupervised from large and massive datasets (e.g. PubMed) have been shown to significantly improve tagging performance (Mikolov et al. 2013; Collobert et al. 2011). A myriad of experimentations of neural networks, with or without CRFs, are appearing: *Long Short-Term Memory* (LTSM), LSTMs with CRFs (LSTM-CRF), *bi-directional* LTSM (BI-LSTM), bi-directional LTSM with a CRF layer (BI-LSTM-CRF), *recurrent neural networks* (RNNs), *convolutional neural networks* (CNNs), LSTMs or bi-directional with CNNs (LSTM-CNN, BI-LSTM-CNN), ... A detailed description of these methods is beyond the scope of this work and can be found elsewhere, (Jagannatha and H. Yu 2016; Z. Huang, Xu, and K. Yu 2015; Chiu and Nichols 2015; Strubell et al. 2017; Q. Wei et al. 2016; Strubell et al. 2017; Hu et al. 2016). Likewise, the machine learning models *support vector machines* (*SVMs*) (Cortes and Vapnik 1995) and *random forests* (T. K. Ho 1995), have been used extensively for relation extraction, but deep learning models are now in vogue (Nguyen and Grishman n.d.; Zeng et al. 2014). Named-entity disambiguation methods necessarily require to work with dictionaries of names, e.g. by looking up hashed terms that allow flexible small differences in spelling, (Binder et al. 2014). Yet, word embeddings and

CNNs are also showing improved performance for entity disambiguation (Gottapu, Dagli, and Ali 2016).

With respect to active learning, research gears towards the optimization of achieving high machine performance with the least possible number of queries to human annotators, i.e. in a cost-effective manner. A survey of the field is found in (Settles 2009). Research is ongoing, (K. Wang et al. 2017; W.-N. Hsu and H.-T. Lin 2015; C. L. Li, Ferng, and H. T. Lin n.d.). Research is also active to know how to best include humans for the resolution of many different tasks, e.g. entities disambiguation (Gottapu, Dagli, and Ali 2016), database engines that combine SQL-like queries with queries posed to human crowds (Franklin et al. 2011; Marcus et al. n.d.), optimization of relational database queries (Park and Widom 2013), or just any problem where human supervision can be beneficial (Jamieson et al. 2015). These techniques that try to leverage the intelligence of crowds and humans are called *human-in-the-loop* (HITL), also referred as *human intelligence tasks* (HIT), closely related to generally gathering information from *human intelligence* (HUMINT). As wearable technologies carried over or integrated into human bodies rapidly advance, some authors are already discussing the concept of *humanistic intelligence* (HI), (Minsky, Kurzweil, and Mann 2013; Fung and Mann 2002; Mann 2013), wherein intelligence arises from an instant feedback loop between a human, and a inextricably intertwined computation.

Few BioNLP tools have proven useful in practice yet. A notable exception is the *Textspresso* system (Muller, Kenny, and Sternberg 2004), which was put in place to assist some parts of the curation pipeline in multiple model organism databases (Van Auken et al. 2012), including *WormBase* (T. W. Harris et al. 2014), *dictyBase* (Basu et al. 2013), and *TAIR* (Berardini et al. 2015). Textspresso uses a combination of dictionary look-up methods, *hidden markov models* (HMMs) (Rabiner 1989), and SVMs to aid in document filtering and classification (triage) and entity recognition. With similar technologies, and also combining human input, the system helped in parts of the discovery and indexing of the search engine *Neuroscience Information Framework* (NIF), (Bandrowski et al. 2012; NIF 2010). Textpresso was also used to aid entity normalization resolution by providing suggested hyperlinks in interactive HTML/PDF articles, from words predicted to be entities to the database entries URLs that uniquely identify them (Rangarajan et al. 2011). The system also informs users about entities whose links could not be resolved, and users have the option to edit, accept, or reject the pre-filled links. Other successful tools similarly automatically add or provide links of entities and concepts in texts, although not necessarily to assist database curation, but to ease the reading of scientific articles. These include most importantly the *Reflect* system (Pafilis et al. 2009) and *Utopia* (Attwood et al. 2010). Utopia is a downloadable open software PDF reader (for Windows, Mac, and Linux), that automatically shows to the user contextual information relevant for the displayed article, for instance links to concepts or citations made by other authors. The tool Reflect lets users, either via a browser add-on or via its own page, to post URL pages to be marked up with identifiable entities. Reflect can further be called with a REST API, and thus indeed label and index text documents. Moreover,

Reflect also allows users to collectively provide feedback on the predicted annotations, with looks to keep improving its system over time. Looking at a different application, an early successful example of biomedical text mining was the tool and database *PubGene* (Jenssen et al. 2001), that could draw a large network of interacting genes (139,756 pairs of related genes for 13,712 total genes) based on co-citation in a same publication (Fig. 1.9). The authors showed that this approach could reveal clusters of interacting genes that had been previously assessed experimentally in patients with lymphoma. Finally, the reader finds in (Thessen, Cui, and Mozzherin 2012) a review of other existing BioNLP tools.

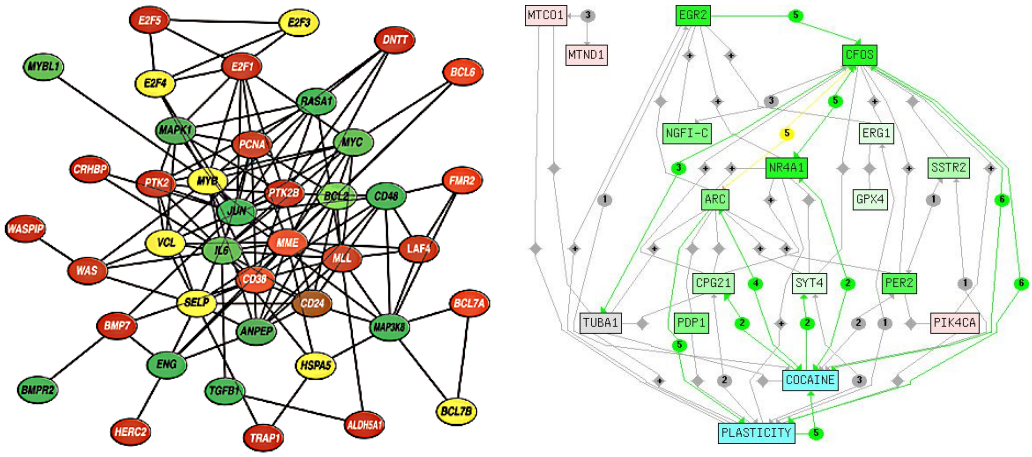


Fig. 1.9. Examples of biological networks drawn by text mining. On the left, a cluster of interacting genes found to be co-cited in PubMed abstracts; *source*: (Jenssen et al. 2001). On the right, a cluster of genes regulated by the drug cocaine text-mined from all PubMed abstracts, at the time; *source*: (H. Chen and Sharp 2004).

1.4 Overview of this work

In Chapter 2, I describe *tagtog*, an interactive web interface designed to aid database curators. The system automatically machine-predicts annotations of entities (e.g. genes) and users can provide feedback on those, to either accept, reject, or edit the annotations. Consequently, the feedback is used to retrain and so improve the internal machine learning systems. In this work, we collaborated with the model organism database FlyBase to annotate hundreds of last published full-text articles and, in doing so, demonstrated a cost-effective annotation approach. We used and describe techniques for active learning and named-entity recognition.

In Chapter 3, I describe *nala*, a new method that text-mines genetic variations, i.e. descriptions in the literature of gene mutations (e.g. "glutamic acid was substituted by valine at residue 6"). The system was optimized to recognize those types of complex natural language mentions, in contrast to simpler mentions (e.g. "E6V"), which were the focus of previous tools. In this work, we demonstrated how to apply active learning-based data labeling to achieve in parallel two things: 1) the creation of the largest collection of mutation descriptions in the literature to date; and 2) a high-performing method that could find all the results of other tools, and uniquely discover new variations that remained until this moment unhidden. We used and show techniques for active learning, named-entity recognition, and deep learning.

In Chapter 4, I describe *LocText*, a new method designed to extract the native localization of proteins from the literature. The method can help in adding novel Cellular Component GO annotations to public databases, most importantly to the standard reference for protein annotations, UniProtKB. We demonstrated upon manual inspection that the text-mined annotations by the new method were highly accurate. We (non experts in database curation) could assess the quality of the automatic annotations rapidly: we could add more than one hundred novel and verified annotations per work day. In this work, we used and discuss techniques for named-entity recognition, named-entity disambiguation, relation extraction, and semi-automatic database curation.

Finally, in Chapter 5, I summarize the main results of this dissertation and discuss the main aim of the work: how text mining can assist human experts.

1.5 References

ACL (2016). "The 54th Annual Meeting of the Association for Computational Linguistics". In: *Proceedings of the Conference* 2.

Adar, E. (2004). "SaRAD: a Simple and Robust Abbreviation Dictionary". In: *Bioinformatics* 20.4, pp. 527–33. ISSN: 1367-4803 (Print) 1367-4803 (Linking). DOI: 10.1093/bioinformatics/btg439. URL: https://www.ncbi.nlm.nih.gov/pubmed/14990448.

Arighi, C. N. et al. (2011). "Overview of the BioCreative III Workshop". In: *BMC Bioinformatics* 12 Suppl 8, S1. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-12-S8-S1. URL: https://www.ncbi.nlm.nih.gov/pubmed/22151647.

Ashburner, M. et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium". In: *Nat Genet* 25.1, pp. 25–9. ISSN: 1061-4036 (Print) 1061-4036 (Linking). DOI: 10.1038/75556. URL: https://www.ncbi.nlm.nih.gov/pubmed/10802651.

Attwood, T. K. et al. (2010). "Utopia documents: linking scholarly literature with research data". In: *Bioinformatics* 26.18, pp. i568–i574. ISSN: 1367-4803 1367-4811. DOI: 10.1093/bioinformatics/btq383. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2935404/.

Bairoch, A. and R. Apweiler (1999). "The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999". In: *Nucleic Acids Res* 27.1, pp. 49–54. ISSN: 0305-1048 (Print) 0305-1048 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/9847139.

Bandrowski, A. E. et al. (2012). "A hybrid human and machine resource curation pipeline for the Neuroscience Information Framework". In: *Database: The Journal of Biological Databases and Curation* 2012, bas005. ISSN: 1758-0463. DOI: 10.1093/database/bas005. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3308161/.

Barrett, Neil and Jens Weber-Jahnke (2011). "Building a biomedical tokenizer using the token lattice design pattern and the adapted Viterbi algorithm". In: *BMC Bioinformatics* 12.Suppl 3, S1–S1. ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-S3-S1. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3111587/.

Basu, Siddhartha et al. (2013). "dictyBase 2013: integrating multiple Dictyostelid species". In: *Nucleic Acids Research* 41.D1, pp. D676–D683. ISSN: 0305-1048. DOI: 10.1093/nar/gks1064. URL: http://dx.doi.org/10.1093/nar/gks1064.

Bateman, A. (2010). "Curators of the world unite: the International Society of Biocuration". In: *Bioinformatics* 26.8, p. 991. ISSN: 1367-4811 (Electronic) 1367-4803 (Linking). DOI: 10.1093/bioinformatics/btq101. URL: https://www.ncbi.nlm.nih.gov/pubmed/20305270.

Bentley, D. R. et al. (2008). "Accurate whole human genome sequencing using reversible terminator chemistry". In: *Nature* 456.7218, pp. 53–9. ISSN: 1476-4687 (Electronic) 0028-0836 (Linking). DOI: 10.1038/nature07517. URL: https://www.ncbi.nlm.nih.gov/pubmed/18987734.

Berardini, T. Z. et al. (2015). "The Arabidopsis information resource: Making and mining the "gold standard" annotated reference plant genome". In: *Genesis* 53.8, pp. 474–85. ISSN: 1526-968X (Electronic) 1526-954X (Linking). DOI: 10.1002/dvg.22877. URL: https://www.ncbi.nlm.nih.gov/pubmed/26201819.

Binder, J. X. et al. (2014). "COMPARTMENTS: unification and visualization of protein subcellular localization evidence". In: *Database (Oxford)* 2014, bau012. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bau012. URL: https://www.ncbi.nlm.nih.gov/pubmed/24573882.

Blake, J. A. et al. (2017). "Mouse Genome Database (MGD)-2017: community knowledge resource for the laboratory mouse". In: *Nucleic Acids Res* 45.D1, pp. D723–D729. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1040. URL: https://www.ncbi.nlm.nih.gov/pubmed/27899570.

Bodenreider, O. (2004). "The Unified Medical Language System (UMLS): integrating biomedical terminology". In: *Nucleic Acids Res* 32.Database issue, pp. D267–70. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkh061. URL: https://www.ncbi.nlm.nih.gov/pubmed/14681409.

Boutet, E. et al. (2016). "UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View". In: *Methods Mol Biol* 1374, pp. 23–54. ISSN: 1940-6029 (Electronic) 1064-3745 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/26519399.

Broschart, Jürgen (1997). "Why Tongan does it differently: Categorial distinctions in a language without nouns and verbs". In: *Linguistic Typology* 1-2, pp. 123–165.

Burge, S. et al. (2012). "Biocurators and biocuration: surveying the 21st century challenges". In: *Database (Oxford)* 2012, bar059. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bar059. URL: https://www.ncbi.nlm.nih.gov/pubmed/22434828.

Chang, J. T., H. Schutze, and R. B. Altman (2002). "Creating an online dictionary of abbreviations from MEDLINE". In: *J Am Med Inform Assoc* 9.6, pp. 612–20. ISSN: 1067-5027 (Print) 1067-5027 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/12386112.

Chen, H. and B. M. Sharp (2004). "Content-rich biological network constructed by mining PubMed abstracts". In: *BMC Bioinformatics* 5, p. 147. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-5-147. URL: https://www.ncbi.nlm.nih.gov/pubmed/15473905.

Cherry, J. M. et al. (2012). "Saccharomyces Genome Database: the genomics resource of budding yeast". In: *Nucleic Acids Res* 40.Database issue, pp. D700–5. ISSN: 1362-4962

(Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkr1029. URL: https://www.ncbi.nlm.nih.gov/pubmed/22110037.

Chiu, Jason P. C. and Eric Nichols (2015). "Named Entity Recognition with Bidirectional LSTM-CNNs". In: *ArXiv e-prints* 1511. URL: http://adsabs.harvard.edu/abs/2015arXiv151108308C.

Choi, M., J. Zobel, and K. Verspoor (2016). "A categorical analysis of coreference resolution errors in biomedical texts". In: *J Biomed Inform* 60, pp. 309–18. ISSN: 1532-0480 (Electronic) 1532-0464 (Linking). DOI: 10.1016/j.jbi.2016.02.015. URL: https://www.ncbi.nlm.nih.gov/pubmed/26925515.

Cohen, K. B. et al. (2010). "The structural and content aspects of abstracts versus bodies of full text journal articles are different". In: *BMC Bioinformatics* 11, p. 492. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-11-492. URL: https://www.ncbi.nlm.nih.gov/pubmed/20920264.

Collobert, Ronan et al. (2011). "Natural Language Processing (Almost) from Scratch". In: *J. Mach. Learn. Res.* 12, pp. 2493–2537. ISSN: 1532-4435.

CoNLL (1997). "Conference on Computational Language Learning". In: *Proceedings of the Conference*.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: http://dx.doi.org/10.1007/BF00994018.

Degtyarenko, K. et al. (2008). "ChEBI: a database and ontology for chemical entities of biological interest". In: *Nucleic Acids Res* 36.Database issue, pp. D344–50. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkm791. URL: https://www.ncbi.nlm.nih.gov/pubmed/17932057.

Dridan, Rebecca and Stephan Oepen (2012). "Tokenization: returning to a long solved problem a survey, contrastive experiment, recommendations, and toolkit". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*. 2390750: Association for Computational Linguistics, pp. 378–382.

Dunnen, Johan T. den et al. (2016). "HGVS Recommendations for the Description of Sequence Variants: 2016 Update". In: *Hum. Mutat.* 37.6, pp. 564–569. ISSN: 1059-7794. DOI: 10.1002/humu.22981. URL: http://dx.doi.org/10.1002/humu.22981%20http://www.ncbi.nlm.nih.gov/pubmed/26931183.

Europe, P. M. C. Consortium (2015). "Europe PMC: a full-text literature database for the life sciences and platform for innovation". In: *Nucleic Acids Res* 43.Database issue, pp. D1042–8. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gku1061. URL: https://www.ncbi.nlm.nih.gov/pubmed/25378340.

First, M. B. et al. (2015). "The development of the ICD-11 Clinical Descriptions and Diagnostic Guidelines for Mental and Behavioural Disorders". In: *World Psychiatry* 14.1, pp. 82–90. ISSN: 1723-8617 (Print) 1723-8617 (Linking). DOI: 10.1002/wps.20189. URL: https://www.ncbi.nlm.nih.gov/pubmed/25655162.

Franklin, Michael J. et al. (2011). "CrowdDB: answering queries with crowdsourcing". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 1989331: ACM, pp. 61–72. DOI: 10.1145/1989323.1989331.

Fung, J. and S. Mann (2002). "Exploring humanistic intelligence through physiologically mediated reality". In: *Proceedings. International Symposium on Mixed and Augmented Reality*, pp. 275–276. DOI: 10.1109/ISMAR.2002.1115110.

Gillen, J. E. et al. (2004). "Design, implementation and management of a web-based data entry system for ClinicalTrials.gov". In: *Stud Health Technol Inform* 107.Pt 2, pp. 1466–70. ISSN: 0926-9630 (Print) 0926-9630 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/15361058.

Gottapu, Ram Deepak, Cihan Dagli, and Bharami Ali (2016). "Entity Resolution Using Convolutional Neural Network". In: *Procedia Computer Science* 95, pp. 153–158. ISSN: 1877-0509. DOI: http://dx.doi.org/10.1016/j.procs.2016.09.306. URL: http://www.sciencedirect.com/science/article/pii/S1877050916324796.

Gramates, L. S. et al. (2017). "FlyBase at 25: looking to the future". In: *Nucleic Acids Res* 45.D1, pp. D663–D671. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1016. URL: https://www.ncbi.nlm.nih.gov/pubmed/27799470.

Hanna, J. et al. (2013). "Building a drug ontology based on RxNorm and other sources". In: *J Biomed Semantics* 4.1, p. 44. DOI: 10.1186/2041-1480-4-44. URL: https://www.ncbi.nlm.nih.gov/pubmed/24345026.

Harris, T. W. et al. (2014). "WormBase 2014: new views of curated biology". In: *Nucleic Acids Res* 42.Database issue, pp. D789–93. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkt1063. URL: https://www.ncbi.nlm.nih.gov/pubmed/24194605.

Hatzivassiloglou, V., P. A. Duboue, and A. Rzhetsky (2001). "Disambiguating proteins, genes, and RNA in text: a machine learning approach". In: *Bioinformatics* 17 Suppl 1, S97–106. ISSN: 1367-4803 (Print) 1367-4803 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/11472998.

Hayden, Erika Check (2014). "Is the $1,000 genome for real?" In: *Nature News*. DOI: doi:10.1038/nature.2014.14530.

He, Ying and Mehmet Kayaalp (2006). "A Comparison of 13 Tokenizers on MEDLINE". In: *Bethesda, MD: The Lister Hill National Center for Biomedical Communications*.

Hilbert, M. and P. Lopez (2011). "The world's technological capacity to store, communicate, and compute information". In: *Science* 332.6025, pp. 60–5. ISSN: 1095-9203 (Electronic) 0036-8075 (Linking). DOI: 10.1126/science.1200970. URL: https://www.ncbi.nlm.nih.gov/pubmed/21310967.

Hirschman, L. et al. (2005). "Overview of BioCreAtIvE: critical assessment of information extraction for biology". In: *BMC Bioinformatics* 6 Suppl 1, S1. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-6-S1-S1. URL: https://www.ncbi.nlm.nih.gov/pubmed/15960821.

Ho, Tin Kam (1995). "Random decision forests". In: *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*. 844681: IEEE Computer Society, p. 278.

Hsu, Wei-Ning and Hsuan-Tien Lin (2015). "Active learning by learning". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2886691: AAAI Press, pp. 2659–2665.

Hu, Zhiting et al. (2016). "Harnessing Deep Neural Networks with Logic Rules". In: *ArXiv e-prints* 1603. URL: http://adsabs.harvard.edu/abs/2016arXiv160306318H.

Huang, Chu-Ren et al. (2007). "Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification". In: *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. ACL '07. Prague, Czech Republic: Association for Computational Linguistics, pp. 69–72. URL: http://dl.acm.org/citation.cfm?id=1557769.1557791.

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging". In: *ArXiv e-prints* 1508. URL: http://adsabs.harvard.edu/abs/2015arXiv150801991H.

Hunter, L. and K. B. Cohen (2006). "Biomedical language processing: what's beyond PubMed?" In: *Mol Cell* 21.5, pp. 589–94. ISSN: 1097-2765 (Print) 1097-2765 (Linking). DOI: 10.1016/j.molcel.2006.02.012. URL: https://www.ncbi.nlm.nih.gov/pubmed/16507357.

IHGSC (2001). "The sequence of the human genome". In: *Science* 291.5507, pp. 1304–51. ISSN: 0036-8075 (Print) 0036-8075 (Linking). DOI: 10.1126/science.1058040. URL: https://www.ncbi.nlm.nih.gov/pubmed/11181995.

Jagannatha, Abhyuday and Hong Yu (2016). "Structured prediction models for (RNN) based sequence labeling in clinical text". In: *CoRR* abs/1608.00612. URL: http://arxiv.org/abs/1608.00612.

Jamieson, Kevin et al. (2015). "NEXT: a system for real-world development, evaluation, and application of active learning". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. 2969536: MIT Press, pp. 2656–2664.

Jenssen, Tor-Kristian et al. (2001). "A literature network of human genes for high-throughput analysis of gene expression". In: *Nat Genet* 28.1, pp. 21–28. ISSN: 1061-4036. DOI: http://www.nature.com/ng/journal/v28/n1/suppinfo/ng0501_21_S1.html. URL: http://dx.doi.org/10.1038/ng0501-21.

Jiang, Y. et al. (2016). "An expanded evaluation of protein function prediction methods shows an improvement in accuracy". In: *Genome Biol* 17.1, p. 184. ISSN: 1474-760X (Electronic) 1474-7596 (Linking). DOI: 10.1186/s13059-016-1037-6. URL: https://www.ncbi.nlm.nih.gov/pubmed/27604469.

Jimeno Y., A. and K. Verspoor (2014). "Literature mining of genetic variants for curation: quantifying the importance of supplementary material". In: *Database (Oxford)* 2014,

bau003. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bau003. URL: https://www.ncbi.nlm.nih.gov/pubmed/24520105.

Jimeno Y., Antonio and Karin Verspoor (2014). "Mutation extraction tools can be combined for robust recognition of genetic variants in the literature". In: *F1000Res.* 3, p. 18. ISSN: 2046-1402. DOI: 10.12688/f1000research.3-18.v2. URL: http://dx.doi.org/10.12688/f1000research.3-18.v2%20http://www.ncbi.nlm.nih.gov/pubmed/25285203%20http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4176422%20http://f1000research.com/articles/10.12688/f1000research.3-18.v2/doi.

Kibbe, W. A. et al. (2015). "Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data". In: *Nucleic Acids Res* 43.Database issue, pp. D1071–8. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gku1011. URL: https://www.ncbi.nlm.nih.gov/pubmed/25348409.

Kim, Jin-Dong et al. (2009). "Overview of BioNLP'09 shared task on event extraction". In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. Boulder, Colorado: Association for Computational Linguistics, pp. 1–9.

Kim, Su Nam et al. (2010). "SemEval-2010 task 5: Automatic keyphrase extraction from scientific articles". In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. 1859668: Association for Computational Linguistics, pp. 21–26.

Kim, S. et al. (2016). "BioCreative V BioC track overview: collaborative biocurator assistant task for BioGRID". In: *Database (Oxford)* 2016. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/baw121. URL: https://www.ncbi.nlm.nih.gov/pubmed/27589962.

Kohler, S. et al. (2017). "The Human Phenotype Ontology in 2017". In: *Nucleic Acids Res* 45.D1, pp. D865–D876. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1039. URL: https://www.ncbi.nlm.nih.gov/pubmed/27899602.

Kroeger, P.R. (2005). *Analyzing Grammar: An Introduction*. Cambridge University Press. ISBN: 9780521816229. URL: https://books.google.de/books?id=PNO3ngEACAAJ.

Kryshtafovych, A. et al. (2014). "Challenging the state of the art in protein structure prediction: Highlights of experimental target structures for the 10th Critical Assessment of Techniques for Protein Structure Prediction Experiment CASP10". In: *Proteins* 82 Suppl 2, pp. 26–42. ISSN: 1097-0134 (Electronic) 0887-3585 (Linking). DOI: 10.1002/prot.24489. URL: https://www.ncbi.nlm.nih.gov/pubmed/24318984.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. 655813: Morgan Kaufmann Publishers Inc., pp. 282–289.

Lander, E. S. et al. (2001). "Initial sequencing and analysis of the human genome". In: *Nature* 409.6822, pp. 860–921. ISSN: 0028-0836 (Print) 0028-0836 (Linking). DOI: 10.1038/35057062. URL: https://www.ncbi.nlm.nih.gov/pubmed/11237011.

Le, H. Q. et al. (2016). "Sieve-based coreference resolution enhances semi-supervised learning model for chemical-induced disease relation extraction". In: *Database (Oxford)* 2016. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/baw102. URL: https://www.ncbi.nlm.nih.gov/pubmed/27630201.

Lek, M. et al. (2016). "Analysis of protein-coding genetic variation in 60,706 humans". In: *Nature* 536.7616, pp. 285–91. ISSN: 1476-4687 (Electronic) 0028-0836 (Linking). DOI: 10.1038/nature19057. URL: https://www.ncbi.nlm.nih.gov/pubmed/27535533.

Lensink, M. F. and S. J. Wodak (2013). "Docking, scoring, and affinity prediction in CAPRI". In: *Proteins* 81.12, pp. 2082–95. ISSN: 1097-0134 (Electronic) 0887-3585 (Linking). DOI: 10.1002/prot.24428. URL: https://www.ncbi.nlm.nih.gov/pubmed/24115211.

Leser, U. and J. Hakenberg (2005). "What makes a gene name? Named entity recognition in the biomedical literature". In: *Brief Bioinform* 6.4, pp. 357–69. ISSN: 1467-5463 (Print) 1467-5463 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/16420734.

Li, C. L., C. S. Ferng, and H. T. Lin (n.d.). "Active Learning Using Hint Information". In: 1530-888X (Electronic).

Lin, J. (2009). "Is searching full text more effective than searching abstracts?" In: *BMC Bioinformatics* 10, p. 46. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-10-46. URL: https://www.ncbi.nlm.nih.gov/pubmed/19192280.

Liu, S. et al. (2005). "RxNorm: prescription for electronic drug information exchange". In: *IT Professional* 7.5, pp. 17–23. ISSN: 1520-9202. DOI: 10.1109/MITP.2005.122.

Maglott, D. et al. (2011). "Entrez Gene: gene-centered information at NCBI". In: *Nucleic Acids Res* 39.Database issue, pp. D52–7. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkq1237. URL: https://www.ncbi.nlm.nih.gov/pubmed/21115458.

Major, P., B. J. Kostrewski, and J. Anderson (1978). "Analysis of the semantic structures of medical reference languages: part 2. Analysis of the semantic power of MeSH, ICD and SNOMED". In: *Med Inform (Lond)* 3.4, pp. 269–81. ISSN: 0307-7640 (Print) 0307-7640 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/370473.

Mann, S. (2013). "Veillance and reciprocal transparency: Surveillance versus sousveillance, AR glass, lifeglogging, and wearable computing". In: *2013 IEEE International Symposium on Technology and Society (ISTAS): Social Implications of Wearable Computing and Augmediated Reality in Everyday Life*, pp. 1–12. ISBN: 2158-3404. DOI: 10.1109/ISTAS.2013.6613094.

Marcus, Adam et al. (n.d.). "Crowdsourced databases: Query processing with people". In: Cidr.

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Minsky, M., R. Kurzweil, and S. Mann (2013). "The society of intelligent veillance". In: *2013 IEEE International Symposium on Technology and Society (ISTAS): Social Implications of Wearable Computing and Augmediated Reality in Everyday Life*, pp. 13–17. ISBN: 2158-3404. DOI: 10.1109/ISTAS.2013.6613095.

Muller, H. M., E. E. Kenny, and P. W. Sternberg (2004). "Textpresso: an ontology-based information retrieval and extraction system for biological literature". In: *PLoS Biol* 2.11, e309. ISSN: 1545-7885 (Electronic) 1544-9173 (Linking). DOI: 10.1371/journal.pbio.0020309. URL: https://www.ncbi.nlm.nih.gov/pubmed/15383839.

Musen, M. A. et al. (2012). "The National Center for Biomedical Ontology". In: *J Am Med Inform Assoc* 19.2, pp. 190–5. ISSN: 1527-974X (Electronic) 1067-5027 (Linking). DOI: 10.1136/amiajnl-2011-000523. URL: https://www.ncbi.nlm.nih.gov/pubmed/22081220.

Nakov, Preslav et al. (n.d.). "Supporting annotation layers for natural language processing". In: *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*. 1225770: Association for Computational Linguistics, pp. 65–68. DOI: 10.3115/1225753.1225770.

Nelson, S. J. (n.d.). "Medical Terminologies That Work: The Example of MeSH". In: *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, pp. 380–384. ISBN: 1087-4089. DOI: 10.1109/I-SPAN.2009.84.

Nguyen, Thien Huu and Ralph Grishman (n.d.). "Relation extraction: Perspective from convolutional neural networks". In: *Proceedings of NAACL-HLT*, pp. 39–48.

NHGRI (2010). "The Human Genome Project Completion: Frequently Asked Questions". In: *NIH News*. URL: https://www.genome.gov/11006943/human-genome-project-completion-frequently-asked-questions/.

NIF (2010). "NIF: Neuroscience Information Framework". In: *Reference Reviews* 24.7, pp. 43–44. DOI: doi:10.1108/09504121011077372. URL: http://www.emeraldinsight.com/doi/abs/10.1108/09504121011077372.

Pafilis, E. et al. (2009). "Reflect: augmented browsing for the life scientist". In: *Nat Biotechnol* 27.6, pp. 508–10. ISSN: 1546-1696 (Electronic) 1087-0156 (Linking). DOI: 10.1038/nbt0609-508. URL: https://www.ncbi.nlm.nih.gov/pubmed/19513049.

Park, Hyunjung and Jennifer Widom (2013). "Query optimization over crowdsourced data". In: *Proc. VLDB Endow.* 6.10, pp. 781–792. ISSN: 2150-8097. DOI: 10.14778/2536206.2536207.

Pyysalo, S. et al. (2015). "Overview of the Cancer Genetics and Pathway Curation tasks of BioNLP Shared Task 2013". In: *BMC Bioinformatics* 16 Suppl 10, S2. ISSN: 1471-

2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-16-S10-S2. URL: https://www.ncbi.nlm.nih.gov/pubmed/26202570.

Rabiner, L. R. (1989). "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286. ISSN: 0018-9219. DOI: 10.1109/5.18626.

Rangarajan, Arun et al. (2011). "Toward an interactive article: integrating journals and biological databases". In: *BMC Bioinformatics* 12.1, p. 175. ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-175. URL: http://dx.doi.org/10.1186/1471-2105-12-175.

Read, Jonathon et al. (2012). "Sentence Boundary Detection: A Long Solved Problem?" In: pp. 985–994. URL: http://aclweb.org/anthology/C/C12/C12-2096.pdf.

Salimi, N. and R. Vita (2006). "The biocurator: connecting and enhancing scientific data". In: *PLoS Comput Biol* 2.10, e125. ISSN: 1553-7358 (Electronic) 1553-734X (Linking). DOI: 10.1371/journal.pcbi.0020125. URL: https://www.ncbi.nlm.nih.gov/pubmed/17069454.

Sayers, E. W. et al. (2010). "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Res* 38.Database issue, pp. D5–16. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkp967. URL: https://www.ncbi.nlm.nih.gov/pubmed/19910364.

Schwartz, L. M. et al. (2016). "ClinicalTrials.gov and Drugs@FDA: A Comparison of Results Reporting for New Drug Approval Trials". In: *Ann Intern Med* 165.6, pp. 421–30. ISSN: 1539-3704 (Electronic) 0003-4819 (Linking). DOI: 10.7326/M15-2658. URL: https://www.ncbi.nlm.nih.gov/pubmed/27294570.

Settles, Burr (2009). *Active Learning Literature Survey*. Report. URL: http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

Shahpori, R. and C. Doig (2010). "Systematized Nomenclature of Medicine-Clinical Terms direction and its implications on critical care". In: *J Crit Care* 25.2, 364 e1-9. ISSN: 1557-8615 (Electronic) 0883-9441 (Linking). DOI: 10.1016/j.jcrc.2009.08.008. URL: https://www.ncbi.nlm.nih.gov/pubmed/19836194.

Simonite, Tom (2005). "Pokemon blocks gene name". In: *Nature* 438.7070, pp. 897–897. ISSN: 0028-0836. URL: http://dx.doi.org/10.1038/438897a.

Smith, B. et al. (2007). "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration". In: *Nat Biotechnol* 25.11, pp. 1251–5. ISSN: 1087-0156 (Print) 1087-0156 (Linking). DOI: 10.1038/nbt1346. URL: https://www.ncbi.nlm.nih.gov/pubmed/17989687.

Stephens, Z. D. et al. (2015). "Big Data: Astronomical or Genomical?" In: *PLoS Biol* 13.7, e1002195. ISSN: 1545-7885 (Electronic) 1544-9173 (Linking). DOI: 10.1371/journal.pbio.1002195. URL: https://www.ncbi.nlm.nih.gov/pubmed/26151137.

Strubell, Emma et al. (2017). "Fast and Accurate Sequence Labeling with Iterated Dilated Convolutions". In: *ArXiv e-prints* 1702. URL: http://adsabs.harvard.edu/abs/2017arXiv170202098S.

The UniProt Consortium (2017). "UniProt: the universal protein knowledgebase". In: *Nucleic Acids Res* 45.D1, pp. D158–D169. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1099. URL: https://www.ncbi.nlm.nih.gov/pubmed/27899622.

Thessen, A. E., H. Cui, and D. Mozzherin (2012). "Applications of natural language processing in biodiversity science". In: *Adv Bioinformatics* 2012, p. 391574. ISSN: 1687-8035 (Electronic) 1687-8027 (Linking). DOI: 10.1155/2012/391574. URL: https://www.ncbi.nlm.nih.gov/pubmed/22685456.

Tsatsaronis, G. et al. (2015). "An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition". In: *BMC Bioinformatics* 16, p. 138. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/s12859-015-0564-6. URL: https://www.ncbi.nlm.nih.gov/pubmed/25925131.

Van Auken, K. et al. (2012). "Text mining in the biocuration workflow: applications for literature curation at WormBase, dictyBase and TAIR". In: *Database (Oxford)* 2012, bas040. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bas040. URL: https://www.ncbi.nlm.nih.gov/pubmed/23160413.

Van Noorden, R. (2014). "Elsevier opens its papers to text-mining". In: *Nature* 506.7486, p. 17. ISSN: 1476-4687 (Electronic) 0028-0836 (Linking). DOI: 10.1038/506017a. URL: https://www.ncbi.nlm.nih.gov/pubmed/24499898.

Wang, Keze et al. (2017). "Cost-Effective Active Learning for Deep Image Classification". In: *ArXiv e-prints* 1701. URL: http://adsabs.harvard.edu/abs/2017arXiv170103551W.

Wang, Q. et al. (2016). "Overview of the interactive task in BioCreative V". In: *Database (Oxford)* 2016. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/baw119. URL: https://www.ncbi.nlm.nih.gov/pubmed/27589961.

Webster, J. J. and C. Kit (1992). "Tokenization as the initial phase in NLP". In: *Proceedings of the 14th conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics. DOI: 10.3115/992424.992434. URL: http://dx.doi.org/10.3115/992424.992434.

Wei, C. H. et al. (2016). "Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task". In: *Database (Oxford)* 2016. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/baw032. URL: https://www.ncbi.nlm.nih.gov/pubmed/26994911.

Wei, Chih-Hsuan et al. (2013). "tmVar: a text mining approach for extracting sequence variants in biomedical literature". In: *Bioinformatics* 29.11, pp. 1433–1439. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt156. URL: http://dx.doi.org/10.1093/bioinformatics/btt156%20http://www.ncbi.nlm.nih.gov/pubmed/23564842%20http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3661051%

20http://bioinformatics.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=23564842.

Wei, Qikang et al. (2016). "Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks". In: *Database: The Journal of Biological Databases and Curation* 2016, baw140. ISSN: 1758-0463. DOI: 10.1093/database/baw140. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5088735/.

Wetterstrand, KA (2017). "DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)". In: URL: https://www.genome.gov/sequencingcostsdata.

Whetzel, P. L. et al. (2011). "BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications". In: *Nucleic Acids Res* 39.Web Server issue, W541–5. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkr469. URL: https://www.ncbi.nlm.nih.gov/pubmed/21672956.

WHO (1992). "The ICD-10 classification of mental and behavioural disorders: Clinical descriptions and diagnostic guidelines". In: *Geneva: World Health Organization*. URL: http://www.who.int/classifications/icd/en/bluebook.pdf.

Yampolskiy, Roman V. (2013). "Turing Test as a Defining Feature of AI-Completeness". In: *Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing*. Ed. by Xin-She Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–17. ISBN: 978-3-642-29694-9. DOI: 10.1007/978-3-642-29694-9_1. URL: http://dx.doi.org/10.1007/978-3-642-29694-9_1.

Zeng, Daojian et al. (2014). "Relation Classification via Convolutional Deep Neural Network". In: *COLING*.

Chapter 2

tagtog: a human-assisted automatic annotation system

2.1 Preface

The great challenge upon biological databases is the explosion of unstructured data (e.g. literature records) and how to transform it into useful structured knowledge (e.g. how a molecular drug and a protein target react with each other). For example, curators at FlyBase, the premier database of the model organism *Drosophila melanogaster*, spend their time thoroughly analyzing all Drosophila-related research articles (Gramates et al. 2017). Human curators, however, cannot cope with the ever increasing number of new publications; for Drosophila alone, from ~1000 in the late 1980s to >3000 a year in the 2010s (Bunt et al. 2012). Databases of other model organisms such as mouse, yeast, or maize, face the same problem. Text mining methods were studied in the past to automatically leverage this tedious work. However, despite extensive efforts, few systems proved useful in practice yet (Mao et al. 2014).

In this work we studied *tagtog*, a new system that combines an automatic domain-independent named-entity recognizer (NER) with a web editor, for users to manually add and correct annotations. We applied *tagtog* to create a new corpus of articles published between 2011 and 2013 with textual annotations of Drosophila genes and symbols. We annotated full-text articles, a harder problem than abstracts only (Cohen et al. 2010). We created the corpus in three iterations: 1) One curator annotated 20 documents. The automatic system of *tagtog* was trained on this set and then used to recognize gene names in 99 unseen documents. The curator reviewed and corrected when corresponding the automatic annotations. 2) The automatic method was retrained on the until then total 119 documents, and tested on an independent test set of 20 documents. 3) Five curators annotated 312 new documents. The automatic system was re-trained on the now total 431 documents, and tested again on the test set. All in all, we annotated 451 full-text articles, one of the largest resources of labeled data for NLP to date. All annotations were useful and integrated into

the FlyBase database. The automatic system of *tagtog* showed increased performance along the iterations. Further, two other curators assessed *tagtog* for possible savings in annotation time. One curator repeated the experiment of annotating Drosophila genes while the other annotated genes in Maize-related articles for the MaizeGDB database (Andorf et al. 2016). The two curators used the manual interface of *tagtog* to annotate, independently, 20 full-text articles. The domain-independent system was again retrained and used to annotate 20 unseen documents. The curators corrected the automatic predictions and compared the spent time on both phases. The combined, computer and user-corrected annotation was 1.6 to 1.9 faster than manual annotation alone.

All methods and data analyses were done by me. I carried out necessary background research. The project was designed by me and Peter McQuilton. The annotation of the corpus was done by Peter McQuilton, Laura Ponting, Steven J. Marygold, Raymund Stefancsik, and Gillian H. Millburn. The manuscript was drafted by me and Peter McQuilton.

2.2 Journal article. Cejuela *et al.*, *Database (Oxford)* 2014;2014(0):bau033

Starts next page.

DATABASE
The Journal of Biological Databases and Curation

Database, 2014, 1–8
doi: 10.1093/database/bau033
Original article

Original article

tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles

Juan Miguel Cejuela[1], Peter McQuilton[2,*], Laura Ponting[2], Steven J. Marygold[2], Raymund Stefancsik[2], Gillian H. Millburn[2], Burkhard Rost[3] and the FlyBase Consortium

[1]Goerresstr. 20, Munich 80798, Germany, [2]Department of Genetics, University of Cambridge, Cambridge CB2 3EH, UK, [3]Department of Informatics, Technical University of Munich (TUM), Garching 85748, Germany

*Corresponding author: Tel: 0044 (0)1223 333963; Fax: 0044 (0)1223 766732; Email: pam51@gen.cam.ac.uk

Citation details: Cejuela,J.M., McQuilton,P., Ponting,L. *et al.* tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database* (2014) Vol. 2014: article ID bau033; doi:10.1093/database/bau033.

Received 24 January 2014; Revised 10 March 2014; Accepted 14 March 2014

Abstract

The breadth and depth of biomedical literature are increasing year upon year. To keep abreast of these increases, FlyBase, a database for Drosophila genomic and genetic information, is constantly exploring new ways to mine the published literature to increase the efficiency and accuracy of manual curation and to automate some aspects, such as triaging and entity extraction. Toward this end, we present the 'tagtog' system, a web-based annotation framework that can be used to mark up biological entities (such as genes) and concepts (such as Gene Ontology terms) in full-text articles. tagtog leverages manual user annotation in combination with automatic machine-learned annotation to provide accurate identification of gene symbols and gene names. As part of the BioCreative IV Interactive Annotation Task, FlyBase has used tagtog to identify and extract mentions of *Drosophila melanogaster* gene symbols and names in full-text biomedical articles from the PLOS stable of journals. We show here the results of three experiments with different sized corpora and assess gene recognition performance and curation speed. We conclude that tagtog-named entity recognition improves with a larger corpus and that tagtog-assisted curation is quicker than manual curation.

Database URL: www.tagtog.net, www.flybase.org

Database, Vol. 2014, Article ID bau033

Introduction

A major challenge facing biological databases today is the increase in data available for curation. Concurrent with an increase in the number of biological journals, there has been a movement from printed literature to web-based HTML and PDF. This has removed many of the financial and technical constraints on the length and the number of articles a journal can publish. For the past 30 years, the number of Drosophila-related primary research articles published each year has steadily increased from ~1000 in 1980 to >2000 a year since 2001 (1). FlyBase (http://flybase.org) is the premier database of *Drosophila melanogaster* genes and genomes (2) and manually curates Drosophila-related information from the published literature. This information hangs from genetic objects, such as genes, alleles and transgenic constructs. Our genetic literature curation pipeline has two main stages: (i) skim or author curation, where the genes in a paper are identified, and flags are added to indicate the presence of a new reagent or data type (e.g. a new allele or gene expression in a perturbed background), and (ii) full curation, where all other genetic objects are added and annotated with phenotypic, molecular, expression and interaction data. Manually curating each gene mentioned in a paper is a time-consuming process and takes a significant amount of curator effort. Finding a way to automate this process would greatly increase curation efficiency, not to mention the number of papers that could be fully curated.

Since the meeting at the BioCreative workshop in 2012, FlyBase has been collaborating with tagtog to identify and extract Drosophila gene mentions from PLOS journals (3). tagtog (http://tagtog.net) is a web-based framework for the annotation of named entities. The tagtog system allows biocurators to annotate gene symbols manually and leverages machine learning methods to perform the same type of annotations computationally (Figure 1). Initially, the tool is trained with a small set of manually annotated documents. tagtog can then be used to process a set of novel documents wherein automatically generated predictions are made, which can be reviewed and validated by the user. This continuous and interactive retraining of the machine learning methods with user feedback can lead to an ever-improving performance in automatic prediction (4). Once optimized, the trained machine learning methods can be used to process and annotate a large volume of documents to a sufficiently accurate level.

In this collaboration between FlyBase and tagtog, we have annotated >450 PLOS journal articles and explored whether the size of the annotated corpus affects the precision and recall of automatic named entity recognition (NER) and whether NER can speed up gene symbol and name annotation.

The tagtog system

In the following section, we briefly showcase some of tagtog's main features.

- **Multiple projects:** users can create different annotation projects and load their own dictionaries and corpora.
- **Team collaboration:** multiple users on the same project are also supported, allowing curation teams to view and annotate the same set of documents.
- **Entity normalization:** entities (such as gene names) can be normalized to unique identifiers (IDs) using a reference dictionary submitted by the user.
- **Active learning:** tagtog actively asks for user feedback on predicted annotations. A proposed mechanism was already developed in an early version of tagtog, presented at the BioCreative 2012 workshop (5).
- **Document searching:** papers can be searched using the search tool at the top of the interface. Options include searching by document ID (based on the digital object identifier), entities or whether a paper has been fully annotated. In the future, we hope to add the facility to search by PubMed ID (PMID).
- **Browser support:** the system runs on all major current browsers only requiring HTML5 and javascript. Chrome and Firefox are officially supported. Other browsers like Opera, Safari and Internet Explorer (9 and 10) are regularly tested but lack official support at this point.
- **Import options:** any paper following the NCBI Journal Publishing Tag Set (6) or the BioMed Central format (7) can be uploaded to tagtog. This includes full-text papers from the PLOS, BioMed Central, Chemistry Central and Springer Open collections. In the near future, we will accept papers from the new JATS format (8) and plain text files.
- **Export options:** three export file formats are supported: a tab-separated list of terms linked to PMIDs (TSV format), the new BioC format (9) and 'anndoc' XML, our in-house format. Further file formats can be added on request.

Defining the annotation guidelines

On project creation in tagtog, the first step for a user is to define the annotation guidelines (Figure 2). These guidelines stipulate what should be annotated and how this relates to the entity class. There are the following options:

- **Entity:** choose the entity class name to annotate. For this project, we chose to annotate all *D. melanogaster* gene mentions, both as symbols (for example, 'dpp' or 'amn') and names (for example, 'decapentaplegic' or

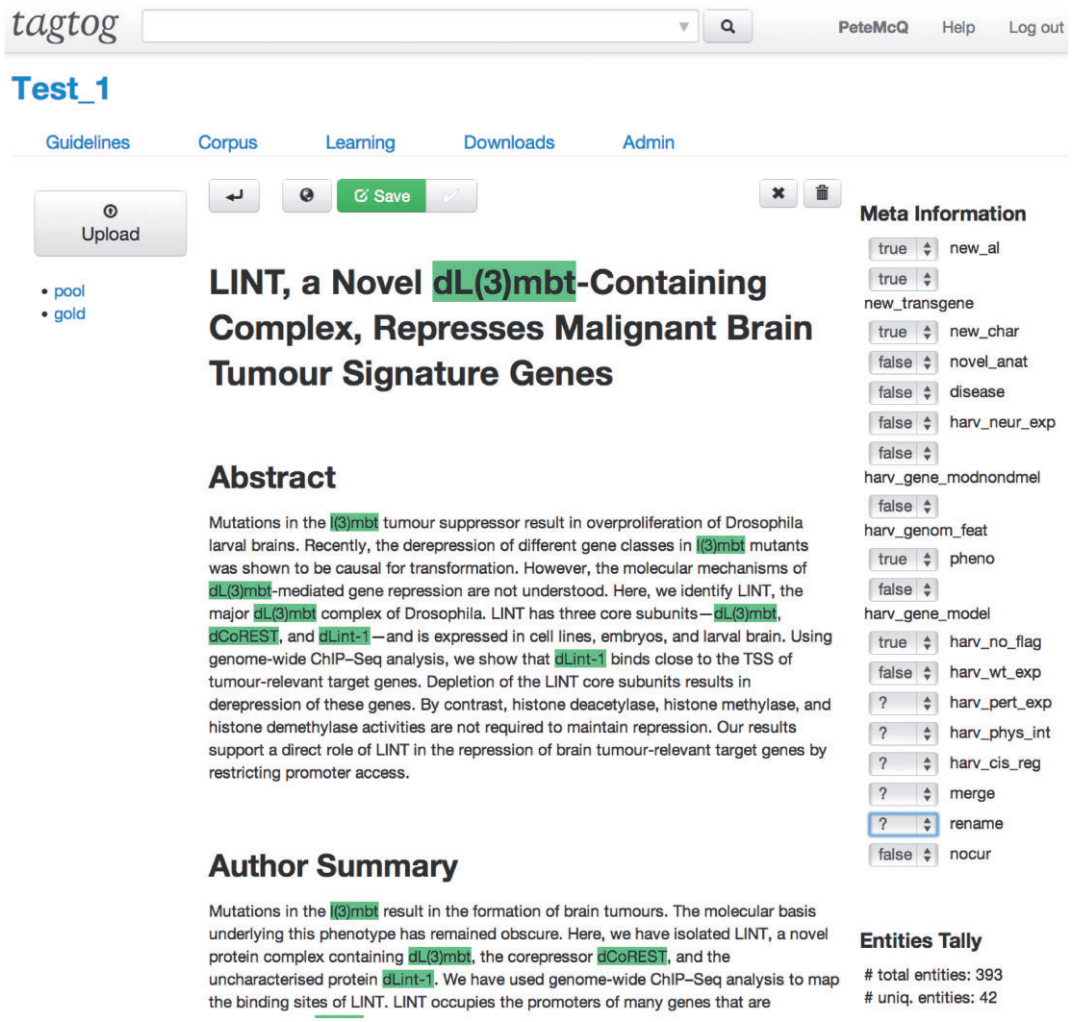


Figure 1. Example of the document display and editor in tagtog.

'amnesiac'), where the gene is a separate string or is separated from another entity by a hyphen. We also included some non-Drosophila genes, such as the commonly used GAL4 drivers from the UAS-GAL4 system (10) and expression markers such as GFP, RFP and lacZ.

- **Entity Dictionary:** upload a user-defined dictionary of collected entity names. The dictionary can contain synonyms and database-specific IDs, allowing data integrity checks and seamless integration of the results with the parent database. We generated a dictionary of FlyBase gene symbols, gene names and gene symbol and name synonyms based on the 'FB_2013_05 release fb_synonym_fb_2013_05.tsv.gz' file available from the files download page on the FlyBase Web site (11).
- **Meta Information:** define a list of checkboxes for document triage, e.g. whether the article contains human disease mentions or information on a new transgene. We generated checkboxes for all the FlyBase triage flags, so the annotation of the tagtog corpus could be used directly in the FlyBase curation.
- **Annotatable material:** select the sections of the full-text articles that can be annotated and trained on. The annotation of captions from figures and images can be decided independently: 'always', 'never' or 'section-dependent'. For this project, we annotated the title, abstract, materials and methods, results and figure legends. We did not annotate gene mentions in the introduction or the conclusion/discussion sections, as per FlyBase curation rules.

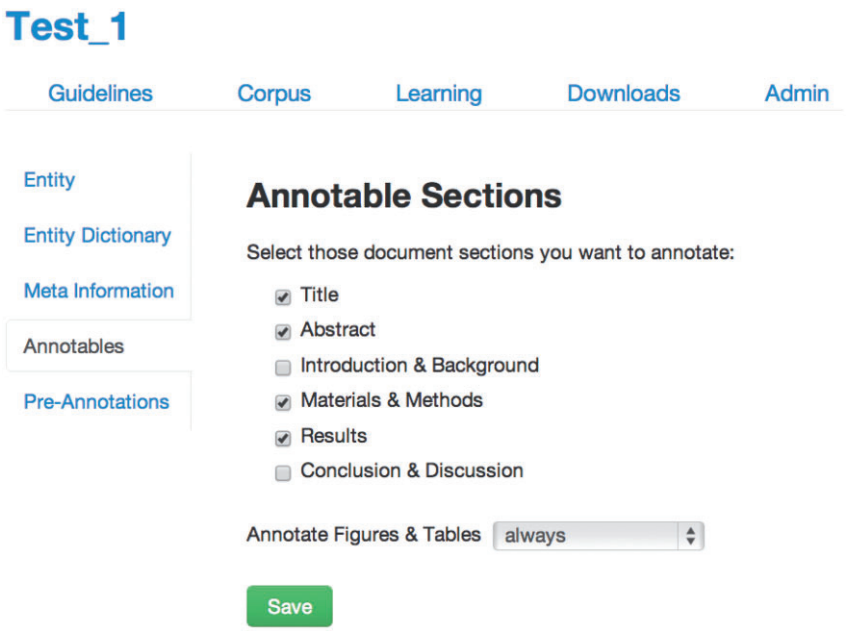


Figure 2. Annotation guidelines.

- **Pre-Annotations:** users can activate or deactivate this feature. Pre-annotations are annotations that are automatically generated within an individual document when a user adds or removes an annotation (i.e. selects or deselects a word). These automatic annotations are generated as follows: if a user selects the entity 'X', in the same document all mentions of 'X' will be pre-annotated and assigned to the same entity class. The converse is true for deselections. Note that the automatic pre-annotations are not machine learning-based but simple matches of equal strings. The pre-annotations are marked with a special flag and have to be validated or removed by the user before the containing article can be used for training.

The machine learning component of tagtog

A core defining characteristic of the tagtog system is that the users can choose the entity class to annotate, such as genes, Gene Ontology terms or diseases. The system boasts a general-purpose named entity recognizer implemented with conditional random fields (CRFs) (12). For the biomedical domain, the CRFs are trained with common features used in previous systems. However, in contrast to best performing methods like AIIAGMT (13), which use the aggregation of various CRF models, we use one sole backward model. This results in a slightly lower performance but has the benefit of an increased speed, which is essential in a user-interactive application. The recognizer can be customized to the prediction task at hand by means of user feedback and by using a dictionary of entity terms. The system can also be expanded with new machine annotators via plug-ins to enable annotation of diverse classes and domain languages within the same document. If desired, the machine learning component of tagtog can be turned off to allow biocurators to use the tagtog interface exclusively for manual curation.

Defining the project corpus

Every project in tagtog manages a corpus of documents, which can be uploaded either individually or in batches. The system's internal parser recognizes the documents' sections, subsections, figures, tables and some additional meta-information such as the paper's original uniform resource locator (URL). The project corpus can be augmented progressively as the user sees fit. Currently, documents are placed in two folders, the 'pool' folder, where most documents are placed, and the 'gold' folder, where a smaller set of manually annotated documents is used exclusively for the evaluation of the machine learning methods' performance. Only the documents in the pool folder can be used for training.

Generating the FlyBase corpus

To date, FlyBase curators have manually annotated 451 full-text articles using the tagtog interface. The PLOS

journal collection was chosen for document sampling because PLOS makes all their research papers fully available for text mining (14), and the PLOS journal collection covers many aspects of Drosophila research. All sampled papers are from between 2011 and 2013. The following document sections were annotated: title, abstract, results, materials and methods and figure and table legends. The paper annotations have been used to iteratively train the machine learning component of tagtog. So far, we have performed three annotation and benchmark iterations. In the first two iterations, annotations were done manually by a sole curator and automatically by the system. In the third iteration, all five FlyBase curators annotated papers manually. All the manual annotations and corrections were performed using tagtog's document editor interface.

Iteration 1: a sole curator (P. McQuilton) manually annotated a training set of 20 articles, representative of the Drosophila-related papers found in PLOS journals. The number of 20 'seed' articles was chosen based on best practices by previous experiments on active learning (15). We searched the PLOS Web site using the term '*Drosophila melanogaster*' from 2011 onward and then randomly selected 20 articles that had been already annotated and incorporated into the FlyBase database. Trained with these documents, the system was applied to predict gene mentions in an unlabeled validation set of 99 articles. The curator then went through the validation set and corrected, added or removed the predicted annotations, when appropriate. Mismatched annotations between the original predictions and the revised annotations were counted as errors.

Iteration 2: the two sets of papers used in Iteration 1 were united to form a training set of 119 articles. For evaluation, the user manually annotated a test set of 20 new articles (which we will refer to as the 'Gold Standard'. The system was retrained on the 119 articles and benchmarked against the 20 Gold Standard articles. In contrast to Iteration 1, prediction errors could be compared directly against the test set.

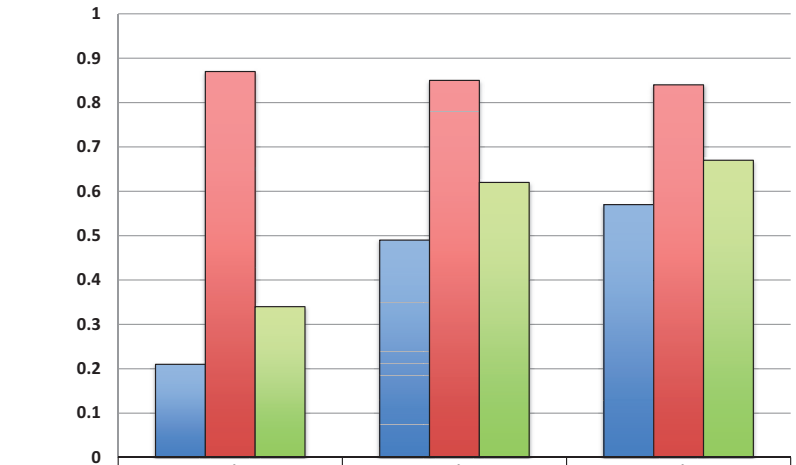
Iteration 3: the previous two sets, plus a further 312 papers curated by five different FlyBase curators, were combined to form an annotated corpus of 451 fly-related papers. These papers were used to retrain tagtog before the assessment on the Gold Standard set (20 papers).

Measuring performance on the FlyBase corpus

We used standard NER evaluation measures to benchmark performance, namely, precision (P), recall (R) and F1 measure (F1). Precision measures the percentage of correct predictions, i.e. the number of correct predictions divided by all predictions. Recall measures the percentage of correctly identified entities, i.e. the number of correctly identified entities divided by all entities present in the test document. There is typically a trade-off between precision and recall; F1 averages the two into one sole measure. More precisely, F1 is the harmonic mean between precision and recall. Only exact matches between the 'tagtog' predictions and the test annotations are counted as correct, i.e. the predictions have to match the exact word boundaries [for example, 'Su(H)' but not 'Su(H) protein']. Two types of counts were considered: (i) unique entities on a document basis. That is, for a test entity X in a document, the predictions are right if at least one mention of that entity can be identified in that document, wrong otherwise (for example, at least one mention of the gene 'dpp' is correctly identified, no matter whether other mentions may be missed). Equivalently, all unique entities identified by the predictions but not present on the test annotations are counted as errors. (ii) All entity mentions for all documents. That is, for all entity mentions, matching predictions and test annotations are counted as correct, whereas mismatched mentions, either false-positive findings or false-negative findings, are counted as errors (so in this case, three correct mentions of 'dpp' can be identified, while one mention is missed and recorded as a false negative). Note that for testing, only the annotatable sections defined by the curator are compared.

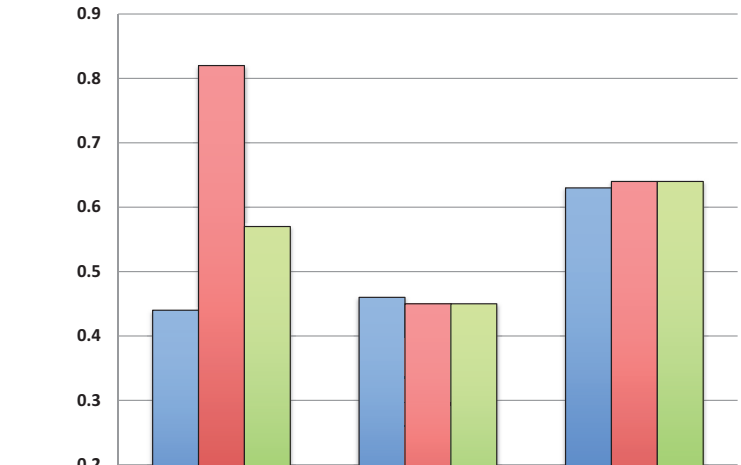
Figure 3 shows the entity recognition performance for all entity mentions in a paper, i.e the ability of tagtog to identify the presence of a gene mention, either as a symbol or name. The figure shows that the performance has steadily improved (taking the F1 measure) in proportion to the corpus size. The same performance improvement behavior is seen for unique entity recognition (Figure 4), that is, the ability to identify the presence of a gene at least once in a paper. In this case, however, we found a large reduction in precision performance from Iteration 1 ($P = 0.82$) to Iteration 2 ($P = 0.45$). We observed numerous false-negative findings that were repeated only once in the text, examples: 'BamH1' in 'journal.pgen.1003042' or 'oskar' in 'journal.pgen.1003079'. False-negative findings can significantly impact performance of unique entities, but leave the performance of all mentions mostly unaffected if the unique false-negative findings represent a small fraction of the total number of mentions. Nevertheless, in Iteration 3, both the precision and the recall for unique entities increased considerably ($P = 0.64$ and $R = 0.63$).

The final number of 451 papers consists of a test set of 20 manually annotated documents plus a training set of 431 documents, which combine manual and automatic annotations (that have subsequently been manually validated). We have deposited this corpus in the BioC



	Iteration 1	Iteration 2	Iteration 3
Recall	0.21	0.49	0.57
Precision	0.87	0.85	0.84
F1 measure	0.34	0.62	0.67

Figure 3. Entity recognition performance over all three corpora sizes.



	Iteration 1	Iteration 2	Iteration 3
Recall	0.44	0.46	0.63
Precision	0.82	0.45	0.64
F1 measure	0.57	0.45	0.64

Figure 4. Unique entity recognition performance over all three corpora sizes.

format (7) at the BioC repository (http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/BioC/) for use by other text-mining groups. We believe the corpus to be the largest and the most complete gene mention annotation set in full-text articles currently available.

The BioCreative interactive annotation task challenge—curation time

Within the framework of the BioCreative IV workshop, the purpose of the interactive annotation task (IAT) was to ascertain the possible benefit in terms of curation effort of machine-assisted annotation versus manual annotation. The task for tagtog was divided as follows:

1. Manual annotation: using the tagtog interface, a biocurator manually annotated a set of 20 documents with an entity class of her choice. The machine learning component of tagtog was consequently trained on the first manual set and provided automatic annotations for a second set of 20 documents.
2. Assisted annotation: using the tagtog interface, the biocurator reviewed and corrected, where appropriate, the automatic predictions of the second set of 20 documents.

Curation time was measured for both subtasks, and the results were compared. Two biocurators participated in the task, Mary Schaeffer from MaizeDB (first) and Ritu Khare from NCBI (second):

- The first biocurator chose to annotate maize-related genes and uploaded a self-defined dictionary of terms. She is an expert in this kind of annotation. A total of 6 h and 34 min was taken for the manual annotation and 4 h and 5 min for the assisted annotation. This indicates a reduction in curation time of ~1.6-fold.
- The second biocurator chose to annotate Drosophila gene names and symbols and uploaded the same dictionary as used with the FlyBase corpus. The second curator is not an expert in this kind of annotation. She spent 9 h and 19 min for the manual annotation and 4 h and 49 min for the assisted annotation. This indicates a reduction in curation time of ~1.9-fold.

Conclusions

We have shown that tagtog can be used successfully to annotate Drosophila gene symbols and names. We have also shown that the accuracy of these annotations increases with the size of the training corpus. In addition, we have shown that tagtog-assisted NER can reduce overall curation time. This gradual improvement in accuracy, combined with the shortening of curation time by 1.6- to 1.9-fold compared with completely manual curation, illustrates the benefit of including text-mining techniques, such as tagtog, in curation. To our knowledge, these preliminary results represent one of the first NER evaluations with a substantial amount of full-text articles in the biomedical field.

Given the encouraging nature of the curation time experiments, we plan to expand our analysis of curation with tagtog to assess whether the increase in curator speed is due to familiarity with the tool or assisted annotation. These experiments have also shown that tagtog can be used to annotate gene symbols from species outside of Drosophila, such as maize.

In future work, we will check for the presence of repeated entities between documents that could bias the NER evaluation between iterations and assess inter-annotator agreement between the five FlyBase curators to allow performance benchmarking. NER with full-text articles is understood to be considerably more difficult than for abstracts (16, 17), and although we have not specialized the machine learning methods used here for Drosophila gene mention extraction, we are pleased with the level of performance. The continuous learning of tagtog is designed to generate cheaper (in terms of manual curation effort) training data, by taking advantage of semiautomatic annotation. We will continue to add to the FlyBase corpus, with the aim of increasing NER accuracy and the potential incorporation of tagtog (or the output from tagtog) into our genetic literature curation pipeline.

In this article, we have illustrated how tagtog-assisted annotation can benefit manual curation from the literature. We have shown how the identification of *D. melanogaster* gene symbol and name mentions has gradually improved with more training data and user feedback. This illustrates the adaptability of the tagtog system to the specific curation requirements of the user, and there seems to be a potential for further improvement in NER performance. Thanks to our participation in the BioCreative IV IAT challenge, we have been able to achieve promising results in the reduction of curation time through the use of tagtog-assisted curation compared with manual gene mention extraction. As a result of our experiments, we have generated the FlyBase corpus, one of the largest corpora of full-text articles with entity annotations in the field of biomedical text mining. We have made this available in BioC format for use by the text-mining community.

Author contributions

J.M.C. and P.M. devised the experiments and wrote the article. P.M., S.M., L.P., R.S. and G.M. annotated the

Database, Vol. 2014, Article ID bau033

corpus and provided feedback on the tagtog interface. J.M.C. developed tagtog.

Acknowledgements

The authors would like to thank the BioCreative initiative for the opportunity to participate in the interactive annotation task and our interactive annotators Mary Schaeffer and Ritu Khare for taking the time to thoroughly test the tagtog system. They would also like to thank all members of FlyBase for their helpful comments and suggestions on the article. The current FlyBase Consortium comprises: William Gelbart, Nicholas H. Brown, Thomas Kaufman, Kathy Matthews, Maggie Werner-Washburne, Richard Cripps, Kris Broll, Madeline Crosby, Gilberto dos Santos, David Emmert, L. Sian Gramates, Kathleen Falls, Beverley B. Matthews, Susan Russo, Andrew Schroeder, Susan E. St. Pierre, Pinglei Zhou, Mark Zytkovicz, Boris Adryan, Helen Attrill, Marta Costa, Steven Marygold, Peter McQuilton, Gillian Millburn, Laura Ponting, Raymund Stefancsik, Susan Tweedie, Josh Goodman, Gary Grumbling, Victor Strelets, Jim Thurmond and Harriett Platero.

Funding

NHGRI/NIH (HG000739 to W. Gelbart, Harvard University, PI; N. H. Brown, University of Cambridge, coPI); private funding (to J.M.C.). Funding for open access charge: the National Human Genome Research Institute, the National Institutes of Health [P41 HG00739], and tagtog.

Conflict of interest. None declared.

References

1. Bunt,S.M., Grumbling,G.B., Field,H.I. *et al.* (2012) Directly e-mailing authors of newly published papers encourages community curation. *Database (Oxford)*, 2012, bas024.
2. St Pierre,S.E., Ponting,L., Stefancsik,R. *et al.* (2014). FlyBase 102—advanced approaches to interrogating FlyBase. *Nucleic Acids Res.*, 42, D780–D788.
3. PLOS journal homepage. http://www.plos.org/ (February 2014, date last accessed)
4. Culotta,A., Kristjansson,T., McCallum,A. *et al.* (2006) Corrective feedback and persistent learning for information extraction. *Artif. Intell.*, 170, 1101–1122.
5. 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 (Oxford)*, 2013, bas056.
6. NCBI Journal Publishing Tag Set. http://jats.nlm.nih.gov/publishing/ (February 2014, date last accessed)
7. BioMed Central Format. http://www.biomedcentral.com/about/xml/ (February 2014, date last accessed)
8. Journal Article Tag Suite. http://jats.nlm.nih.gov/ (February 2014, date last accessed)
9. Comeau,D.C., Islamaj Doğan,R., Ciccarese,P. *et al.* (2013) BioC: a minimalist approach to interoperability for biomedical text processing. *Database (Oxford)*, 2013, bat064.
10. Brand,A.H. and Perrimon,N. (1993) Targeted gene expression as a means of altering cell fates and generating dominant phenotypes. *Development*, 118, 401–415.
11. FlyBase Precomputed Files. http://flybase.org/static_pages/downloads/bulkdata7.html
12. Lafferty,J.D., McCallum,A. and Pereira,F.C.N. (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA. pp. 282–289.
13. Hsu,C.H., Chang,Y.M., Kuo,C.J. *et al.* (2008) Integrating high dimensional bi-directional parsing models for gene mention tagging. *Bioinformatics*, 24, i286–i294.
14. PLOS Open Access Policy. http://www.plos.org/open-access/
15. Tomanek,K. and Hahn,U. (2009) Semi-supervised active learning for sequence labeling. In: *Annual Meeting of the Association of Computational Linguistics 2009*. Suntec, Singapore. http://clair.eecs.umich.edu/aan/paper.php?paper_id=P09-1117.
16. Larry,S., Lorraine,T., Rie,A. *et al.* (2008) Overview of BioCreative II gene mention recognition. *Genome Biol.*, 9 (Suppl 2), S2.
17. Zhiyong,L., Hung,K.Y., Chih,W.H., *et al.* (2011) The gene normalization task in BioCreative III. *BMC Bioinformatics*, 12 (Suppl 8), S2.

2.3 References

Andorf, C. M. et al. (2016). "MaizeGDB update: new tools, data and interface for the maize model organism database". In: *Nucleic Acids Res* 44.D1, pp. D1195–201. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkv1007. URL: https://www.ncbi.nlm.nih.gov/pubmed/26432828.

Bunt, S. M. et al. (2012). "Directly e-mailing authors of newly published papers encourages community curation". In: *Database (Oxford)* 2012, bas024. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bas024. URL: https://www.ncbi.nlm.nih.gov/pubmed/22554788.

Cohen, K. B. et al. (2010). "The structural and content aspects of abstracts versus bodies of full text journal articles are different". In: *BMC Bioinformatics* 11, p. 492. ISSN: 1471-2105 (Electronic) 1471-2105 (Linking). DOI: 10.1186/1471-2105-11-492. URL: https://www.ncbi.nlm.nih.gov/pubmed/20920264.

Gramates, L. S. et al. (2017). "FlyBase at 25: looking to the future". In: *Nucleic Acids Res* 45.D1, pp. D663–D671. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1016. URL: https://www.ncbi.nlm.nih.gov/pubmed/27799470.

Mao, Y. et al. (2014). "Overview of the gene ontology task at BioCreative IV". In: *Database (Oxford)* 2014. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bau086. URL: https://www.ncbi.nlm.nih.gov/pubmed/25157073.

Chapter 3

nala: extraction of genetic variations mentions written in natural language

3.1 Preface

Genetic variations are of vital importance to understand and consequently treat diseases. Descriptions of mutations and their experimentally-observed effects are deposited in the literature. Several automatic methods tried in the past to recognize such mentions, but primarily only focused on simple description forms (e.g. "E6V"). Complex natural descriptions (e.g. "glutamic acid was substituted by valine at residue 6") remain largely untapped

In this work, we first studied the impact of natural language (NL) mutation mentions. That is, how often such descriptions are used by authors in scientific manuscripts. We created three independent corpora semi-automatically annotated with the *tagtog* tool (Cejuela et al. 2014). The first two corpora, *IDP4 and nala_known*, gathered (as was customary in previous corpora) full-text and abstract articles listed in heavily-indexed (i.e. well understood) sources such as UniProtKB (The UniProt Consortium 2017) or *dbSNP* (Sherry et al. 2001). The third corpus, *nala_discoveries*, was based on a comprehensive, unbiased PubMed search of mutation-related, most recent publications in the highly-renowned journals, Nature, Science, and Cell. Altogether, the three new corpora constituted the largest source of labeled mutation mentions (5660) and thwarted previous corpora (all combined, 2933 mentions). In the previous and new heavily-indexed articles, *SetsKnown*, 28%-36% of documents had at least one NL mutation description. In recent publications, *nala_discoveries*, the number was substantially higher: 67%-77%. All these genetic variations would be missed by existing text mining solutions.

Consequently, we developed a new automatic method to recognize all sorts of mutation types (e.g. SNPs/SAVs, insertions, deletions, repetitions, or large chromosomal rearrangements) written in simple form or complex NL forms. The new method, *nala*, combined probabilistic combined conditional random fields (CRFs) (Lafferty, McCallum, and Pereira 2001) with word embedding features (Mikolov et al. 2013). The word embeddings were

learnt unsupervised from the entire PubMed, using *word2vec* with the neural network architecture *CBOW* (continuous bag of words). These unsupervised features accounted for the biggest performance improvements in mutation mention recognition. Moreover, the new method and the *nala_known* corpus were developed in parallel in an *active learning* setting: articles with automatic predictions with low confidence or erroneous, were purposely selected for annotation, precisely to learn those. In *SetsKnown* articles, the new method *nala* performed consistently equal or better than previous methods. In *nala_discoveries* (latest publications), the *nala* method did not miss any of the found mutation mentions by other methods, and discovered 33% of the mentions uniquely. Further, *nala* was the only method to identify NL, long mentions of variations.

The implementation of methods and analysis results were done by me, Aleksandar Bojchevski, and Carsten Uhlig. The study design was conceived by me and Burkhard Rost. The annotations were done by me, Aleksandar Bojchevski, Carsten Uhlig, Rustem Bekmukhametov, Sanjeev Kumar Karn, and Shpend Mahmuti. Ashish Baghudana provided extra software implementations. Ankit Dubey provided background research. Venkata P. Satagopam contributed with overall guidance and proofreading. The manuscript was drafted by me and Burkhard Rost.

3.2 Journal article. Cejuela *et al.*, *Bioinformatics* 2017

Starts next page.

Bioinformatics, 2017, 1–7
doi: 10.1093/bioinformatics/btx083
Advance Access Publication Date: 13 February 2017
Original Paper

OXFORD

Data and text mining

nala: text mining natural language mutation mentions

Juan Miguel Cejuela[1,2,*], Aleksandar Bojchevski[1,2], Carsten Uhlig[1], Rustem Bekmukhametov[1,3], Sanjeev Kumar Karn[1,4], Shpend Mahmuti[1], Ashish Baghudana[1,5], Ankit Dubey[1,6], Venkata P. Satagopam[7] and Burkhard Rost[1,8]

[1]TUM, Department of Informatics, Bioinformatics & Computational Biology – i12, Garching, Munich 85748, Germany, [2]TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Garching 85748, Germany, [3]Microsoft, WA 98008, Bellevue, USA, [4]Ludwig Maximilian University, 80538 Munich & Siemens AG, Corporate Technology, Munich 81739, Germany, [5]BITS-Pilani K. K. Birla Goa Campus, Goa 403726, India, [6]Concur (Germany) GmbH, Frankfurt am Main 60528, Germany, [7]Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, L-4367 Belvaux, Luxembourg and [8]Institute of Advanced Study (TUM-IAS) & Institute for Food and Plant Sciences WZW – Weihenstephan & New York Consortium on Membrane Protein Structure (NYCOMPS) & Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY 10032, USA

*To whom correspondence should be addressed.
Associate Editor: Jonathan Wren

Received on October 17, 2016; revised on January 13, 2017; editorial decision on February 6, 2017; accepted on February 8, 2017

Abstract

Motivation: The extraction of sequence variants from the literature remains an important task. Existing methods primarily target standard (ST) mutation mentions (e.g. 'E6V'), leaving relevant mentions natural language (NL) largely untapped (e.g. 'glutamic acid was substituted by valine at residue 6').
Results: We introduced three new corpora suggesting named-entity recognition (NER) to be more challenging than anticipated: 28–77% of all articles contained mentions only available in NL. Our new method *nala* captured NL and ST by combining conditional random fields with word embedding features learned unsupervised from the entire PubMed. In our hands, *nala* substantially outperformed the state-of-the-art. For instance, we compared all unique mentions in new discoveries correctly detected by any of three methods (SETH, tmVar, or *nala*). Neither SETH nor tmVar discovered anything missed by *nala*, while *nala* uniquely tagged 33% mentions. For NL mentions the corresponding value shot up to 100% *nala*-only.
Availability and Implementation: Source code, API and corpora freely available at: http://tagtog.net/-corpora/IDP4+.
Contact: nala@rostlab.org
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Genetic variations drive biological evolution. Yet, most mutations might harm (Rost, 1996; Rost *et al.*, 2003; Sawyer *et al.*, 2007). Experimental studies elucidating the effects of sequence variation remain precious and expansive. Today, the important results from such studies are still published in papers. Repositories, such as OMIM, rely primarily on labor-intensive and time-consuming expert curation. Searching PubMed with relevant keywords (http://1.usa.gov/1rCrKwR) brought up >1M articles; most of those (>630K) for variation in human. An equivalent search of UniProtKB/Swiss-Prot

(Boutet *et al.*, 2016; UniProt, 2015) revealed ~13K indexed publications, and the professional version of the Human Gene Mutation Database (HGMD) (Stenson *et al.*, 2003) listed ~179K mutations. These numbers sketch the immense information gap between literature and database annotations (Jimeno and Verspoor, 2014a,b, Database). Despite two decades of high-level efforts to increase the incentive for authors to link their findings to databases, this gap is likely to expand even more rapidly in the future. Instead of requiring administrative overhead, the text mining of free literature pursues a solution that could scale and substantially narrow the gap (Krallinger *et al.*, 2008).

Mutation mentions refers to the format used to report experimental results for sequence variants. Mining mutation mentions is referred to as *named-entity recognition* (NER). We focused on the task to recognize and parse text fragments such as the following two equivalent mutation mentions: 'glutamic acid was substituted by valine at residue 6' or 'p.6E > V'. The two differ only in their syntax: the first is written in natural language (NL), the second follows a standardized format (ST).

Existing extraction methods primarily target simple and standardized mutation mentions. MutationFinder (MF) (Caporaso *et al.*, 2007a,b) uses a large set of regular expressions (*regexes*) to recognize single nucleotide or amino acid variants written in simple ST form (e.g. 'E6V') and slightly more complex semi-standard (SST) form (e.g. 'Glu 6 to Val' or 'glutamic acid for valine 6'). *SETH* (Thomas *et al.*, 2016) recognizes other short sequence variations such as insertions and deletions (*indels*, e.g. 'c.76_77insG' and 'c.76delA', resp.) by implementing a formal grammar and regexes that cover recommended, deviations and deprecated cases of the HGVS nomenclature (den Dunnen *et al.*, 2016). The HGVS nomenclature aims to frame mutation mentions in a canonical *normalized* language (e.g. the complete form 'p.Glu6Val' is preferred over alternatives). *tmVar* (Wei *et al.*, 2013) has introduced probabilistic methods and recognizes ST mentions for a large variety of variant types: point variants (SNVs: Single Nuclear Variants, SAVs: Single Amino acid Variants), structural variations (insertions, deletions, frameshifts: e.g. 'p.(Arg97fs)', duplications: e.g. 'c.76dupA'), and *rsids* (reference SNP ID numbers, e.g. 'rs206437', i.e. dbSNP accession numbers (Sherry *et al.*, 2001)). None of these three methods appear to extract genetic markers (e.g. 'D17S250') nor large-scale mutations, i.e. variations of regions longer than a few nucleotides or amino acids (e.g. 'TP73Δex2/3' or 'abrogated loss of Chr19'). Existing methods are reviewed in detail elsewhere (Jimeno and Verspoor, 2014a,b, F1000Res.; Nagel *et al.*, 2009). Mapping the variant E6V to a particular sequence, e.g. that of hemoglobin S in human with the SWISS-PROT identifier *hbb_human* and relating it to sickle cell anemia (SKCA) and finally identifying that the variants is actually at position 7 in the sequence, i.e. should have been named E7V (p.Glu7Val), are all essential steps toward 'parsing the meaning' of the annotation. We ignored these mapping problems in this work. Instead, our work focused on presenting the first comprehensive study of the significance of natural language mutation mentions (e.g. 'in-frame deletion of isoleucine 299'). Our new method completed the picture by recognizing different mutation types (for both genes and proteins) written in simple form or complex natural language.

2 Materials and methods

2.1 Classification of mutation mentions: ST, SST and NL

There is no single reliable classification of natural language (NL) or standard (ST) mutation mentions. Some annotators might consider *'alanine 27 substitution for valine'* as NL because it does not follow the standard HGVS nomenclature. Others might consider it as standard or semi standard (SST) because simple regexes might capture this mention. Previous mutation extraction methods primarily used regexes and did not capture long mutation mentions.

As an operational definition, we considered any long mention that was not recognized by previous methods as NL, any mention that resembled the HGVS nomenclature as ST, and any mention in between as SST. We defined the following if-else chain algorithm to capture this idea: given a mutation mention, if it matches custom regexes or those from tmVar, then it is ST; else if it has 5 or more words or contains 2 or more English-dictionary words, then it is NL; else if it contains 1 English-dictionary word, then it is SST; else it is ST (examples in Table 1). Our custom regexes matched one-letter-coded mentions such as 'p.82A > R' or 'IVS46: del T -39...-46' (Supplementary Table S9). The collected *tmVar* regexes were used by the authors (Wei *et al.*, 2013) as features of the *tmVar* probabilistic model and as post-processing (PstPrc) rules.

2.2 Evaluation measures

We considered a named entity as successfully *extracted* if its *text offsets* (character positions in a text-string) were correctly identified (tp: *true positive*). We considered two modes for *tp*: *exact* matching (two entities match if their text offsets are *identical*) and *partial* matching (text offsets *overlap*). Any other prediction was considered as a *false positive* (fp) and any missed entity as a *false negative* (fn). Partial matching is more suitable to evaluate NL mentions lacking well-defined boundaries. For instance, in finding *'[changed conserved] glutamine at 115 to proline'*, we did not distinguish solutions with and without the words in brackets, because we focused on the extraction of the mention not on that of additional annotations (here '*conserved*'). We computed performance for all cases and for the subclasses (ST, SST and NL). A test entity of subclass X was considered as correctly identified if any predicted entity matched. We then used the standard evaluation measures for named-entity recognition, namely, *precision* (P: $tp/tp+fp$), *recall* (R: $tp/tp+fn$) and *F-Measure* (F: $2*\left(P*R\right)/(P+R)$). Within a corpus, we computed

Table 1. Classification of mutation mentions

Class	*Examples*	*MF*	*SETH*	*tmVar*
ST	• Q115P; Asp8Asn; 76A>T	yes	yes	yes
	• c.925delA; g.3912G>C; rs206437	no	yes	yes
	• c.388 + 3insT	no	no	yes
	• delPhe1388; F33fsins; IVS3(+1); D17S250;	no	no	no
	• TP73Δex2/3			
SST	• 3992-9g–>a mutation; codon 92, TAC–>TAT	no yes	no yes	yes no
	• Gly 18 to Lys; leucine for arginine 90	no	no	no
	• G643 to A; abrogated loss of Chr19			
NL	• glycine to arginine substitution at codon 20	yes no	yes no	no no
	• glycine was substituted by lysine at residue 18	no	no	no
	• deletion of 10 and 8 residues from the N- and C-terminals			

Note: Examples of mutation mentions of increasing level of complexity as found in the literature (*ST*: *standard*; *SST*: *semi-standard*; *NL*: *natural language*). The columns *MF*, *SETH* and *tmVar* indicate if the methods MutationFinder, SETH and tmVar, respectively, recognize the examples listed.

the StdErr by randomly selecting 15% of the test data without replacement in 1000 (n) bootstrap samples. With $<x>$ as the overall performance for the entire test set and x_i for subset i, we computed:

$$StdErr = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \langle x \rangle)^2} = \frac{\sigma}{\sqrt{n}} \quad (1)$$

Across corpora, we did not merge documents. Rather, we computed the mean of P, R and F between the considered corpora, and computed the StdErr of the mean without subsampling.

2.3 Previous corpora

Some well-known corpora annotate mutation mentions and specific text offsets, including: *SETH* (Thomas *et al.*, 2016), *tmVar* (Wei *et al.*, 2013) and *Variome* (Verspoor *et al.*, 2013). All corpora contain different mutation types, including SNPs, frameshifts, or deletions (primarily in ST or SST forms). *SETH* and *tmVar* annotated abstracts, *Variome* full-text articles. The *Variome* corpus annotated many vague mentions (e.g. '*de novo* mutation' or 'large deletion'). With *Variome120* we referred to a *Variome* subset of position-specific variants with 118 mentions as described earlier (Jimeno and Verspoor, 2014a,b, F1000Res.) plus two new annotations with reference to both a DNA and a protein mutation.

2.4 Three new corpora: *IDP4, nala* and *nala_discoveries*

We annotated three new corpora (*IDP4*, *nala* and *nala_discoveries*) at different times and with slightly different objectives. These solutions substantially enriched the *status quo*. All three were annotated with the tool *tagtog* (Cejuela *et al.*, 2014). The differences were as follows.

2.4.1 IDP4 corpus

We introduced the *IDP4* corpus to offer an unbiased representation of mutation mention forms (NL in particular). Previous corpora focused on ST or SST mentions. We annotated the entities *Mutation*, *Organism* and *GGP* (gene or gene product), as well as, relations between *GGP* and both *Mutation* and *Organism*. We included abstract-only and full-text documents. Documents were selected in four steps. (1) Include particular organisms/sources (*Homo sapiens, Arabidopsis thaliana, Drosophila melanogaster, Caenorhabditis elegans, Schizosaccharomyces pombe, Saccharomyces cerevisiae, Mus musculus, Rattus norvegicus* and *HIV*). (2) Collect the PubMed identifiers linked from SWISS-PROT (Boutet *et al.*, 2016) that cite the keywords variation or mutagenesis. (3) Accept all abstracts that contain any of five keywords (*mutation, variation, insertion, deletion, SNP*). (4 optionally) Retrieve full-text articles through keyword *open access* (on PubMed Central).

Our method and thus our annotation guidelines needed mutation mentions with three components: (1) W (word): a clear word or pattern giving the variant and its type (W is binary, i.e. present or not), e.g. W = yes as in 'His72 *substitution to* Arg' or '24bp *duplication* of ARX exon 2'. (2) L (letter): giving the mutated nucleotides or residues (L is binary, i.e. present or not), e.g. L = yes as in 'delta Phe581' and L = no as in 'deletion at pos. 581'. (3) P (position): giving the sequence location of the variation (P has three values: exact, vague, or no, i.e. not applicable), e.g. P = exact as in 'Tyr838 mutation' or 'Del 1473-IVS16(+2)' and P = vague as in 'placed immediately downstream of I444' or 'at the carboxyl end'.

We annotated two cases: (1) W = yes, L = yes, P = yes|vague, e.g. 'p.Phe54Ser', 'Arg-Thr insertion between 160 and 161 residues', or '(499)leucine (TTA) to isoleucine (ATA)'; (2) W = yes, L = no, P = yes, e.g. 'point mutation at amino acid 444', 'SNPs affecting residues, 282, 319 and 333'. The rationale was that we could assign to the missing nucleotide/residue the unknown value *X*. We also annotated total gene knockouts ('Δ/Δ'), deletions of subparts ('deleted C1 domain'), or deletions of larger regions ('deletions of chromosome 9p22.3'). We considered those positions as specific. Moreover, we annotated rsids.

We measured the agreement between annotators (F-Measure of the inter-annotator agreement: *F_IAA*) as proxy for the consistency of the annotations. Four annotators participated. Across 53 overlapping documents, for IDP4 we observed F_IAA = 91 for all mutation mentions and F_IAA = 77 for NL mentions. In total, the *IDP4* corpus collected 157 documents (72 full text + 85 abstracts) with 3337 mutation annotations: 3113 ST mentions (93%), 198 NL (6%) and 26 SST (1%).

2.4.2 nala corpus

We introduced the *nala* corpus to expand the amount of NL mutation mentions necessary for the training of probabilistic methods. No previous corpus tagged enough (Results) (Ravikumar *et al.*, 2012). We annotated only abstracts for they contained higher densities (number of mentions/number of words) of NL mentions than full articles. In particular, the *IDP4*, *Variome* and *Variome120* corpora contained more NL mentions per word in abstracts than in full texts (ratios: 5.5, 1.6 and 3.8). We selected documents as for the *IDP4* corpus but applied *active learning* to simultaneously build corpus and method (details below). The *nala* corpus consisted of two disjoint sets: *nala_training* and *nala_known*. The latter 'blind' set with 90 randomly chosen abstracts (15% of the entire *nala* corpus) was used only to test. We stopped adding abstracts to this test set when the standard error estimate plateaued. Moreover, *nala_known* contained 8 documents (9% of test) without any annotation, i.e. no mutation mentions, to effectively probe the precision of methods.

Annotating NL mentions strictly following our *IDP4* corpus guidelines was more challenging. For example, mutation positions were often vague and/or referenced indirectly in other sentences than the variant and often in different paragraphs. In particular, we relaxed the rules more for insertions and deletions, e.g. '2-bp deletion in exon 6', 'somatic 16-bp deletion', or 'in-frame insertion of 45 nucleotides'. Another unique feature of the *nala* corpus was the annotation of genetic markers. To limit the workload, for the *nala* corpus we refrained from annotating organisms or GGP terms. Only to ease the reading of mutation mentions, we used the GNormPlus tagger (Wei *et al.*, 2015) to automatically annotate gene/protein terms.

Three experts annotated *nala*; their agreement over 30 documents was F_IAA = 95 for all mutation mentions and F_IAA = 89 for NL. The *nala* corpus collected 591 abstracts with 2108 mutation annotations. Despite the explicit focus on NL mentions, ST mentions still dominated (presumably because they are easier to annotate): 1097 ST (52%) versus 841 NL (40%) and 170 SST (8%). As a result, the *nala_known* set benchmarked both ST and NL mentions (SST mentions were underrepresented).

2.4.3 nala_discoveries corpus

We introduced another novel corpus, *nala_discoveries*, to gauge automatic tagging of papers with 'new discoveries'. The idea is best explained in comparison to our generic *nala* corpus: there we picked the PubMed articles beginning from identifiers of genes and proteins that had already been described experimentally and annotated in SWISS-PROT (Boutet *et al.*, 2016). We had not realized how crucial this constraint was until we created a new corpus just before

submitting the manuscript. The usage of previously-indexed articles and knowledge has been common practice, e.g. for SNPs indexed by *dbSNP* or HGVS-compliant mentions (*SETH* corpus), disease- and mutation-specific MeSH terms indexed by PubMed (*tmVar* corpus), mutation-specific citations indexed by SWISS-PROT (*IDP4* and *nala*). Only the *Variome* corpus directly searched PubMed, but it was limited to three Lynch syndrome genes. For *nala_discoveries*, we found all articles in PubMed using the keyword *mutation* and published between 2013 and 2016 in the journals Nature, Science and Cell, without further filtering (exact search: http://bit.ly/2aHthKP). To limit the workload, we randomly selected abstracts with at least one mutation mention (any form) and stopped at 60 abstracts with *at least one NL* mention. We applied the guidelines used for *IDP4* and *nala*. Compared to other corpora, we found more large-scale mutations (e.g. chromosomal translocations) and significant differences in the semantics of mutation mentions. The numbers for *nala_discoveries* were: 78 abstracts (18 with ST or SST mentions only) and 215 mutation annotations spanning 104 ST mentions (48%), 71 NL (33%) and 40 SST (19%). The corpus *nala_discoveries* effectively benchmarked all mention classes (incl. SST) and was annotated by the same three annotators as the *nala* corpus.

2.5 New method: *nala*

The new method *nala* was based on conditional random fields (CRFs) (Lafferty *et al.*, 2001). Techniques for CRFs are amply described (Settles and Burr, 2004; Wei *et al.*, 2013; Wei *et al.*, 2015). We used the *python-crfsuite* implementation, a python binding of the *CRFSuite* C++ library (software URLs in Supplementary Table S10). We used our in-house implementation of the *tmVar tokenizer* (Wei *et al.*, 2013), but did not split tokens upon case changes at the sentence beginning ('The' not 'T'+'he'). We applied *BIEO* token labeling: tokens at the *beginning* of a mutation mention were labeled as *B*; continuing (*inside*) tokens as *I*; *ending* tokens as *E*; all other tokens (*outside* a mention) as *O*. For NL, *BIEO* outperformed our implementation of the 11 *tmVar* labels. We also included standard features such as token stems, word patterns, prefix and suffix characters, presence of numbers, or the word belonging to term dictionaries such as nucleotides, amino acids, or other common entities. We also added PstPrc rules such as fixing small boundary problems ('+1858C > T' not '1858C > T'). Finally, we introduced two optional post-processing (PstPrc) regex-based filters that can be switched on or off by users: 1) annotate rsids or not, and 2) annotate genetic markers or not.

Word embedding features (WE) contributed most to our new method. *WE* features had already helped in biomedical named-entity recognition (Guo *et al.*, 2014; Passos *et al.*, 2014; Seok *et al.*, 2016; Tang *et al.*, 2014). Specifically, we used neural networks with the CBOW architecture (continuous bag of words) (Mikolov *et al.*, 2013) and trained on all PubMed abstracts until mid 2015. We used window = 10 and dimension D = 100. Tokens were converted to lowercase and digits were normalized to 0. For each token, the vector of 100 real values was translated into 100 features. The real values were used as weights in the CRF features, e.g.: word_embedding[0]=0.00492302. In analogy to the optional PstPrc filters, users also have the option to run *nala* with WE features (default) or not (the features are not computed).

We built the *nala* corpus and method in parallel through iterative active learning (Fig. 1). We implemented a base version (*nala_1)* using the features from *tmVar* and trained on the *IDP4* corpus (*iteration_1* training set). For later iterations (*iteration_t*), we used the previous model (*nala_t-1*) and a high-recall set of regexes to select documents with non-ST mentions. We selected only documents

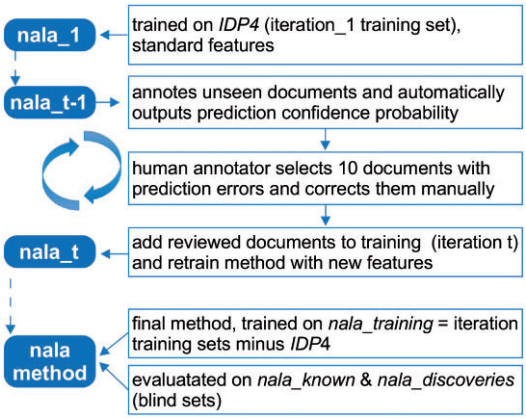


Fig. 1. *nala* method active learning process. Each blue box represents an iteration state of the *nala* method. The method and the iteration training sets are implemented in parallel. The previous iteration method (*nala_t-1*) is used to automatically annotate unseen documents. Selected documents with outstanding errors are reviewed manually and added to the iteration training set *t*. New features are evaluated in 5-fold cross validation and the method is re-trained with all previous sets (*nala_t*). At the end, the sum of iteration training sets without *IDP4* form the *nala_training* corpus. The final *nala* method is trained on *nala*_training (only) and evaluated against the *nala_known* and *nala*_discoveries corpora

with ≥1 NL mention. In each iteration, we arbitrarily selected ten documents. These were pre-annotated by *nala_t-1* and then posted to the *tagtog* annotation tool for expert review and refinement; the reviewed annotations were saved as *iteration_t*. In each iteration step, we trained through 5-fold cross-validation. Annotators selected documents with annotation errors (missing entities, wrong offsets, or false positives) to learn those. In the end, the merging of iteration sets without *IDP4* created the *nala_training* corpus. We trained the final method solely on *nala_training* (without using *IDP4* as training data), due to two reasons. Firstly, NL mentions were learned much better with *nala_training*. Secondly, ST mentions were learned better including *IDP4*, yet the small improvement did not justify the complexity of two separate models (ST and NL). We used *nala_known* and *nala_discoveries* only to evaluate the final method.

2.6 Methods for comparison

We compared *nala* with two state-of-the-art methods, namely *SETH* and *tmVar*. To run *SETH* locally, we slightly modified the original *scala* code to print out the results in *brat format*. To run *tmVar*, we used its official API. We could not benchmark the *tmVar* API on the *tmVar* test set, as it had been trained on this set. For each method, we evaluated its default and its *best* performance. To compute the *best* performance, we filtered out some test annotations and predictions originating from arbitrary annotation guidelines of the individual corpora. For example, the *best* performance of *tmVar* on the *SETH* corpus disregarded rsids; *tmVar* predicts rsids but the *SETH* corpus does not consistently annotate them (9 out of 69). Analogously, *nala* predicted many NL mentions not annotated in the *SETH*, *tmVar*, or *Variome120* corpora. Overall, we applied the two PstPrc filters (rsids and genetic markers) and the usage or not of WE features (only for *nala*). WE features improved the performance for NL mentions (details below) but without WE features *nala* did better on the ST-scoped corpora. For all methods, the difference between default and *best* performance was consistently and substantially larger than the standard

error within the corpus. This underlined the significance of annotation guidelines. Consequently, we reported (Results) the averages for default and best performance and their standard errors (individual results in Supplementary Tables S1–S5).

3 Results and discussion

3.1 Natural language (NL) mutation mentions important

The *Variome120* and *IDP4* corpora (no bias in mention forms) had much higher fractions of NL over ST or SST mentions (8% and 6%, respectively; Fig. 2, grayed out bars) than *SETH* (4%) and *tmVar* (2%). Removing repetitions, the fraction of unique NL mentions increased to 17% and 13% (Fig. 2, highlighted bars). The *Variome* corpus contained the largest fraction of SST mentions (53% with and 19% without repetitions). NL mentions dominated abstracts even more (12% in *Variome120* and 13% in *IDP4* with mention repetitions and 29% and 17% without repetitions). The *nala* corpus, introduced here, was built with a higher fraction of NL mentions (40% with repetitions and 49% without repetitions). All these corpora relied on well-annotated genes and proteins (indexed articles). In contrast, the *nala_discoveries* corpus randomly sampled abstracts without considering previous functional annotations (no previous indices). It contained the largest percentage of combined NL + SST mentions (52% with repetitions and 65% without repetitions).

How many experimental results will methods miss from the three corpora (*IDP4*, *Variome* and *Variome120*) that focus on ST or SST mentions? 28–36% of all abstracts contained at least one NL mention not in ST form (Table 2). The corresponding per-mention fractions were 13–27% (Table 2). For *nala_discoveries* the numbers were substantially higher: 67–77% (per-document) and 43–51% (per-mention).

3.2 New method *nala* performed top throughout

In our hands, the new method *nala* compared favorably with existing tools for extracting standard (ST) mutation mentions and significantly outperformed the status-quo for natural language (NL) mutation mentions (Fig. 3). This baseline was valid for all evaluations that we carried out. We found it more difficult to yield a single answer for the performance of *nala* (and from *nala* compared to other methods) because the performance depended crucially on the corpus. Each corpus has its own focus and bias. Which one best reflects what users expect?

We tried to simplify by grouping results into those for previously indexed mutations (*SetsKnown* corpora: *SETH*, *tmVar_test*, *Variome120* and *nala_known*; Supplementary Table S6) and those without prior knowledge (*nala_discoveries*; Supplementary Table S5). To establish the performance on well-annotated genes and proteins, the *SetsKnown* corpora might provide the least biased estimate: the *nala* method overall obtained F = 89 ± 3 compared to the highest performing competitor, i.e. *tmVar* with F = 87 ± 3 (Table 3). In contrast, the *nala_discoveries* corpus best established how well text mining works for new articles: the *nala* method reached F = 55 ± 7 compared to the highest performing competitors *SETH* and *tmVar* with F = 41 ± 10 (Table 3). Precision was very high for all methods on all evaluations and always lower than recall (for *nala* avg. on *SetsKnown* P = 87/R = 92; on *nala_discoveries* P = 90/R = 40). Thus, precision is a proxy for the performance on documents without mutation.

Table 2. Significance of NL mentions

	IDP4		Variome	Var.120	nala_discoveries		
Annotator*	(1)	(2)			(1)	(2)	(3)
Documents	30%	42%	22%	33%	78%	62%	77%
Mentions	14%	19%	6%	40%	52%	39%	49%

Note: Percentages of documents (3rd row) or mentions (4th row) that contain at least one NL (natural language) or SST (semi-standard) for which no ST (standard) mention exists in the same text. *Two different annotators were compared for the corpus *IDP4*; three different annotators were compared for the corpus *nala_discoveries*.

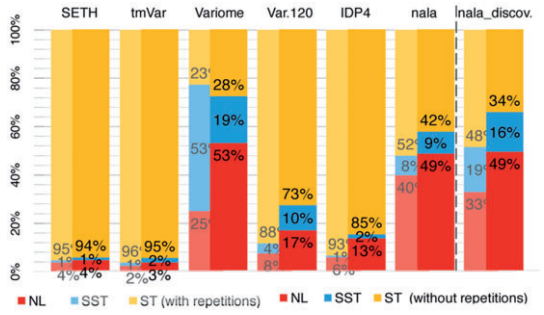


Fig. 2. Natural language (NL) mutation mentions important. What type of mutation mentions dominates annotated corpora that somehow sample the literature: standard (ST, e.g. E6V), semi-standard (SST), or natural language (NL)? Grayed out bars indicate counts with repetitions, full bars unique mentions (e.g. E6V occurring twice in the same paper, is counted twice for the grayed out values and only once per paper for the others). The *Variome*, *Variome120*, *IDP4* and *nala_discoveries* corpora assembled different representations of NL mentions. The dashed line separates corpora with papers describing well-known, well-indexed genes and proteins (left of dashed line: *SETH*, *tmVar*, *Variome*, *Variome120*, *IDP4* and *nala_known*) and articles describing more recent discoveries that still have to be indexed in databases (right of dashed line: *nala_discoveries*)

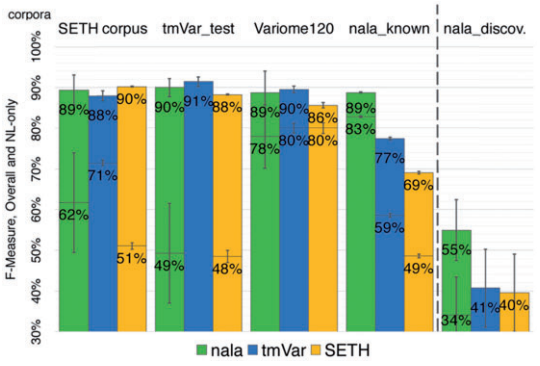


Fig. 3. *nala* performed well for all corpora. The bars give two different results: values above the horizontal lines in bars reflect the F-measures for all mentions, while values below the horizontal lines in bars reflect the F-measures for the subset of NL-mentions in the corpus (high error bars indicate corpora with few NL mentions). The exception was the result for the method tmVar on the corpus tmVar_test, which was taken from the original publication of the method in which no result was reported for NL-only (Wei *et al.*, 2013). That publication reports only *exact matching* performance, i.e. its *overlapping* performance might be higher than shown here. *nala* consistently matched or outperformed other top-of-the-line methods in well-indexed corpora (*SetsKnown;* left of dashed line) and substantially improved over the *status quo* in recent non-indexed discoveries (*nala_discoveries*; right of dashed line). The F-measures of tmVar and SETH for NL-only on *nala_discoveries* was essentially zero (two rightmost bars)

Table 3. Previously indexed versus new discoveries

	SetsKnown (indexed texts)			*nala_discoveries* (no indices)		
method	P	R	F ± StdErr	P	R	F ± StdErr
nala	87	92	89 ± 3	90	40	55 ± 7
tmVar	95	79	87 ± 3	93	26	41 ± 10
SETH	97	74	83 ± 5	93	25	40 ± 10

Note: Precision (P), Recall (R) and F-Measure (F) for methods on corpora with previously indexed articles (*SetsKnown*: *SETH*, *tmVar_test*, *Variome120*, *nala_known*) and a corpus directly sampled from PubMed without index (*nala_discoveries*).

Our new method *nala* essentially constituted a superset for the other two top methods in the following sense. The mutations correctly detected by *tmVar* and *SETH* were also found by *nala*. On top, *nala* correctly detected many mutations that had been missed by both other methods (Supplementary Fig. S1). Specifically, we looked at the subset of mentions correctly detected by any of the three methods (without considering repetitions, i.e. counting the detection of E6V only once per publication): 12% (*SetsKnown* corpora) and 33% (*nala_discoveries*) of mentions were exclusively found by *nala* (Fig. 4). In contrast, only 1% and 0% (*SetsKnown* and *nala_discoveries)* were exclusively found by *tmVar*; *SETH* added no exclusive detection. Moreover, 50% (*SetsKnown*) and 100% (*nala_discoveries*) of NL mentions were exclusively found by *nala* and only *tmVar* found 2% of novel NL mentions in the *SetsKnown*.

3.3 WE features are crucial/large variants are challenging

The Word Embedding (WE) features contributed significantly to the success of *nala* (Fig. 5). WE features improved performance for all mention types, most importantly for NL mentions (from F(WE = off)=70 to F(WE = on)=83 on *nala_known* corpus *and from* F(WE = off)=5 to F(WE = on)=34 on *nala_discoveries corpus*). In particular, WE vastly improved recall and even slightly improved the precision (Supplementary Table S8). All other features by the *nala* method were specific to mutation mentions and resulted from a laborious expert optimization. In contrast, WE features leveraged unsupervised data, i.e. can be adopted with minor modifications to any task or corpus.

We studied NER – Named Entity Recognition and ignored the considerably more difficult problem to map mutation mentions to sequences as needed to curate databases. Recent methods aim at this end (Mahmood *et al.*, 2016; Ravikumar *et al.*, 2015; Vohra and Biggin, 2013). However, all methods still primarily target SNVs/SAVs. We plan to extend the new corpora with exhaustive mapping annotations and to adapt the *nala* method to better cope with large-scale variations (predominant in *nala_discoveries*).

On new discoveries, the recall was 40%, i.e. 60% of the annotations were missed. 70% of these were large-scale variants, i.e. variations of regions longer than a few nucleotides or amino acids (presumably because their descriptions were less well-defined). For 44 of the 70% missed annotations, the annotators succeeded to position the sequence region (e.g. 'Deletion of the class 2 KNOTTED1-LIKE HOMEOBOX' or 'Robertsonian translocation between chromosomes 15 and 21' or 'amplification of 3q26/28 and 11q13/22'). For the remaining 26 of the 70% the descriptions of the variants were so vague that we could not assign sequences, but recognized large chromosomal changes (e.g. 'DNA double-strand breaks' or 'copy-number variants'). To complete the analysis of the 60% annotations missed in *nala_discoveries*: 22 of the 'small variation' 30% (100-70 = 30) were SAVs and SNVs, and 8% were other short

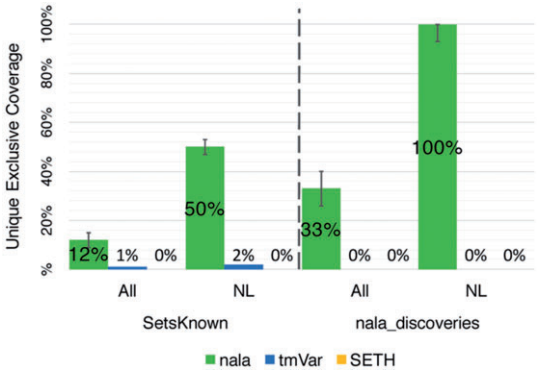


Fig. 4. *nala* could fully replace other methods. For each publication we considered all mentions correctly identified by one of the top three methods and kept only the findings unique in each publication. The y-axis plots the percentage of those mentions identified uniquely by one of the methods (All: all mentions, NL: NL-only mentions). For all corpora containing publications of genes and proteins indexed in the databases (*SetsKnown*), 1% of the mentions were detected only by *tmVar* and 12% only by *nala*, while *SETH* found no mention in this dataset that *nala* had not detected. Only *nala* correctly detected NL-only mentions in abstracts with new discoveries (100% bar on right triplet)

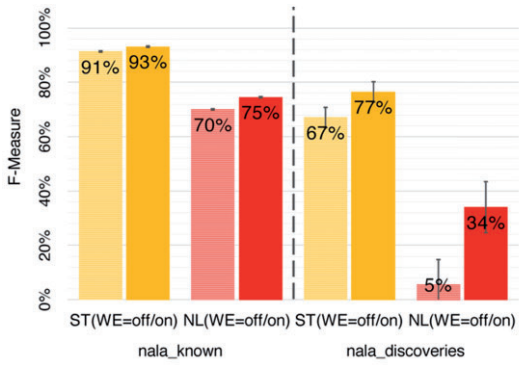


Fig. 5. Word embedding (WE) features crucial for success. The inclusion of WE features (WE = on versus WE = off) substantially improved performance for both *nala_known* (texts previously indexed) and *nala_discoveries* (no previous indices). The increase in performance was highest for NL mentions, but for ST mentions it was also significant

variants such as insertions, deletions and frameshifts involving only a few nucleotides. This implied that methods missed at least 2-3 times more single variants (SAVs and SNVs) in *nala_discoveries* than in *SetsKnown*, i.e. in proteins without previous annotations (data not shown; cf. 92% recall on *SetsKnown*, i.e. 8% missed annotations). As a practical use, we plan to research the performance of *nala* to effectively map HIV mutation mentions from whole PubMed (Davey *et al.*, 2014).

4 Conclusion

Previous accounts (Jimeno and Verspoor, 2014a,b, F1000Res.; Thomas *et al.*, 2016; Wei *et al.*, 2013) suggested that the strict named-entity recognition (NER) of mutation mentions constitutes a solved problem with performance levels reported to be F > 85. Despite this optimism, the

same authors (Caporaso *et al.*, 2007a,b; Jimeno and Verspoor, 2014a,b, Database) observed that methods failed to identify many mutations for database curation. Our work shed some light on this apparent paradox. First, mutation mentions often use natural language (NL) and were often missed by existing tools as they focused on standard (ST) forms. Second, existing corpora and methods primarily treated articles that had been previously indexed in databases. We showed that the percentage of publications with at least one mention in only NL ranged from 28 to 36% for indexed articles (*SetsKnown*) while it was twice as high (67–77%) for new discoveries (*nala_discoveries*, Table 2). Thus, most mentions relevant for database curation are only captured by methods versatile in NL.

We introduced the method *nala* designed to handle NL and ST mentions. In particular, word embedding (WE) features boosted performance for NL mentions (Fig. 5). In our hands, *nala* at least matched the best existing tools for publications that have already been curated in databases (corpora *SetsKnown*, dominated by ST mentions (F(nala)=89 ± 3 vs. F(tmVar)=87 ± 3, Table 3). Randomly sampling PubMed for new discoveries (*nala_discoveries*), *nala* was substantially better than existing methods (F(nala)=55 ± 7 versus F(SETH, tmVar)=40-41 ± 10, Table 3).

What do users have to expect: F = 89 or F = 55? The answer depends on what is known about the genes/proteins you are looking for. For *older* articles, point mutations, or *indels*, the current performance of all methods may suffice. For novel work or large-scale mutations, *nala* identifies many mutation mentions that are missed by others (Fig. 4). However, *nala* still missed about half of all variants described in the literature.

An important contribution of this work was the addition of three new corpora (*IDP4*, *nala_known* and *nala_discoveries*). These three new corpora accumulated the largest collection of mutation mentions: 826 documents (72 full texts), 627,953 tokens and 5660 mutation annotations (1110 NL). In comparison, the previous *SETH*, *tmVar* and *Variome120* corpora combined collect: 1,140 documents (10 full texts), 355,518 tokens and 2,933 mutation annotations (216 NL). In other words, this work boosted the available resources manifold. We released the new method as an open source python library and as API service and made the new corpora freely available: http://tagtog.net/-corpora/IDP4+

Acknowledgements

Thanks to Tim Karl for invaluable help with hardware and software; to Inga Weise for more than excellent administrative support; to Tatyana Goldberg for assistance in preparing and submitting the manuscript; to Maria Biryukov and Esteban Peguero Sánchez for helpful comments on the manuscript.

Funding

This work was supported by a grant from the Alexander von Humboldt foundation through the German Federal Ministry for Education and Research (BMBF).

Conflict of Interest: none declared.

References

Boutet,E. *et al.* (2016) UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view. *Methods Mol. Biol.*, **1374**, 23–54.

Caporaso,J.G. *et al.* (2007a) MutationFinder: a high-performance system for extracting point mutation mentions from text. *Bioinformatics*, **23**, 1862–1865.

Caporaso,J.G. *et al.* (2007b) Intrinsic evaluation of text mining tools may not predict performance on realistic tasks. In: *Biocomputing 2008*.

Cejuela,J.M. *et al.* (2014) tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. *Database (Oxford)*, **2014**, bau033.

Davey,N.E. *et al.* (2014) The HIV mutation browser: a resource for human immunodeficiency virus mutagenesis and polymorphism data. *PLoS Comput. Biol.*, **10**, e1003951.

den Dunnen,J.T. *et al.* (2016) HGVS recommendations for the description of sequence variants: 2016 update. *Hum. Mutat.*, **37**, 564–569.

Guo,J. *et al.* (2014) Revisiting Embedding Features for Simple Semi-supervised Learning. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP).

Jimeno,Y., A. and Verspoor,K. (2014a) Literature mining of genetic variants for curation: quantifying the importance of supplementary material. *Database (Oxford)*, **2014**, bau003. [WorldCat]

Jimeno,Y., A. and Verspoor,K. (2014b) Mutation extraction tools can be combined for robust recognition of genetic variants in the literature. *F1000Res*, **3**, 18.

Krallinger,M. *et al.* (2008) Linking genes to literature: text mining, information extraction, and retrieval applications for biology. *Genome Biol.*, **9**, S8.

Lafferty,J.D. *et al.* (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. p. 282–289. Morgan Kaufmann Publishers Inc.

Mahmood,A.S.M.A. *et al.* (2016) DiMeX: a text mining system for mutation-disease association extraction. *PLoS One*, **11**, e0152725.

Mikolov,T. *et al.* (2013) Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* 2013.

Nagel,K. *et al.* (2009) Annotation of protein residues based on a literature analysis: cross-validation against UniProtKb. *BMC Bioinformatics*, **10**, S4.

Passos,A. *et al.* (2014) Lexicon Infused Phrase Embeddings for Named Entity Resolution. In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*.

Ravikumar,K. *et al.* (2012) Literature mining of protein-residue associations with graph rules learned through distant supervision. *J. Biomed. Seman.*, **3**, S2.

Ravikumar,K.E. *et al.* (2015) Text mining facilitates database curation – extraction of mutation-disease associations from Bio-medical literature. *BMC Bioinformatics*, **16**.

Rost,B. (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol.*, **266**, 525–539.

Rost,B. *et al.* (2003) Automatic prediction of protein function. *Cell Mol. Life Sci.*, **60**, 2637–2650.

Sawyer,S.A. *et al.* (2007) Prevalence of positive selection among nearly neutral amino acid replacements in Drosophila. *Proc. Natl. Acad. Sci.*, **104**, 6504–6510.

Seok,M. *et al.* (2016) Named entity recognition using word embedding as a feature. *Int. J. Softw. Eng. Appl.*, **10**, 93–104.

Settles,B. and Burr,S. (2004) Biomedical named entity recognition using conditional random fields and rich feature sets. In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications – JNLPBA '04*.

Sherry,S.T. *et al.* (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, **29**, 308–311.

Stenson,P.D. *et al.* (2003) Human Gene Mutation Database (HGMD®): 2003 update. *Hum. Mutat.*, **21**, 577–581.

Tang,B. *et al.* (2014) Evaluating word representation features in biomedical named entity recognition tasks. *Biomed. Res. Int.*, **2014**, 240403.

Thomas,P. *et al.* (2016) SETH detects and normalizes genetic variants in text. *Bioinformatics*, **32**, 2883–2885.

UniProt,C. (2015) UniProt: a hub for protein information. *Nucleic Acids Res.*, **43**, D204–D212.

Verspoor,K. *et al.* (2013) Annotating the biomedical literature for the human variome. *Database*, **2013**, bat019.

Vohra,S. and Biggin,P.C. (2013) Mutationmapper: a tool to aid the mapping of protein mutation data. *PLoS One*, **8**, e71711.

Wei,C.H. *et al.* (2013) tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, **29**, 1433–1439.

Wei,C.H. *et al.* (2015) GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. *Biomed Res. Int.*, **2015**, 918710.

Supporting online material for:

nala: text mining of natural language mutations mentions

Juan Miguel Cejuela, Aleksandar Bojchevski, Carsten Uhlig, Rustem Bekmukhametov, Sanjeev Kumar Karn, Shpend Mahmuti, Ashish Baghudana, Ankit Dubey, Venkata P. Satagopam, & Burkhard Rost

1 Short description of Supporting Online Material

Some results not shown in main paper but supporting some described findings.

2 Material

(starts in next page; one Table/Figure per page)

Table S1. Individual results of all methods (default and *best* performances) on the *SETH* corpus (Thomas, et al., 2014). In bold the values considered for the final manuscript: averages of the default and *best* partial F-Measures and possibly maximum standard errors.

SETH corpus	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9195	0.8875	0.9032	0.0012	0.9333	0.8981	0.9154	0.0011		
SETH_best	0.9195	0.8875	0.9032	0.0012	0.9333	0.8981	0.9154	**0.0011**	**0.9154**	0.0000
tmVar	0.8994	0.8190	0.8573	0.0017	0.9158	0.8341	0.8731	0.0017		
tmVar_best	0.9601	0.8171	0.8828	0.0016	0.9769	0.8324	0.8989	0.0015	**0.8860**	**0.0129**
nala	0.7769	0.9049	0.8360	0.0016	0.8398	0.9727	0.9014	0.0012		
nala_best	0.9379	0.8759	0.9058	0.0010	0.9786	0.9206	0.9487	0.0007	**0.9251**	**0.0237**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9091	0.2941	0.4444	0.0089	1.0000	0.3429	0.5106	0.0080		
SETH_best	0.9091	0.2941	0.4444	0.0089	1.0000	0.3429	0.5106	**0.0080**	**0.5106**	0.0000
tmVar	0.8235	0.4242	0.5600	0.0085	1.0000	0.5556	0.7143	0.0066		
tmVar_best	0.8235	0.4242	0.5600	0.0085	1.0000	0.5556	0.7143	**0.0066**	**0.7143**	0.0000
nala	0.1071	0.4545	0.1734	0.0040	0.3291	1.0000	0.4952	0.0052		
nala_best	0.1897	0.3333	0.2418	0.0078	0.6184	0.9216	0.7402	0.0059	**0.6177**	**0.1225**
All	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9196	0.8597	0.8886	0.0012	0.9345	0.8713	0.9018	0.0011		
SETH_best	0.9196	0.8597	0.8886	0.0012	0.9345	0.8713	0.9018	0.0011	0.9018	0.0000
tmVar	0.8959	0.7998	0.8451	0.0018	0.9186	0.8208	0.8670	0.0018		
tmVar_best	0.9545	0.7978	0.8691	0.0017	0.9777	0.8191	0.8914	0.0015	0.8792	0.0122
nala	0.6818	0.8816	0.7689	0.0017	0.7656	0.9736	0.8571	0.0013		
nala_best	0.8779	0.8507	0.8640	0.0012	0.9421	0.9185	0.9302	0.0009	0.8937	0.0366

Table S2. Individual results of all methods (default and *best* performances) on the *tmVar_test* corpus (Wei, et al., 2013). In bold the values considered for the final manuscript: averages of the default and *best* partial F-Measures and possibly maximum standard errors. ***** tmVar could not be tested on tmVar_test and its results were taken from those reported in (Wei, et al., 2013). All following calculations use the reported exact performance results of tmVar instead of ignoring them, which in turn increased the overall performance of tmVar on *SetsKnown*.

tmVar_test	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9671	0.7933	0.8716	0.0019	0.9947	0.8198	0.8988	0.0016		
SETH_best	0.9671	0.7933	0.8716	0.0019	0.9947	0.8198	0.8988	**0.0016**	**0.8988**	0.0000
tmVar										
tmVar_best										
nala	0.8131	0.8112	0.8121	0.0022	0.9471	0.9641	0.9555	0.0010		
nala_best	0.9014	0.8427	0.8711	0.0019	0.9823	0.9310	0.9560	0.0009	**0.9558**	**0.0002**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.3333	0.2353	0.2759	0.0198	0.6154	0.4000	0.4848	0.0150		
SETH_best	0.3333	0.2353	0.2759	0.0198	0.6154	0.4000	0.4848	**0.0150**	**0.4848**	0.0000
tmVar										
tmVar_best										
nala	0.0700	0.4118	0.1197	0.0044	0.2273	1.0000	0.3704	0.0051		
nala_best	0.1190	0.2941	0.1695	0.0086	0.4528	0.9600	0.6154	0.0058	**0.4929**	**0.1225**
All	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9471	0.7716	0.8504	0.0019	0.9820	0.8008	0.8822	0.0016		
SETH_best	0.9471	0.7716	0.8504	0.0019	0.9820	0.8008	0.8822	**0.0016**	**0.8822**	0.0000
tmVar*	0.9138	0.9140	0.9139							
tmVar_best*	0.9138	0.9140	0.9139						**0.9139**	**0.0000**
nala	0.6571	0.7931	0.7188	0.0023	0.8025	0.9660	0.8767	0.0015		
nala_best	0.8102	0.8190	0.8146	0.0021	0.9114	0.9329	0.9220	0.0011	**0.8994**	**0.0227**

Table S3. Individual results of all methods (default and *best* performances) on the *Variome120* corpus (Jimeno Yepes and Verspoor, 2014). In bold the values considered for the final manuscript: averages of the default and *best* partial F-Measures and possibly maximum standard errors.

Var.120	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9398	0.7647	0.8432	0.0106	0.9770	0.8095	0.8854	0.0066		
SETH_best	0.9398	0.7647	0.8432	0.0106	0.9770	0.8095	0.8854	**0.0066**	**0.8854**	0.0000
tmVar	0.8627	0.8627	0.8627	0.0057	0.9266	0.9352	0.9309	0.0068		
tmVar_best	0.8627	0.8627	0.8627	0.0057	0.9266	0.9352	0.9309	**0.0068**	**0.9309**	0.0000
nala	0.6870	0.8738	0.7692	0.0119	0.7746	0.9821	0.8661	0.0098		
nala_best	0.8667	0.8922	0.8792	0.0093	0.9386	0.9817	0.9596	0.0053	**0.9129**	**0.0468**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.6667	0.3077	0.4211	0.0100	1.0000	0.6667	0.8000	0.0133		
SETH_best	0.6667	0.3077	0.4211	0.0100	1.0000	0.6667	0.8000	**0.0133**	**0.8000**	0.0000
tmVar	0.6667	0.3077	0.4211	0.0095	1.0000	0.6667	0.8000	0.0121		
tmVar_best	0.6667	0.3077	0.4211	0.0095	1.0000	0.6667	0.8000	**0.0121**	**0.8000**	0.0000
nala	0.1071	0.2500	0.1500	0.0181	0.5405	1.0000	0.7018	0.0092		
nala_best	0.2667	0.3077	0.2857	0.0167	0.7826	0.9474	0.8571	0.0052	**0.7795**	**0.0777**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9213	0.6833	0.7847	0.0107	0.9794	0.7600	0.8559	0.0068		
SETH_best	0.9213	0.6833	0.7847	0.0107	0.9794	0.7600	0.8559	**0.0068**	**0.8559**	0.0000
tmVar	0.8440	0.7667	0.8035	0.0071	0.9250	0.8672	0.8952	0.0084		
tmVar_best	0.8440	0.7667	0.8035	0.0071	0.9250	0.8672	0.8952	**0.0084**	**0.8952**	0.0000
nala	0.5629	0.7833	0.6551	0.0105	0.7225	0.9857	0.8338	0.0062		
nala_best	0.7638	0.8083	0.7854	0.0096	0.9048	0.9779	0.9399	0.0039	**0.8869**	**0.0531**

Table S4. Individual results of all methods (default and *best* performances) on the *nala_known* corpus. In bold the values considered for the final manuscript: averages of the default and *best* partial F-Measures and possibly maximum standard errors.

nala_known	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9764	0.7209	0.8294	0.0040	0.9922	0.7356	0.8449	0.0040		
SETH_best	0.9764	0.7294	0.8350	0.0041	0.9922	0.7442	0.8505	0.0039	**0.8477**	**0.0028**
tmVar	0.9424	0.7616	0.8424	0.0043	0.9931	0.8045	0.8889	0.0037		
tmVar_best	0.9424	0.7706	0.8479	0.0040	0.9931	0.8136	0.8944	0.0035	**0.8917**	**0.0027**
nala	0.8389	0.7267	0.7788	0.0044	0.9884	0.8763	0.9290	0.0024		
nala_best	0.8389	0.7267	0.7788	0.0044	0.9884	0.8763	0.9290	0.0024	**0.9290**	**0.0000**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.4242	0.0979	0.1591	0.0029	1.0000	0.3210	0.4860	0.0041		
SETH_best	0.4242	0.0979	0.1591	0.0029	1.0000	0.3210	0.4860	**0.0041**	**0.4860**	0.0000
tmVar	0.3571	0.1056	0.1630	0.0035	1.0000	0.4142	0.5858	0.0045		
tmVar_best	0.3571	0.1056	0.1630	0.0035	1.0000	0.4142	0.5858	**0.0045**	**0.5858**	0.0000
nala	0.4896	0.3310	0.3950	0.0036	0.9310	0.7459	0.8282	0.0023		
nala_best	0.4896	0.3310	0.3950	0.0036	0.9310	0.7459	0.8282	0.0023	**0.8282**	**0.0000**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.8439	0.4294	0.5692	0.0038	0.9847	0.5302	0.6893	0.0031		
SETH_best	0.8439	0.4320	0.5714	0.0038	0.9847	0.5331	0.6918	0.0032	**0.6906**	**0.0012**
tmVar	0.7650	0.4513	0.5677	0.0039	0.9959	0.6312	0.7727	0.0028		
tmVar_best	0.7650	0.4540	0.5698	0.0040	0.9959	0.6345	0.7751	0.0030	**0.7739**	**0.0012**
nala	0.6755	0.5280	0.5927	0.0034	0.9658	0.8208	0.8874	0.0016		
nala_best	0.6755	0.5280	0.5927	0.0034	0.9658	0.8208	0.8874	**0.0016**	**0.8874**	0.0000

Table S5. Individual results of all methods (default and *best* performances) on the *nala_discoveries* corpus. In bold the values considered for the final manuscript: averages of the default and *best* partial F-Measures. * The reported standard errors for nala_discoveries consider those of *SetsKnown + nala_discoveries*, Supplementary Table S7.

nala_discov.	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr*	Avg. F	Avg. StdErr*
SETH	0.8545	0.4519	0.5912	0.0078	0.9322	0.5093	0.6587	0.0082		
SETH_best	0.8545	0.4519	0.5912	0.0078	0.9322	0.5093	0.6587	0.0082	**0.6587**	0.0000
tmVar	0.8571	0.4615	0.6000	0.0082	0.9333	0.5185	0.6667	0.0079		
tmVar_best	0.8571	0.4615	0.6000	0.0082	0.9333	0.5185	0.6667	0.0079	**0.6667**	0.0000
nala	0.7385	0.4615	0.5680	0.0078	0.9494	0.6410	0.7653	0.0069		
nala_best	0.7385	0.4615	0.5680	0.0078	0.9494	0.6410	0.7653	0.0069	**0.7653**	0.0000
NL	P	R	F	StdErr	P	R	F	StdErr*	Avg. F	Avg. StdErr*
SETH	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
SETH_best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0000**	0.0000
tmVar	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
tmVar_best	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0000**	0.0000
nala	0.4167	0.0704	0.1205	0.0042	0.8889	0.2105	0.3404	0.0066		
nala_best	0.4167	0.0704	0.1205	0.0042	0.8889	0.2105	0.3404	0.0066	**0.3404**	0.0000
All	P	R	F	StdErr	P	R	F	StdErr*	Avg. F	Avg. StdErr*
SETH	0.8545	0.2186	0.3481	0.0057	0.9322	0.2511	0.3957	0.0060		
SETH_best	0.8545	0.2186	0.3481	0.0057	0.9322	0.2511	0.3957	0.0060	**0.3957**	0.0000
tmVar	0.8596	0.2279	0.3603	0.0057	0.9344	0.2603	0.4071	0.0061		
tmVar_best	0.8596	0.2279	0.3603	0.0057	0.9344	0.2603	0.4071	0.0061	**0.4071**	0.0000
nala	0.6585	0.2512	0.3636	0.0052	0.9020	0.3948	0.5493	0.0058		
nala_best	0.6585	0.2512	0.3636	0.0052	0.9020	0.3948	0.5493	0.0058	**0.5493**	0.0000

Table S6. Individual results of all methods (default and *best* performances) on the *SetsKnown* corpus (merge of: SETH corpus, tmVar_test, Variome120, and nala_known). In bold the values considered for the final manuscript: averages of the default and *best* partial F-Measures and possibly maximum standard errors. ***** tmVar could not be tested on tmVar_test and its results were taken from those reported in (Wei, et al., 2013). All following calculations used the reported exact performance results of tmVar instead of ignoring them, which in turn increased the overall performance of tmVar on *SetsKnown*.

SetsKnown	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9507	0.7916	0.8619	0.0163	0.9743	0.8158	0.8861	0.0150		
SETH_best	0.9507	0.7937	0.8633	0.0155	0.9743	0.8179	0.8875	0.0138	**0.8868**	**0.0007**
tmVar	0.9015	0.8144	0.8541	0.0061	0.9452	0.8579	0.8976	0.0172		
tmVar_best	0.9217	0.8168	0.8645	0.0101	0.9655	0.8604	0.9081	0.0115	**0.9029**	**0.0052**
nala	0.7790	0.8292	0.7990	0.0154	0.8875	0.9488	0.9130	0.0191		
nala_best	0.8862	0.8344	0.8587	0.0277	0.9720	0.9274	0.9483	0.0068	**0.9307**	**0.0177**
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.5833	0.2338	0.3251	0.0667	0.9039	0.4327	0.5704	0.0768		
SETH_best	0.5833	0.2338	0.3251	0.0667	0.9039	0.4327	0.5704	0.0768	**0.5704**	**0.0000**
tmVar	0.6158	0.2792	0.3814	0.1163	1.0000	0.5455	0.7000	0.0622		
tmVar_best	0.6158	0.2792	0.3814	0.1163	1.0000	0.5455	0.7000	0.0622	**0.7000**	**0.0000**
nala	0.1935	0.3618	0.2095	0.0628	0.5070	0.9365	0.5989	0.1025		
nala_best	0.2663	0.3165	0.2730	0.0472	0.6962	0.8937	0.7602	0.0543	**0.6796**	**0.0807**
All	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9080	0.6860	0.7732	0.0713	0.9702	0.7406	0.8323	0.0486		
SETH_best	0.9080	0.6867	0.7738	0.0708	0.9702	0.7413	0.8329	**0.0480**	**0.8326**	0.0003
tmVar*	0.8547	0.7330	0.7826	0.0751	0.9465	0.7731	0.8622	0.0314		
tmVar_best*	0.8693	0.7331	0.7891	0.0765	0.9531	0.8087	0.8689	**0.0317**	**0.8656**	0.0034
nala	0.6443	0.7465	0.6839	0.0383	0.8141	0.9365	0.8638	0.0118		
nala_best	0.7819	0.7515	0.7642	0.0594	0.9310	0.9125	0.9199	0.0114	**0.8918**	**0.0281**

Table S7. Individual results of all methods (default and *best* performances) on *SetsKnown* corpus + *nala_discoveries* (merge of: SETH corpus, tmVar_test, Variome120, nala_known, and nala_discoveries). In bold the values considered for the final manuscript: possibly maximum standard errors. ***** tmVar could not be tested on tmVar_test and its results were taken from those reported in (Wei, et al., 2013).

SetsKnown+nala_discov.	Exact				Partial					
ST	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.9315	0.7237	0.8077	0.0556	0.9659	0.7545	0.8406	0.0470		
SETH_best	0.9315	0.7254	0.8088	0.0557	0.9659	0.7562	0.8418	**0.0470**	0.8412	0.0006
tmVar	0.8904	0.7262	0.7906	0.2369	0.9422	0.7731	0.8399	0.2495		
tmVar_best	0.9056	0.7280	0.7984	0.2399	0.9575	0.7749	0.8477	**0.2522**	0.8438	0.0039
nala	0.7709	0.7556	0.7528	0.0477	0.8999	0.8872	0.8835	0.0331		
nala_best	0.8567	0.7598	0.8006	0.0620	0.9675	0.8701	0.9117	**0.0370**	0.8976	0.0141
NL	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.4667	0.1870	0.2601	0.0831	0.7231	0.3461	0.4563	0.1286		
SETH_best	0.4667	0.1870	0.2601	0.0831	0.7231	0.3461	0.4563	**0.1286**	0.4563	0.0000
tmVar	0.4618	0.2094	0.2860	0.1506	0.7500	0.4091	0.5250	0.2357		
tmVar_best	0.4618	0.2094	0.2860	0.1506	0.7500	0.4091	0.5250	**0.2357**	0.5250	0.0000
nala	0.2381	0.3035	0.1917	0.0518	0.5834	0.7913	0.5472	0.0948		
nala_best	0.2963	0.2673	0.2425	0.0476	0.7347	0.7571	0.6763	**0.0939**	0.6117	0.0645
All	P	R	F	StdErr	P	R	F	StdErr	Avg. F	Avg. StdErr
SETH	0.8973	0.5925	0.6882	0.1014	0.9626	0.6427	0.7450	0.0951		
SETH_best	0.8973	0.5930	0.6886	0.1013	0.9626	0.6433	0.7455	**0.0950**	0.7452	0.0002
tmVar*	0.8557	0.6319	0.6981	0.1026	0.9435	0.6449	0.7712	0.0942		
tmVar_best*	0.8674	0.6321	0.7033	0.1042	0.9583	0.6453	0.7765	**0.0956**	0.7739	0.0027
nala	0.6472	0.6474	0.6198	0.0706	0.8317	0.8282	0.8009	0.0635		
nala_best	0.7572	0.6514	0.6841	0.0924	0.9252	0.8090	0.8458	**0.0746**	0.8233	0.0225

nala: text mining natural language mutations mentions

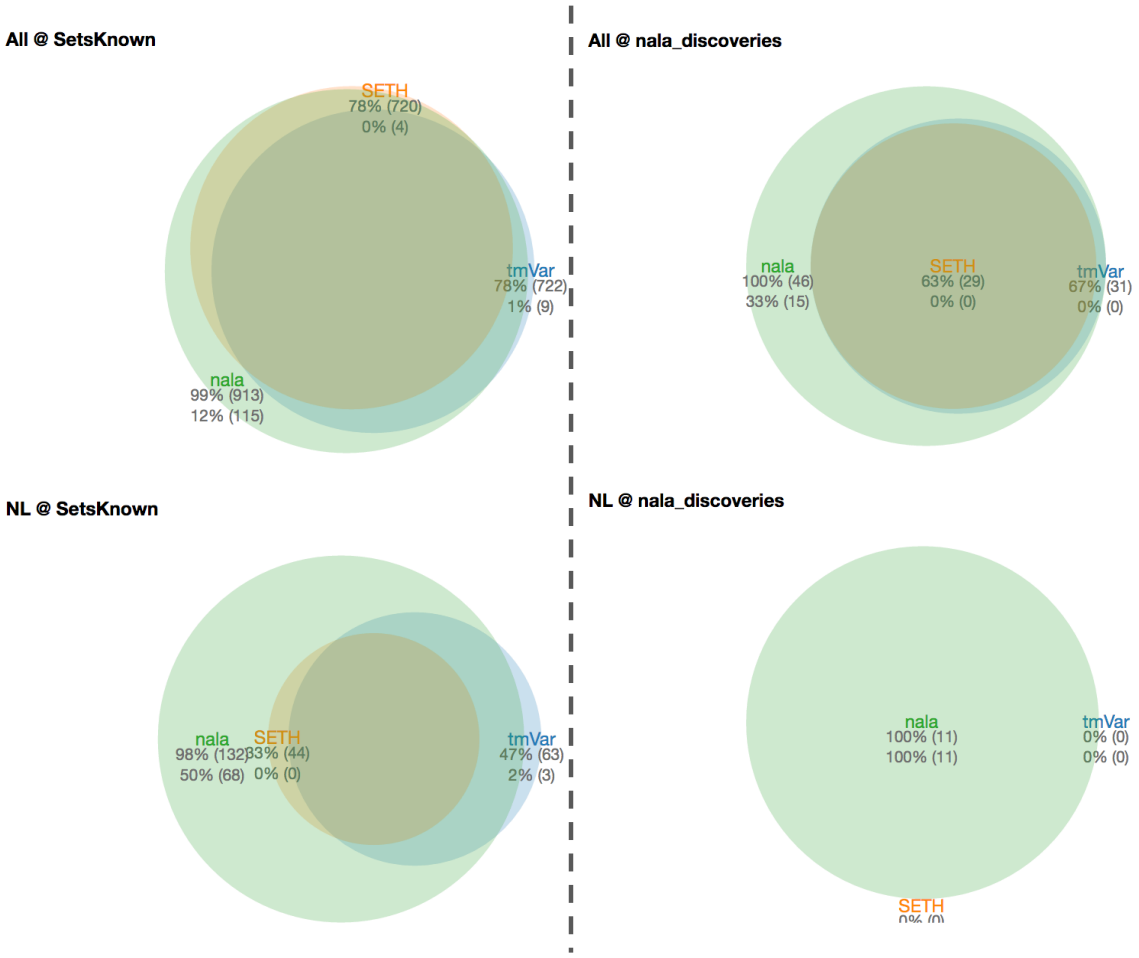


Fig. S1: ***nala*** **could fully replace other methods, Venn Diagrams.** Here, we looked at the following subset of all mentions. For each publication we considered all the mentions correctly identified by one of the top three methods and kept only the findings unique in each publication (the first sublabel after a method's name shows its correctly recovered percentage of unique mentions and, in parenthesis, the exact number of recovered unique mentions). We then asked the number of those had been identified uniquely by one of the methods distinguishing between all mentions and NL-only mentions (the second sublabel after a method's name shows its correctly recovered percentage of unique mentions that were not found by any other method and, in parenthesis, the exact number of recovered unique mentions that were not found by any other medhod). For instance, for all corpora containing publications of genes and proteins indexed in the databases (SetsKnown), 9 of the mentions (1%) were detected only by *tmVar* and 115 (12%) only by *nala*, while *SETH* found no mention in this data set that *nala* had not detected. One the other end, only *nala* correctly detected NL-only mentions in papers reporting discoveries on genes/proteins not indexed in databases, 11 (100%), right-bottwon Venn diagram.

Table S8. Detailed results of the effect on performance (*nala* method) of the word embedding (WE) features. * The possibly maximum StdErr for *nala_known* was taken from Supplementary Table S4 and the possibly maximum StdErr for *nala_discoveries* was taken from Supplementary Table S7.

	P	R	F	StdErr	Max StdErr*
nala_known					
ST WE=off	99	84	91	0	0
ST WE=on	99	88	93	0	0
NL WE=off	98	54	70	0	0
NL WE=on	93	75	83	0	0
nala_discoveries					
ST WE=off	93	52	67	1	4
ST WE=on	95	64	77	1	4
NL WE=off	100	3	5	0	9
NL WE=on	89	21	34	1	9

Table S9. Custom regular expressions we used to classify some one-letter-coded mutation mentions as standard form (ST). The regular expressions are here written in JavaScript –compatible form.

Regular Expression
\\w+: del [ACTGRNDEQHILKMFPSWYV] -?\\d+\\.\\.\\.-?\\d+
\\d+.*[ACTGRNDEQHILKMFPSWYV] --> [ACTGRNDEQHILKMFPSWYV]
[cgp]\\. ?\\d+ ?[ACTGRNDEQHILKMFPSWYV]>[ACTGRNDEQHILKMFPSWYV]

Table S10. List of software or utilities used with name, creation or version date, URL, and last accessed date.

Software Resources
2016. Online Mendelian Inheritance in Man, OMIM®. http://omim.org/. (2016/5/13 date last accessed)
NCBI. 2015. NCBI Text Mining Tools. http://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/. (2016/5/13 date last accessed)
Okazaki, N. 2007. CRFsuite - A fast implementation of Conditional Random Fields (CRFs). http://www.chokkan.org/software/crfsuite/. (2016/5/13 date last accessed)
Stenetorp, P., Pyysalo, S. and Topić, G. 2014. Standoff format - brat rapid annotation tool. http://brat.nlplab.org/standoff.html. (2016/5/13 date last accessed)|
tpeng. 2015. tpeng/python-crfsuite. https://github.com/tpeng/python-crfsuite. (2016/5/13 date last accessed)

nala: text mining natural language mutations mentions

References

Jimeno Yepes, A. and Verspoor, K. Mutation extraction tools can be combined for robust recognition of genetic variants in the literature. *F1000Res.* 2014;3:18.
Thomas, P.*, et al.* 2014. SETH - SNP Extraction Tool for Human Variations. https://rockt.github.io/SETH/. (2016/5/13 date last accessed)|.
Wei, C.-H.*, et al.* tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics* 2013;29(11):1433-1439.

3.3 References

Ashburner, M. et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium". In: *Nat Genet* 25.1, pp. 25–9. ISSN: 1061-4036 (Print) 1061-4036 (Linking). DOI: 10.1038/75556. URL: https://www.ncbi.nlm.nih.gov/pubmed/10802651.

Boutet, E. et al. (2016). "UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View". In: *Methods Mol Biol* 1374, pp. 23–54. ISSN: 1940-6029 (Electronic) 1064-3745 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/26519399.

Cejuela, J. M. et al. (2014). "tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles". In: *Database (Oxford)* 2014.0, bau033. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bau033. URL: https://www.ncbi.nlm.nih.gov/pubmed/24715220.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: http://dx.doi.org/10.1007/BF00994018.

Goldberg, Tatyana et al. (2015). "Linked annotations: a middle ground for manual curation of biomedical databases and text corpora". In: *BMC Proceedings* 9.Suppl 5, A4–A4. ISSN: 1753-6561. DOI: 10.1186/1753-6561-9-S5-A4. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4582921/.

Goldberg, T. et al. (2014). "LocTree3 prediction of localization". In: *Nucleic Acids Res* 42.Web Server issue, W350–5. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gku396. URL: https://www.ncbi.nlm.nih.gov/pubmed/24848019.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. 655813: Morgan Kaufmann Publishers Inc., pp. 282–289.

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Ng, Andrew Y. (2004). "Feature selection, L1 vs. L2 regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*. 1015435: ACM, p. 78. DOI: 10.1145/1015330.1015435.

Sayers, E. W. et al. (2010). "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Res* 38.Database issue, pp. D5–16. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkp967. URL: https://www.ncbi.nlm.nih.gov/pubmed/19910364.

Sherry, S. T. et al. (2001). "dbSNP: the NCBI database of genetic variation". In: *Nucleic Acids Res* 29.1, pp. 308–11. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/11125122.

Simpson, J. C. and R. Pepperkok (2003). "Localizing the proteome". In: *Genome Biol* 4.12, p. 240. ISSN: 1474-760X (Electronic) 1474-7596 (Linking). DOI: 10.1186/gb-2003-4-12-240. URL: https://www.ncbi.nlm.nih.gov/pubmed/14659010.

The UniProt Consortium (2017). "UniProt: the universal protein knowledgebase". In: *Nucleic Acids Res* 45.D1, pp. D158–D169. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1099. URL: https://www.ncbi.nlm.nih.gov/pubmed/27899622.

UniProt, Consortium (2015). "UniProt: a hub for protein information". In: *Nucleic Acids Res.* 43.Database issue, pp. D204–12. ISSN: 0305-1048. DOI: 10.1093/nar/gku989. URL: http://dx.doi.org/10.1093/nar/gku989%20http://www.ncbi.nlm.nih.gov/pubmed/25348405%20http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4384041%20http://nar.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=25348405.

Chapter 4

LocText: relation extraction of protein localizations to assist database curation

4.1 Preface

The subcellular localization of proteins defines and constraints their range of functionality. Knowing the localization guides the development of drugs that act and target proteins of specific regions. Despite the importance, however, the annotation of experimentally-established localizations is not remotely complete yet, even for well-studied species (Simpson and Pepperkok 2003; T. Goldberg et al. 2014). Text mining methods may discover annotations that are hidden in the literature and so assist database curation. However, few past methods proved useful yet. Most existing solutions only coarsely related all proteins and localizations that were co-mentioned, as in a same sentence or a same document.

In this work, we developed a new method, *LocText*, to extract protein localization relationships from the literature. The new method learned language patterns from syntax trees. We modeled features with support vector machines (SVMs) (Cortes and Vapnik 1995). Nearly all features were unsupervised and automatically selected with L1 regularization (Ng 2004). We trained and cross-validated the *LocText* method on the homonymous *LocTextCorpus*, previously annotated with the *tagtog* tool (Cejuela et al. 2014; Tatyana Goldberg et al. 2015), and in this work improved. The *STRING Tagger* was used to automatically recognize the mentions of proteins, subcellular locations, and organisms. These were mapped to identifiers of standard biological databases, respectively: UniProtKB (UniProt 2015), Gene Ontology (Ashburner et al. 2000), and NCBI Taxonomy (Sayers et al. 2010). The new method was compared to a *Baseline* that relates all proteins and locations co-mentioned in the same sentence. On the *LocTextCorpus*, both *LocText* and *Baseline* missed many annotations (coverage of 43% vs. 50%). However, *LocText* was highly accurate (86% vs 51%).

We used the new method to find novel protein-location annotations in last scientific pub-

lications. Thereby, we manually asserted 60 of the text-mined, potentially novel annotations (i.e. not annotated in UniProtKB/Swiss-Prot (Boutet et al. 2016)). A 65% of the verified predictions for human (*Homo sapiens*) were correct. The success rate was higher for budding yeast (*Saccharomyces cerevisiae*) and thale cress (*Arabidopsis thaliana*): 85% and 80%, respectively. We were able to verify the 60 text-mined annotations in 3 person-hours. In other words, with an average successful rate of ~77%, we could add over one hundred new and correct annotations to highly-accurate databases such as UniProtKB/Swiss-Prot.

The methods were designed by me, Shrikant Vinchurkar, and Tatyana Goldberg. The final implementation of *LocText* was done by me. Tatyana Goldberg and Madhukar Sollepura Prabhu Shankar added new annotations to the *LocTextCorpus*. The results evaluation on the *LocTextCorpus* was done by me. The results analysis of new publications was done by me, Tatyana Goldberg, and Burkhard Rost. Additional research and code development were done by Madhukar Sollepura Prabhu Shankar, Ashish Baghudana, Aleksander Bojchevski, Carsten Uhlig, André Ofner, and Pandu Raharja-Liu. Finally, the manuscript was prepared by me, Lars Juhl Jensen, and Burkhard Rost.

4.2 Journal article. Cejuela *et al.*, *BMC Bioinformatics* 2018; 19

Starts next page.

Cejuela et al. BMC Bioinformatics (2018) 19:15
DOI 10.1186/s12859-018-2021-9

BMC Bioinformatics

RESEARCH ARTICLE Open Access

LocText: relation extraction of protein localizations to assist database curation

Juan Miguel Cejuela[1*], Shrikant Vinchurkar[2], Tatyana Goldberg[1], Madhukar Sollepura Prabhu Shankar[1], Ashish Baghudana[3], Aleksandar Bojchevski[1], Carsten Uhlig[1], André Ofner[1], Pandu Raharja-Liu[1], Lars Juhl Jensen[4*] and Burkhard Rost[1,5,6,7,8*]

Abstract

Background: The subcellular localization of a protein is an important aspect of its function. However, the experimental annotation of locations is not even complete for well-studied model organisms. Text mining might aid database curators to add experimental annotations from the scientific literature. Existing extraction methods have difficulties to distinguish relationships between proteins and cellular locations co-mentioned in the same sentence.

Results: *LocText* was created as a new method to extract protein locations from abstracts and full texts. *LocText* learned patterns from syntax parse trees and was trained and evaluated on a newly improved *LocTextCorpus*. Combined with an automatic named-entity recognizer, *LocText* achieved high precision (P = 86% ± 4). After completing development, we mined the latest research publications for three organisms: human (*Homo sapiens*), budding yeast (*Saccharomyces cerevisiae*), and thale cress (*Arabidopsis thaliana*). Examining 60 novel, text-mined annotations, we found that 65% (human), 85% (yeast), and 80% (cress) were correct. Of all validated annotations, 40% were completely novel, i.e. did neither appear in the annotations nor the text descriptions of Swiss-Prot.

Conclusions: *LocText* provides a cost-effective, semi-automated workflow to assist database curators in identifying novel protein localization annotations. The annotations suggested through text-mining would be verified by experts to guarantee high-quality standards of manually-curated databases such as Swiss-Prot.

Keywords: Relation extraction, Text mining, Protein, Subcellular localization, GO, Annotations, Database curation

Background

The subcellular location of a protein is an important aspect of its function because the spatial environment constrains the range of operations and processes. For instance, all processing of DNA happens in the nucleus or the mitochondria. In fact, subcellular localization is so important that the Gene Ontology (GO) [1], the standard vocabulary for protein functional annotation, described it by one of its three hierarchies (*Cellular Component*). Many proteins function in different locations. Typically, one of those constitutes the *native* location, i.e. the one in which the protein functions most importantly.

Despite extensive annotation efforts, experimental GO annotations in databases are not nearly complete [2]. Automatic methods may close the annotation gap, i.e. the difference between experimental knowledge and database annotations.

Numerous methods predict location from homology-based inference or sequence-based patterns (sorting signals). These include: *WoLF PSORT* [3], *SignalP* [4], *CELLO* [5], *YLoc* [6], *PSORTb* [7], and *LocTree3* [8]. Text mining-based methods can also "predict" (extract) localization, with the added benefit of linking annotations to the original sources. Curators can compare those resources to validate the suggested annotations and add annotations to high-quality resources such as Swiss-Prot [9] or those for model organisms, e.g. *FlyBase* [10]. An alternative to finding annotations in the free literature is mining controlled texts, such as descriptions and annotation tags in databases [11–13]. Despite numerous past

*Correspondence: loctext@rostlab.org; lars.juhl.jensen@cpr.ku.dk; rost@rostlab.org
[1]Bioinformatics & Computational Biology, Department of Informatics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany
Full list of author information is available at the end of the article

efforts, however, very few text mining systems succeeded in assisting GO curation [14]. A notable exception is *Textpresso* [15], which was integrated into the GO cellular component annotation pipeline of *WormBase* [16] and sped up annotation tenfold over manual curation [17]. Similar computer-assisted curation pipelines have since been implemented for other model organisms [18], but no generic solution for the usage of text mining tools to experts is extensively used yet [19, 20].

Literature-based text mining methods begin with *named-entity recognition* (*NER*), namely the recognition of names of entities, such as proteins or cellular compartments, mentioned within the text. These entities then have to be *normalized*, i.e. disambiguated by mapping the names to exact identifiers in controlled vocabularies (e.g. proteins mapped to UniProtKB [21] and cell compartments to GO). The next task is the *relation extraction* (*RE*) in which relationships between the entities have to be deduced from the semantic context. As an example, in the sentence "CAT2 is localized to the tonoplast in transformed Arabidopsis protoplasts", PMID (PubMed Identifier) 15377779, the relationship of "CAT2" (UniProtKB: P52569) localized to "tonoplast" (GO:0009705) must be established. Most existing GO annotation methods either coarsely associate all pairs of entities that are co-mentioned in a same sentence or otherwise aggregate the statistics of one or more levels of co-mention (such as the same sentence, paragraph, section, or document). Examples of this include the *CoPub Mapper* [22], *EBIMed* [23], and the *COMPARTMENTS* database [24]. *Textpresso* used manually defined regular expressions. Few methods machine-learned the semantics of text, even if only learning *bags of words* (i.e. disregarding grammar) [25, 26]. Newer methods modeled the syntax of text too (i.e. considering grammar) though were not validated yet in practice for database curation [27–30]. The most recent method of this type [31] probed the discovery of novel protein localizations in unseen publications. However, the method performed poorly in extracting unique relations, i.e. to find out that the same localization relation is described in a publication multiple times but using different synonymous (e.g. due to abbreviations or different spellings). Related to this, the method did not normalize tagged entities; thus, the relations could not be mapped to databases.

To the best of our knowledge, the new method, *LocText*, is the first method to implement a fully-automated pipeline with NER, RE, normalized entities, and linked original sources (necessary for database curation) that machine-learnt the semantics and syntax of scientific text. The system was assessed to achieve high accuracy in a controlled corpus (*intrinsic evaluation*), and to retrieve novel annotations from the literature in a real task (*extrinsic evaluation*).

Results

Most relations found in same or consecutive sentences

The controlled *LocTextCorpus* had annotated 66% of all protein-location unique relations (i.e. collapsing repetitions, "Methods" section) in the same sentence (D0, where *Dn* means that the relation covers entities *n* sentences apart) and 15% in consecutive sentences (D1; Fig. 1). When the GO hierarchy was also considered to collapse redundant relations, D0 (same sentence) increased to 74% (e.g. "lateral plasma membrane", GO:0016328, overshadowed the less detailed "plasma membrane", GO:0005886). Consequently, a method that extracted only same-sentence relationships could maximally reach a recall of 74%; at 100% precision, the maximal F-score of such a method would be 85%. Methods that extracted both D0 (same-sentence) and D1 (consecutive sentences) would have a maximal recall of 89% (max. F = 94%). Considering more distant sentences would rapidly increase the pairs of entities to classify and, with this, likely reduce a method's precision and substantially increase processing time. *LocTextCorpus* had annotated relationships up to sentence distances of nine (D9). However, after collapsing repeated relations, the maximum distance was six (D6).

Intrinsic evaluation: relation extraction (RE) and named-entity extraction (NER) succeeded

LocText (RE) and *STRING Tagger* (NER) (Methods) independently performed well on the *LocTextCorpus*: *LocText* (RE only) reached P = 93% at R = 68% (F = 79% ± 3; Table 1). A high precision was achieved while closely reaching the maximum possible recall for considering only same-sentences relations (D0; max. R = 74%). The *Baseline* (using manually-annotated entities; Methods) also performed well (P = 75% at R = 74%; F = 74% ± 3). A comparative Precision-Recall (PR) curve analysis is shown in Additional file 1: Figure S3. The *STRING Tagger* benchmarked on overlapping normalized entities obtained an aggregated F = 81% ± 1, for the entities Protein (F = 79% ± 2), Location (F = 80% ± 3), and Organism (F = 94% ± 1; Table 1). The precision for the entities Location (P = 90%) and Organism (P = 96%) was much higher than for Protein (P = 80%).

The full *LocText* relation extraction pipeline (NER + RE) achieved high precision (P = 86%) at the cost of low recall (R = 43%; F = 57% ± 4, Fig. 2). The *Baseline* (using tagged entities) remained low in precision (P = 51%) and recall (R = 50%; $F = 51\% \pm 3$). Recall might be so low because the errors in RE and NER cumulate: mistakes in identifying the protein, the location, or their relation lead to wrong annotations.

Extrinsic evaluation: high accuracy enables database curation

Encouraged by the high precision of *LocText*, it was applied to extract protein localization GO annotations

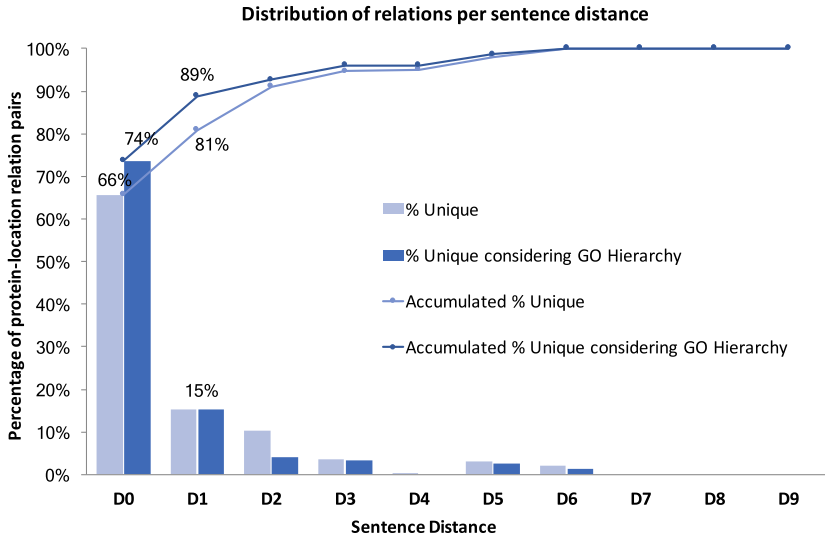


Fig. 1 Most related protein and localizations closed to each other. Repetitions of relationships were collapsed at the document level after normalizing the entities: proteins to UniProtKB and localizations to GO. In the *LocTextCorpus*, the majority of unique relations were annotated between entities occurring in the same sentence (distance 0 = D0; 66% of all relations) or in adjacent sentences (dist. 1 = D1; 15%). Combined, D0+D1 accounted for 81% of the relations. Removing repetitions when considering the GO hierarchy (children identifiers are more exact than their parents), D0+D1 accounted for 89% of all unique relationships

from recent PubMed abstracts (*NewDiscoveries_human*, *NewDiscoveries_yeast*, and *NewDiscoveries_cress*; "Methods" section). *LocText* extracted ∼24k unique GO annotations, ∼11k of which (46%) were not found in Swiss-Prot. Some annotations were found in several abstracts. The reliability of the *LocText* annotations increased when found more often. For instance, 10% of the human annotations were found in three or more abstracts (corresponding numbers for yeast: 14%, and thale cress: 6%).

For each organism, the first 20 annotations observed in exactly three abstracts were reviewed. Of the 20 GO annotations for human, 13 (65%) were novel (Table 2; examples of mined novel GO annotations in Additional file 1: Table S2); three of these were more detailed versions of the Swiss-Prot annotations (i.e. child terms in the GO hierarchy). 10 of the 20 had no related annotation in Swiss-Prot (50%). For yeast and cress the novelty fraction was even higher: 85% for yeast (60% without related annotation) and 80% for thale cress (55% without related annotation). The total number of correct novel GO annotations was 46

Table 1 *LocText* (RE only) and *STRING Tagger* (NER); intrinsic evaluation

Method and evaluation	P	R	F ±*StdErr*
STRING Tagger Total	84%	78%	81% ± 1
STRING Tagger on Protein	80%	78%	79% ± 2
STRING Tagger on Location	90%	71%	80% ± 3
STRING Tagger on Organism	96%	92%	94% ± 1
LocText, with manual entities	93%	68%	79% ± 3
Baseline, with manual entities	75%	74%	74% ± 3

Performances of the NER and RE components independently evaluated on the *LocTextCorpus*; P=precision, R=recall, F ±*StdErr*=F-measure with standard error

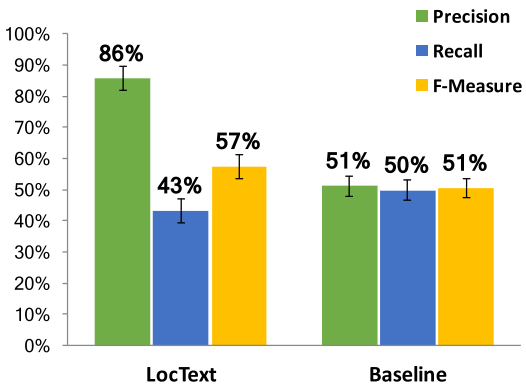


Fig. 2 *LocText* full pipeline (NER + RE); intrinsic evaluation. Using the *STRING Tagger*-extracted ("predicted") entities, both *LocText* and *Baseline* had low and comparable F-measure (F=57%± 4 and F=51%± 3, resp.), however *LocText* was optimized for precision (P=86%)

Table 2 *LocText* found novel GO annotations in latest publications; extrinsic evaluation

Org.	#	C	C&NR	C&NT	C&NR,NT
Human	20	13 (65%)	10 (50%)	9 (45%)	7 (35%)
Yest	20	17 (85%)	12 (60%)	6 (30%)	4 (20%)
Cress	20	16 (80%)	11 (55%)	9 (45%)	7 (35%)
Total	60	46 (77%)	33 (55%)	24 (40%)	18 (30%)

LocText mined protein location relations not tagged in Swiss-Prot in latest publications: 2012-2017 for (column *Org.*=organism) human and 1990-2017 for yeast and cress. (#) 60 novel text-mined annotations (20 for each organism) were manually verified: (*C*=correct) 77% were correct; 55% were correct and had *no relation* (*NR*) in Swiss-Prot; 40% were correct and were *not in text* (*NT*) descriptions of Swiss-Prot; 30% were correct and neither had a relation nor appeared in text descriptions

of 60 (77%) of which 33 (55%) had no related Swiss-Prot annotation.

Upon closer inspection of Swiss-Prot, we found that some of the allegedly novel predictions could have been found in Swiss-Prot text descriptions or other annotations (e.g. biological processes). Still, 9 of the 20 (45%) human annotations were not found (considering also texts) in Swiss-Prot (35% without related annotation in Swiss-Prot considering the GO hierarchy). At that point, we could have gone back and dug deeper, but we could not automate the identification of "find in Swiss-Prot" because the relations were not found through the standard Swiss-Prot tags. The corresponding numbers for yeast and cress were 30% (20% without related annotation) and 45% (35% without related annotation), respectively. The total number of verified completely novel GO annotations not in Swiss-Prot remained as high as 24 out of 60 (40%), of these 18 (30% of 60) had no relation in Swiss-Prot.

23% of the verified predictions were wrong. Half of these errors originated from incorrect proteins, typically due to short and ambiguous abbreviations in the name. For example, "NLS" was wrongly normalized to protein O43175, yet in all texts they referred to "nuclear localization signals". "FIP3" was wrongly recognized as "NF-kappa-B essential modulator" (Q9Y6K9) while in the three abstracts in which it was found, it referred to "Rab11 family-interacting protein 3" (O75154). The same abbreviation is used for both proteins making this a perfect example how text mining can be beaten by innovative naming. Another 14% of the errors were due to a wrong named-entity localization prediction. For example, in PMID 22101002, the P41180 was correctly identified with the abbreviation CaR, and yet a same abbreviation in the text was also wrongly predicted to be the localization "contractile actomyosin ring".

The remaining 36% of the errors were due to a wrong relationship extraction. For example, the relation that the protein Cx43 (connexin 43, or "gap junction alpha-1 protein" P17302) is/acts in microtubules could not be fully ascertained from the sentence: "Although it is known that Cx43 hemichannels are transported along microtubules to the plasma membrane, the role of actin in Cx43 forward trafficking is unknown" (PMID 22328533). Another wrongly predicted relationship was OsACBP2 (Q9STP8) to cytosol where the seemingly text proof explicitly negated the relationship: "Interestingly, three small rice ACBP (OsACBP1, OsACBP2 and OsACBP3) are present in the cytosol in comparison to one (AtACBP6) in Arabidopsis" (PMID 26662549). Other wrongly extracted relationships did not show any comprehensible language patterns and were likely predicted for just finding the protein and location co-mentioned.

Discussion

Achieving high precision might be the most important feature for an automatic method assisting in database curation. Highly-accurate databases such as Swiss-Prot or those of model organisms need to expert-verify all annotations. Focusing on few reliable predictions, expert curators minimize the resources (time) needed to confirm predictions. The manual verification of the 60 GO annotations extracted with *LocText* from recent PubMed abstracts took three person-hours (20 annotations per hour; 60 abstracts per hour). Seventy seven percent of the *LocText* predicted annotations were correct, i.e. an unexperienced expert (we) could easily add ∼120 new annotations on an average 9-5 day to the UniProtKB repository.

The *LocText* method was very fast: it took 45 min to process ∼ 37k PubMed abstracts on a single laptop (MacBook Pro 13-inch, 2013, 2 cores). These ∼37k abstracts spanned a wide range of the most recent (from 2012 to 2017) research on human proteins localizations. Twenty one percent of the running time was spent to extract the named entities (*STRING Tagger)*, 26% on text parsing (spaCy), and 52% on pure relationship extraction (*LocText*). If parallelized, *LocText* could process the entire PubMed in near real time.

We discarded relations spanning over more than two sentences (distance$\geq$1), as the marginal improvements in recall and F-measure did not justify the significant drops in precision. Nevertheless, extracting relations between two neighbor sentences (D1) might increase recall in the future (from 66 to 81% unique relations disregarding the GO hierarchy and 74 to 89% considering the hierarchy).

One important question often neglected in the text mining literature is how well the performance estimates live up to the reality of users, for instance of database curators. Much controversy has followed the recent observations that many if not most published results even in highly-regarded journals (Science and Nature) are not reproducible or false [32–34]. As a curiosity, a GO annotation predicted by *LocText* (deemed wrong upon

manual inspection) was found in three journals that were retracted (PMIDs 22504585 and 22504585; the third 23357054 duplicated 22504585). The articles, written by the same authors, were rejected after publication as "expert reviewers agreed that the interpretation of the results was not correct" (PMID 22986443). This work has added particular safe-guards against over-estimating performance (additional data set not used for development), and for gauging performance from the perspective of the user (extrinsic vs. intrinsic evaluation). With all these efforts, it seems clear that novel *GO annotations* suggested by *LocText* have the potential to significantly reduce annotation time (as compared to curators manually searching for new publications and reading those) yet still require further expert verification.

Conclusions

Here, we presented *LocText*, a new text mining method optimized to assist database curators for the annotation of protein subcellular localizations. *LocText* extracts protein-*in*-location relationships from texts (e.g. PubMed) using syntax information encoded in parse trees. Common language patterns to describe a localization relationship (e.g. "co-localized in") were learned unsupervised and thus the methodology could extrapolate to other annotation domains.

LocText was benchmarked on an improved version of *LocTextCorpus* [35] and compared against a *Baseline* that relates all proteins and locations co-mentioned in a same sentence. Benchmarking only the relation extraction component, i.e. with manually annotated entities, *LocText* and *Baseline* appeared to perform comparably. However, *LocText* achieved much higher precision (P(*LocText*) = 93% vs. P(*Baseline*) = 75%). The full pipeline combining the *STRING Tagger* (NER) with *LocText* (RE) reached a low F-measure (F = 57%± 4) and a low recall (R = 43%). However, it was optimized for the high precision (P(*LocText*) = 86% vs. P(*Baseline*) = 51%).

LocText found novel GO annotations in the latest literature for three organisms: human, yeast, and thale cress. 77% of the examined predictions were correct localizations of proteins and were not annotated in Swiss-Prot. More novel annotations could successfully be extracted for yeast and cress (∼80%) than for human (∼65%). Novel annotations that were not traceable from Swiss-Prot (either from annotation tags or from text descriptions) were analyzed separately. Using this definition for *novel annotations*, 40% of all findings were novel. Unexperienced curators (we) validated 20 predicted GO annotations in 1 person-hour. Assisted by the new *LocText* method, curators could enrich UniProtKB with ∼120 novel annotations on a single job day. Advantaging existing automatic methods (*Baseline* with accuracy of 40%-50%), *LocText* could cut curation time in half. Compared to solely manual curation (still common in biological databases), the new method can reduce efforts and resources greatly.

All code, data, and results were open sourced from the start and are available at http://tagtog.net/-corpora/LocText. The new written code added relationship extraction functionality to the *nalaf* framework of natural language processing [36].

Methods

Named-entity recognition (NER)

The complete *LocText* pipeline consisted of a NER component stacked with a pure RE component (Fig. 3). The RE component was the focus of this work, and its implementation is explained in the following subsections. For NER we reused the existing dictionary-based *STRING Tagger*, which is described in detail in earlier publications [24, 37]. We employed *STRING Tagger* to extract the entities from the text: proteins (more generally, gene or gene products), subcellular localizations, and organisms. Next, we needed to map these to databases, namely to UniProtKB accession numbers, to GO Cellular Component identifiers, and to NCBI Taxonomy identifiers (note:

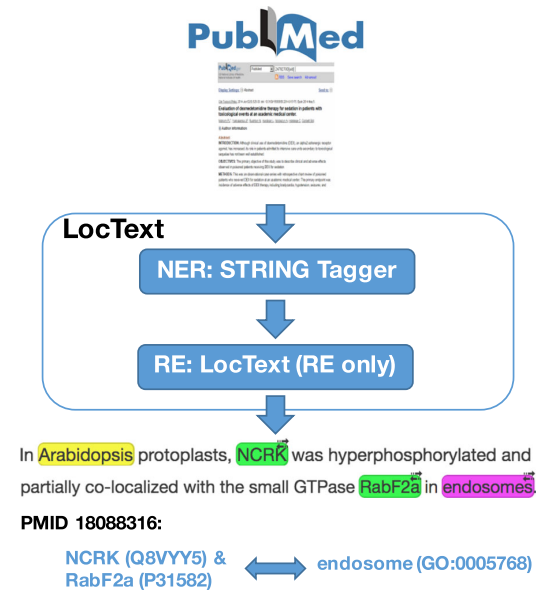


Fig. 3 *LocText* pipeline. The input are text documents (e.g. PubMed). First, the *STRING Tagger* recognizes named entities (NER): proteins (green in the example; linked to UniProtKB), cellular localizations (pink; linked to GO), and organisms (yellow; linked to NCBI Taxonomy). Then, the relation extractor (RE) of *LocText* resolves which proteins and localizations are related (as in "localized in"). The output is a list of text-mined relationships (GO annotations) linked to the original text sources (e.g. PMIDs)

this map is referred to as *normalization* in the text mining community). The method extracts text mentions and the normalized identifiers of entities; it maps proteins to STRING identifiers. We mapped these to UniProtKB accession numbers and ran the Python-wrapped tagger through an in-house Docker-based web server.

The *STRING Tagger* allows the selective usage of organism-dependent dictionaries for protein names. We ran the tagger against the *LocTextCorpus* (see, "Text corpora" section) having selected the dictionaries of human (NCBI Taxonomy: 9606), yeast (NCBI 4932), and thale cress (NCBI 3702). On the sets of documents *NewDiscoveries_human, NewDiscoveries_yeast*, and *NewDiscoveries_cress* (Text corpora), we selected only the corresponding organism. We did not consider this selective choice of articles and dictionaries to bias results as this is standard for the curation of model organisms [10, 18, 36]. As another option of the *STRING Tagger*, we also annotated the proteins of other organisms if the protein and organism names were written close to each other in text. For reference, we ran the tagger against *LocTextCorpus* with exact parameters (options): *ids=-22,-3,9606,4932,3702 autodetect=true*. We did not modify the tagger in any way except for removing "Golgi" from the list of *stopwords* (blacklist of names not to annotate) as it likely referred to "Golgi apparatus" in publications known to mention cellular components. We filtered the results by GO identifier to only allow those that were (part of) cell organelles, membranes, or extracellular region. We also explicitly filtered out all tagged cellular components that constituted a "macromolecular complex" (GO:0032991) as in most cases they were enzyme protein complexes, which we did not study (they overlap with the molecular function and biological process hierarchies of the GO ontology). We evaluated the *STRING Tagger* in isolation for NER ("Results" section).

Relation extraction (RE)

We reduced the problem of relationship extraction to a binary classification: for pairs of entities Prot/Loc (protein/location), decide if they are related (true or false). Several strategies for the generation of candidate pairs are possible, e.g. the enumeration of all combinations from all {Prot/Loc} mentioned in a document. During training, "repeated relation pairs" are used, i.e. the exact text offsets of entities are considered, as opposed to the entity normalizations only (Evaluation). The pairs marked as relations in an annotated corpus (*LocTextCorpus*) are positive instances and other pairs are negative instances. For our new method, we generated only pairs of entities co-occurring in the same sentence. This strategy generated 663 instances (351 positive, 312 negative). Instances were represented as a sentence-based sequence of words along with syntax information (see, Feature selection). We also designed ways to generate and learn from pairs of entities mentioned in consecutive sentences (e.g. the protein mentioned in one sentence and the location in the next). However, we discarded this in the end ("Discussion" section). We modeled the instances with support vector machines (SVMs; [38]). We used the *scikit-learn* implementation with a linear kernel [39, 40]. Neither the tree kernel [41] implemented in SVM-light [42, 43], nor the radial basis function kernel performed better. Other models such as random forests or naive Bayes methods (with either Gaussian, Multinomial, or Bernoulli distributions) also did not perform better in our hands; logistic regression also performed worse, however, within standard error of the best SVM model. For syntactic parsing, we used the python library spaCy (https://spacy.io). For word tokenization, we used our own implementation of the *tmVar*'s tokenizer [36, 44]. This splits contiguous letters and numbers (e.g. "P53" is tokenized as "P" and "53").

Feature selection

An instance (positive or negative) is defined as a protein location pair (Prot/Loc) that carries contextual information (the exact text offsets of entities are used). We contemplated features from five different sources: corpus-based, document-based, sentence-based, syntax-based, and domain-specific. The first four were *domain agnostic*. Tens of thousands of features would be generated (compared to 663, the number of instances). Many features, however, were highly correlated. Thus, we applied feature selection. First, we did leave-one-out feature selection, both through manual and automatic inspection (on the validation set, i.e. when cross-training). In the end, by far the most effective feature selection strategy was the Lasso L1 regularization [45]. We ran the *scikit-learn LinearSVC* implementation with penalty = L1 and C = 2 (SVM trade-off hyperparameter). The sparsity property of the L1 norm effectively reduced the number of features to ~ 300 (ratio of 2 = num. instances / num. features). We applied independent feature selection whether we used the manually annotated entities or the entities identified by *STRING Tagger*. Both yielded almost equal features. Ultimately, we only used the following five feature types.

Entity counts in the sentence (domain agnostic, 2 features): individual entity counts (for protein, location, and organisms too) and the total sum. Counts were scaled to floats [0, 1] dividing them by the largest number found in the training data (independently for each feature). If the test data had a larger number than previously found while training, its scaled float would be bigger than 1 (e.g. if the largest number in training was 10, a count of 11 in testing would be scaled to 1.1).

Is protein a marker (domain specific, 1 feature): for example, green fluorescent protein (GFP), or red fluorescent protein (RFP). This might be a problem of

the *LocTextCorpus* guidelines. Nonetheless, disregarding protein markers seems a reasonable step to curate databases.

Is the relation found in Swiss-Prot (domain specific, 1 feature): we leveraged the existing annotations from Swiss-Prot.

N-grams between entities in linear dependency (domain agonistic, 57% of ~ 300 features): the n-grams ($n = 1, 2$, or 3*)* of tokens in the linear sentence between the pair of entities Prot and Loc. The tokens were mapped in two ways: 1) word lemmas in lower case masking numbers as the special *NUM* symbol and masking tokens of mentioned entities as their class identifier (i.e. *PROTEIN, LOCATION, or ORGANISM)*; 2) words part of speech (POS). In a 2- or 3-gram, the entity on the left was masked as *SOURCE* and the end entity on the right as *TARGET*.

N-grams of syntactic dependency tree (domain agnostic, 42% of ~ 300 features): the shortest path in the dependency parse tree connecting Prot and Loc was computed (Additional file 1: Figure S1). The connecting tokens were mapped in three ways: 1) word lemmas with same masking as before; 2) part of speech, same masking; 3) syntactic dependencies edges (e.g. *preposition* or *direct object*). Again, we masked the pair of entities in the path as *SOURCE* and *TARGET*. The direction of the edges in the dependency tree (going up to the sentence root or down from it) was not outputted after feature selection.

The representation of the sentences as dependency graphs was inspired by Björne's method for event extraction in BioNLP'09 [46]. The n-gram features, both linear- and dependency-tree-based, that were ultimately chosen after unsupervised feature selection yielded comprehensible language patterns (Additional file 1: Table S1). In the Supplementary Online Material (SOM), we listed all the features that were finally selected (Additional file 1: Figure S2).

Evaluation

High performance of a method in a controlled setting (*intrinsic evaluation*) does not directly translate into high performance in a real task (*extrinsic* evaluation) [47]. To address this, we evaluated the new *LocText* method in both scenarios, namely, in a well-controlled corpus using standard performance measures and in the real setting of extracting novel protein localizations from the literature. Either way, and always with database curation in mind, we asked: given a scientific text (e.g. PubMed article), what protein location relationships does it attest to? For instance, a publication may reveal "Protein S" (UniProtKB: P07225) to function in the "plasma membrane" (GO:0005886). To extract this relation, it is indifferent under which names the protein and location are mentioned. For instance, P07225 can also be named "Vitamin K-dependent protein S" or "PROS1" or abbreviated "PS" and GO:0005886 can also be called "cell membrane" or "cytoplasmic membrane" or abbreviated "PM". Further, it does not matter if the relation is expressed with different but semantically equivalent phrases (e.g. "PROS1 was localized in PM" or "PM is the final destination of PROS1"). Regardless of synonymous names and different wordings, repeated attestations of the relation within the same document are all the same. In other words, we evaluated relationship extraction at the document level and for normalized entities.

In intrinsic evaluation, the annotated relations of a corpus were grouped by document and represented as a unique set of normalized entity pairs of the form (Prot=protein, Loc=location), e.g. (P07225, GO:0005886). A tested known relationship ($\text{Prot}_{\text{test}}$, Loc_{test}) was considered as correctly extracted (*true positive* = tp), if at least one text-mined relation ($\text{Prot}_{\text{pred}}$, Loc_{pred}) matched it, with both Prot and Loc correctly normalized: 1) $\text{Prot}_{\text{test}}$ and $\text{Prot}_{\text{pred}}$ must be equal or have a percentage sequence identity 90% (to account for cases where likely a same protein entries can have multiple identifiers in UniProtKB/TrEMBL [48]); and 2) Loc_{test} and Loc_{pred} must be equal or Loc_{pred} must be a leave or child of Loc_{test} (to account for the tree-based GO hierarchy). For example, a tested (P07225, GO:0005886) relation and a predicted (P07225, GO:0016328) relation correctly match: the proteins are the same and GO:0016328 ("lateral plasma membrane") is a part of and thus more detailed than GO:0005886 ("plasma membrane"). Any other predicted relationship was wrong (*false positive* = fp), and any missed known relationship was also punished (*false negative* = fn). We then computed the standard performance measures for *precision* $\left(\text{P} = \frac{tp}{tp+fp}\right)$, *recall* $\left(\text{R} = \frac{tp}{tp+fn}\right)$, and *F-measure* $\left(\text{F} = 2 * \frac{P*R}{P+R}\right)$ (all three multiplied by 100, in percentages).

We evaluated relationship extraction in isolation (using manually-annotated entities, i.e. the proteins and localizations) and as a whole (with predicted entities). Given the importance of the NER module (wrongly predicted entities lead to wrongly predicted relationships), we also evaluated the NER in isolation. We considered a predicted named entity as successfully extracted (*tp*) if and only if its text offsets (character positions in a text-string) *overlapped* those of a known entity and its normalized identifier matched the same test entity's normalization (also accounting for similar proteins and for the GO hierarchy). Any other predicted entity was counted as *fp* and any missed entity as *fn*. In analogy, we computed P, R, and F for named-entity recognition.

We evaluated methods in 5-fold cross-validation with three separate sets as follows. First, we split a fold into the three sets by randomizing the publications; this lessens redundancy as different publications mention

different localizations. Sixty percent of documents served to train (train set), 20% to cross-train (validation set), i.e. to optimize parameters such as in feature or model selection. The remaining 20% were used for testing (test set). The performance on the test set was compiled only after all development had been completed and was thus not used for any optimization. Finally, we repeated the folds four more times, such that each article had been used for testing exactly once. We computed the standard error (*StdErr*) by randomly selecting 15% of the test data without replacement in 1000 (n) bootstrap samples. With $\langle x \rangle$ as the overall performance for the entire test set and x_i for subset i, we computed:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \langle x \rangle)^2} \qquad StdErr = \frac{\sigma}{\sqrt{n}} \quad (1)$$

In extrinsic evaluation, the complete *LocText* pipeline (i.e. NER + RE) extracted from large sets of unannotated PubMed abstracts novel protein localizations (namely, GO annotations not tagged in Swiss-Prot). A unique protein-location relation could be found in one or more documents. The assumption is: the more document hits, the more reliable the extracted relation. For a number of extracted unique relations, one person manually reviewed the originating and linked documents. For each "predicted" relation, we stopped our analysis when we found proof of the annotation. We deemed the prediction to be wrong if we found no textual proof in the abstracts.

Text corpora

To train and formally benchmark the new method (intrinsic evaluation), we had only access to a custom-built corpus, for simplicity referred to as *LocTextCorpus* [35]. We could not reuse other annotated corpora as they did not provide annotations at the text level or had incompatible annotations. Specifically, the *BioNLP'09* corpus [28] and the *BC4GO* corpus [49] appeared very promising but contained particular features that made it impossible for us to use those valuable resources. *BioNLP'09*, for instance, annotated *events* (relationships) not requiring the textual mention of the protein or localization entities, some location mentions contained extraneous words that were part of the phrase but not strictly part of the location names, and some locations were not only subcellular localizations but specific cells or body tissues. *BC4GO* contained neither exact text-level annotations of the entities nor the relationships.

We had previously annotated the *LocTextCorpus* with the *tagtog* tool [50]. For this work, we added 8 missing protein normalizations. *LocTextCorpus* collected 100 abstracts (50 abstracts for human proteins, 25 for yeast, and 25 for thale cress) with 1393 annotated proteins, 558 localizations, and 277 organisms. The organism annotation had been crucial to correctly map the protein sequence, e.g. to distinguish the human *Protein S* (P07225/PROS_HUMAN) from its mouse ortholog (Q08761/PROS_MOUSE). The corpus annotated 1345 relationships (550 protein-localization + 795 protein-organism). When removing repeated relations through entity normalization (Evaluation), the number of unique protein-localization relations was 303. Relationships of entities mentioned in any sentence apart had been annotated (Results). That is, the related protein and location entities could have been mentioned in the same sentence (sentence distance=0, D0), or contiguous sentences (sentence distance=1, D1), or farther away (D$\geq$ 2). The agreement (F-measure) between two annotators (an estimation of the quality of annotations) reached as high as: F = 96 for protein annotation, F = 88 for localization annotation, and F = 80 for protein-localization relationship annotation. *LocTextCorpus* was used to train, select features, and test (in cross-validation) the new *LocText* method.

Furthermore, and to assess how the new method *LocText* could assist in database curation in practice, three sets of PubMed abstracts were added: *NewDiscoveries_human*, *NewDiscoveries_yeast*, *NewDiscoveries_cress*. For each organism, keyword searches on PubMed revealed recent publications that likely evidenced (mentioned) the localization of proteins (e.g. the search for human *http://bit.ly/2nLiRCK*). The search for all human-related journals published between 2012 to 2017/03 yielded $\sim$37k documents (exactly 37454). For publication years from 1990 to 2017/03, the search obtained $\sim$18k (17544) documents for yeast and $\sim$8k (7648) for cress. These documents were not fully tagged. They were only used for final *extrinsic* evaluation, and only after the method had been finalized. In other words, those abstracts never entered any aspect of the development/training phase.

Existing methods for comparison

Two previous methods that used machine learning techniques to model syntax also extracted protein localization relationships [27, 31]. However, neither methods were made available. We found no other machine learning-based methods available for comparison. The *Textpresso* system uses regular expressions and is used in database curation [15]. The method, however, is packaged as a search index (suited to their specialized corpora, e.g. for WormBase) and not as an extraction method. We were not able to run it for new corpora.

Other methods exist that follow a simple heuristic: if two entities are *co-mentioned* then they are related [22–24]. The heuristic of same-sentence co-occurrence

(as opposed to e.g. document co-occurrence) is simple and yields top results. Therefore, this was considered as the *Baseline* to compare the new method against.

Additional file

Additional file 1: Supporting online material. PDF document with supplemental figures and tables (Fig. S1-S3, Tables S1-S2), one per page. (PDF 238 kb)

Abbreviations
F: F-measure; GO: Gene ontology; Loc: Location; NER: Named-entity recognition; P: Precision; Prot: Protein; R: Recall; RE: Relation extraction

Acknowledgements
The authors thank Tim Karl for invaluable help with hardware and software, Inga Weise for more than excellent administrative support, Jorge Campos for proof reading, Shpend Mahmuti for help with docker.

Funding
Alexander von Humboldt Foundation through German Federal Ministry for Education and Research, Ernst Ludwig Ehrlich Studienwerk, and the Novo Nordisk Foundation Center for Protein Research (NNF14CC0001).
This work was supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program.

Availability of data and materials
The *LocTextCorpus* improved and analyzed during the current study is available in the *tagtog* repository, http://tagtog.net/-corpora/LocText.
The sets of PubMed abstracts (*NewDiscoveries_human*, *NewDiscoveries_yeast*, *NewDiscoveries_cress*) analyzed during the current study are publicly available on PubMed; searches: human http://bit.ly/2nLiRCK, yeast http://bit.ly/2pve2Pe, and cress http://bit.ly/2q1Nh4X.

Authors' contributions
JMC, SV, and TG designed the methods; JMC developed the method; JMC, LJJ, and BR prepared the manuscript; MSPS, AB (Baghudana), AB (Bojchevski), CU, AO, and PRL provided supporting research and code development. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Bioinformatics & Computational Biology, Department of Informatics, Technical University of Munich (TUM), Boltzmannstr. 3, 85748 Garching, Germany. [2]Microsoft, Microsoft Development Center Copenhagen, Kanalvej 7, 2800 Kongens Lyngby, Denmark. [3]Department of Computer Science and Information Systems, Birla Institute of Technology and Science K. K. Birla Goa Campus, 403726 Zuarinagar, Goa, India. [4]Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, 2200 Copenhagen N, Denmark. [5]Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany. [6]TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany. [7]Columbia University, Department of Biochemistry and Molecular Biophysics, Columbia University, New York, USA. [8]New York Consortium on Membrane Protein Structure (NYCOMPS), 701 West, 168th Street, 10032 New York, NY, USA.

Received: 25 April 2017 Accepted: 10 January 2018
Published online: 17 January 2018

References
1. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. Gene ontology: tool for the unification of biology. the gene ontology consortium. Nat Genet. 2000;25(1):25–9. https://doi.org/10.1038/75556.
2. Zhou H, Yang Y, Shen HB. Hum-mPLoc 3.0: prediction enhancement of human protein subcellular localization through modeling the hidden correlations of gene ontology and functional domain features. Bioinformatics. 2017;33(6):843–53. https://doi.org/10.1093/bioinformatics/btw723.
3. Horton P, Park KJ, Obayashi T, Fujita N, Harada H, Adams-Collier CJ, Nakai K. WoLF PSORT: protein localization predictor. Nucleic Acids Res. 2007;35(Web Server issue):585–7. https://doi.org/10.1093/nar/gkm259.
4. Petersen TN, Brunak S, von Heijne G, Nielsen H. SignalP 4.0: discriminating signal peptides from transmembrane regions. Nat Methods. 2011;8(10):785–6. https://doi.org/10.1038/nmeth.1701.
5. Yu CS, Chen YC, Lu CH, Hwang JK. Prediction of protein subcellular localization. Proteins. 2006;64(3):643–51. https://doi.org/10.1002/prot.21018.
6. Briesemeister S, Rahnenfuhrer J, Kohlbacher O. YLoc–an interpretable web server for predicting subcellular localization. Nucleic Acids Res. 2010;38(Web Server issue):497–502. https://doi.org/10.1093/nar/gkq477.
7. Yu NY, Wagner JR, Laird MR, Melli G, Rey S, Lo R, Dao P, Sahinalp SC, Ester M, Foster LJ, Brinkman FS. PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. Bioinformatics. 2010;26(13):1608–15. https://doi.org/10.1093/bioinformatics/btq249.
8. Goldberg T, Hecht M, Hamp T, Karl T, Yachdav G, Ahmed N, Altermann U, Angerer P, Ansorge S, Balasz K, Bernhofer M, Betz A, Cizmadija L, Do KT, Gerke J, Greil R, Joerdens V, Hastreiter M, Hembach K, Herzog M, Kalemanov M, Kluge M, Meier A, Nasir H, Neumaier U, Prade V, Reeb J, Sorokoumov A, Troshani I, Vorberg S, Waldraff S, Zierer J, Nielsen H, Rost B. LocTree3 prediction of localization. Nucleic Acids Res. 2014;42(Web Server issue):350–5. https://doi.org/10.1093/nar/gku396.
9. Boutet E, Lieberherr D, Tognolli M, Schneider M, Bansal P, Bridge AJ, Poux S, Bougueleret L, Xenarios I. UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View. Methods Mol Biol. 2016;1374:23–54.
10. Gramates LS, Marygold SJ, Santos GD, Urbano JM, Antonazzo G, Matthews BB, Rey AJ, Tabone CJ, Crosby MA, Emmert DB, Falls K, Goodman JL, Hu Y, Ponting L, Schroeder AJ, Strelets VB, Thurmond J, Zhou P, the FlyBase Consortium. FlyBase at 25: looking to the future. Nucleic Acids Res. 2017;45(D1):663–71. https://doi.org/10.1093/nar/gkw1016.
11. Lu Z, Szafron D, Greiner R, Lu P, Wishart DS, Poulin B, Anvik J, Macdonell C, Eisner R. Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics. 2004;20(4):547–6. https://doi.org/10.1093/bioinformatics/bth026.
12. Shatkay H, Hoglund A, Brady S, Blum T, Donnes P, Kohlbacher O. SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. Bioinformatics. 2007;23(11):1410–7. https://doi.org/10.1093/bioinformatics/btm115.
13. Nair R, Rost B. Inferring sub-cellular localization through automated lexical analysis. Bioinformatics. 2002;18 Suppl 1:78–86.
14. Mao Y, Van Auken K, Li D, Arighi CN, McQuilton P, Hayman GT, Tweedie S, Schaeffer ML, Laulederkind SJ, Wang SJ, Gobeill J, Ruch P, Luu AT, Kim JJ, Chiang JH, Chen YD, Yang CJ, Liu H, Zhu D, Li Y, Yu H, Emadzadeh E, Gonzalez G, Chen JM, Dai HJ, Lu Z. Overview of the gene ontology task at biocreative iv. Database (Oxford) 2014;2014. https://doi.org/10.1093/database/bau086.
15. Muller HM, Kenny EE, Sternberg PW. Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Biol. 2004;2(11):309. https://doi.org/10.1371/journal.pbio.0020309.

16. Harris TW, Baran J, Bieri T, Cabunoc A, Chan J, Chen WJ, Davis P, Done J, Grove C, Howe K, Kishore R, Lee R, Li Y, Muller HM, Nakamura C, Ozersky P, Paulini M, Raciti D, Schindelman G, Tuli MA, Van Auken K, Wang D, Wang X, Williams G, Wong JD, Yook K, Schedl T, Hodgkin J, Berriman M, Kersey P, Spieth J, Stein L, Sternberg PW. WormBase 2014: new views of curated biology. Nucleic Acids Res. 2014;42(Database issue): 789–93. https://doi.org/10.1093/nar/gkt1063.
17. Van Auken K, Jaffery J, Chan J, Muller HM, Sternberg PW. Semi-automated curation of protein subcellular localization: a text mining-based approach to gene ontology (go) cellular component curation. BMC Bioinformatics. 2009;10:228. https://doi.org/10.1186/1471-2105-10-228.
18. Van Auken K, Fey P, Berardini TZ, Dodson R, Cooper L, Li D, Chan J, Li Y, Basu S, Muller HM, Chisholm R, Huala E, Sternberg PW, WormBase C. Text mining in the biocuration workflow: applications for literature curation at WormBase, dictyBase and TAIR. Database (Oxford). 2012;2012: 040. https://doi.org/10.1093/database/bas040.
19. Arighi CN, Carterette B, Cohen KB, Krallinger M, Wilbur WJ, Fey P, Dodson R, Cooper L, Van Slyke CE, Dahdul W, Mabee P, Li D, Harris B, Gillespie M, Jimenez S, Roberts P, Matthews L, Becker K, Drabkin H, Bello S, Licata L, Chatr-aryamontri A, Schaeffer ML, Park J, Haendel M, Van Auken K, Li Y, Chan J, Muller HM, Cui H, Balhoff JP, Chi-Yang Wu J, Lu Z, Wei CH, Tudor CO, Raja K, Subramani S, Natarajan J, Cejuela JM, Dubey P, Wu C. An overview of the BioCreative 2012 Workshop Track III: interactive text mining task. Database (Oxford). 2013;2013:056. https://doi.org/10.1093/database/bas056.
20. Wang Q, S SA, Almeida L, Ananiadou S, Balderas-Martinez YI, Batista-Navarro R, Campos D, Chilton L, Chou HJ, Contreras G, Cooper L, Dai HJ, Ferrell B, Fluck J, Gama-Castro S, George N, Gkoutos G, Irin AK, Jensen LJ, Jimenez S, Jue TR, Keseler I, Madan S, Matos S, McQuilton P, Milacic M, Mort M, Natarajan J, Pafilis E, Pereira E, Rao S, Rinaldi F, Rothfels K, Salgado D, Silva RM, Singh O, Stefancsik R, Su CH, Subramani S, Tadepally HD, Tsaprouni L, Vasilevsky N, Wang X, Chatr-Aryamontri A, Laulederkind SJ, Matis-Mitchell S, McEntyre J, Orchard S, Pundir S, Rodriguez-Esteban R, Van Auken K, Lu Z, Schaeffer M, Wu CH, Hirschman L, Arighi CN. Overview of the interactive task in BioCreative V. Database (Oxford). 2016;2016:. https://doi.org/10.1093/database/baw119.
21. The UniProt Consortium. Uniprot: the universal protein knowledgebase. Nucleic Acids Res. 2017;45(D1):158–69. https://doi.org/10.1093/nar/gkw1099.
22. Alako BT, Veldhoven A, van Baal S, Jelier R, Verhoeven S, Rullmann T, Polman J, Jenster G. CoPub Mapper: mining MEDLINE based on search term co-publication. BMC Bioinformatics. 2005;6:51. https://doi.org/10.1186/1471-2105-6-51.
23. Rebholz-Schuhmann D, Kirsch H, Arregui M, Gaudan S, Riethoven M, Stoehr P. EBIMed–text crunching to gather facts for proteins from Medline. Bioinformatics. 2007;23(2):237–44. https://doi.org/10.1093/bioinformatics/btl302.
24. Binder JX, Pletscher-Frankild S, Tsafou K, Stolte C, O'Donoghue SI, Schneider R, Jensen LJ. Compartments: unification and visualization of protein subcellular localization evidence. Database (Oxford). 2014;2014: 012. https://doi.org/10.1093/database/bau012.
25. Stapley BJ, Kelley LA, Sternberg MJ. Predicting the sub-cellular location of proteins from text using support vector machines. Pac Symp Biocomput. 2002:374–85. https://www.ncbi.nlm.nih.gov/pubmed/11928491.
26. Fyshe A, Liu Y, Szafron D, Greiner R, Lu P. Improving subcellular localization prediction using text classification and the gene ontology. Bioinformatics. 2008;24(21):2512–7. https://doi.org/10.1093/bioinformatics/btn463.
27. Kim MY. Detection of protein subcellular localization based on a full syntactic parser and semantic information. In: 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 4; 2008. p. 407–11. https://doi.org/10.1109/FSKD.2008.529.
28. Kim JD, Ohta T, Pyysalo S, Tsujii YKJ. Overview of BioNLP'09 shared task on event extraction. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task. Boulder, Colorado: Association for Computational Linguistics; 2009. p. 1–9.
29. Kim JD, Wang Y, Takagi T, Yonezawa A. Overview of Genia event task in BioNLP Shared Task 2011. In: Proceedings of the BioNLP Shared Task 2011 Workshop. Portland, Oregon: Association for Computational Linguistics; 2011. p. 7–15.
30. Liu Y, Shi Z, Sarkar A. Exploiting rich syntactic information for relation extraction from biomedical articles. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers. Rochester: Association for Computational Linguistics; 2007. p. 97–100.
31. Zheng W, Blake C. Using distant supervised learning to identify protein subcellular localizations from full-text scientific articles. J Biomed Inform. 2015;57:134–44. https://doi.org/10.1016/j.jbi.2015.07.013.
32. Ioannidis JPA. Why most published research findings are false. PLoS Med. 2005;2(8):124. https://doi.org/10.1371/journal.pmed.0020124.
33. Horton R. Offline: What is medicine's 5 sigma? Lancet. 2015;385(9976): 1380. https://doi.org/10.1016/S0140-6736(15)60696-1.
34. Mullard A. Reliability of 'new drug target' claims called into question. Nat Rev Drug Discov. 2011;10(9):643–4.
35. Goldberg T, Vinchurkar S, Cejuela JM, Jensen LJ, Rost B. Linked annotations: a middle ground for manual curation of biomedical databases and text corpora. BMC Proc. 2015;9(Suppl 5):4–4. https://doi.org/10.1186/1753-6561-9-S5-A4.
36. Cejuela JM, Bojchevski A, Uhlig C, Bekmukhametov R, Kumar Karn S, Mahmuti S, Baghudana A, Dubey A, Satagopam VP, Rost B. nala: text mining natural language mutation mentions. Bioinformatics. 2017. https://doi.org/10.1093/bioinformatics/btx083.
37. Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, Simonovic M, Roth A, Santos A, Tsafou KP, Kuhn M, Bork P, Jensen LJ, von Mering C. STRING v10: protein-protein interaction networks, integrated over the tree of life. Nucleic Acids Res. 2015;43(Database issue):447–52. https://doi.org/10.1093/nar/gku1003.
38. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3): 273–97. https://doi.org/10.1007/BF00994018.
39. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É. Scikit-learn: Machine learning in python. J Mach Learn Res. 2011;12:2825–30.
40. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol. 2011;2(3):1–27. https://doi.org/10.1145/1961189.1961199.
41. Collins M, Duffy N. Convolution kernels for natural language. In: Proceedings of the 14th Conference on Neural Information Processing Systems. Collins:Duffy:01; 2001. http://books.nips.cc/papers/files/nips14/AA58.pdf. Accessed Apr 2017.
42. Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc.; 1999. p. 200–9. 657646.
43. Moschitti A. Making Tree Kernels Practical for Natural Language Learning. In: 11th Conference of the European Chapter of the Association for Computational Linguistics; 2006. p. 113–120. http://www.aclweb.org/anthology/E06-1015.
44. Wei CH, Harris BR, Kao HY, Lu Z. tmVar: a text mining approach for extracting sequence variants in biomedical literature. Bioinformatics. 2013;29(11):1433–9. https://doi.org/10.1093/bioinformatics/btt156.
45. Ng AY. Feature selection, L1 vs. L2 regularization, and rotational invariance. In: Proceedings of the Twenty-first International Conference on Machine Learning. ACM; 2004. p. 78. https://doi.org/10.1145/1015330.1015435.1015435.
46. Björne J, Heimonen J, Ginter F, Airola A, Pahikkala T, Salakoski T. Extracting complex biological events with rich graph-based feature sets. In: Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task. Association for Computational Linguistics; 2009. p. 10–18. 1572343.
47. Caporaso JG, Deshpande N, Fink JL, Bourne PE, Cohen KB, Hunter L. Intrinsic evaluation of text mining tools may not predict performance on realistic tasks; 2008. https://doi.org/10.1142/9789812776136_0061. Accessed Apr 2017.
48. Bairoch A, Apweiler R. The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. Nucleic Acids Res. 1999;27(1): 49–54.
49. Van Auken K, Schaeffer ML, McQuilton P, Laulederkind SJ, Li D, Wang SJ, Hayman GT, Tweedie S, Arighi CN, Done J, Muller HM, Sternberg PW, Mao Y, Wei CH, Lu Z. BC4GO: a full-text corpus for the BioCreative IV

GO task. Database (Oxford). 2014;2014:. https://doi.org/10.1093/database/bau074.
50. Cejuela JM, McQuilton P, Ponting L, Marygold SJ, Stefancsik R, Millburn GH, Rost B, FlyBase C. tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles. Database (Oxford). 2014;2014(0):033. https://doi.org/10.1093/database/bau033.

Supporting online material for:

LocText: relation extraction of protein localizations to assist database curation

Juan Miguel Cejuela, Shrikant Vinchurkar, Tatyana Goldberg, Madhukar Sollepura Prabhu Shankar, Ashish Baghudana, Aleksandar Bojchevski, Carsten Uhlig, André Ofner, Pandu Raharja-Liu, Lars Juhl Jensen, and Burkhard Rost

1. Short description of Supporting Online Material

Some results not shown in main paper but supporting some described findings.

2. Material

(starts in next page; one Table/Figure per page)

Fig. S1. Parse Tree Features. *LocText* derives features from parsed syntax trees: dependencies between tokens (e.g. "to" prepositions “the trans-Golgi network”), noun phrases (e.g. the AP-1 µ adaptin subunit”), part of speech tags (e.g. “localized” is a verb, past participle), linear distance between tokens (e.g. “adaptin” and “Golgi” are 10-tokens away from each other), or dependency distances between tokens (e.g. “adapting” and “Golgi” are 4-dependencies away).

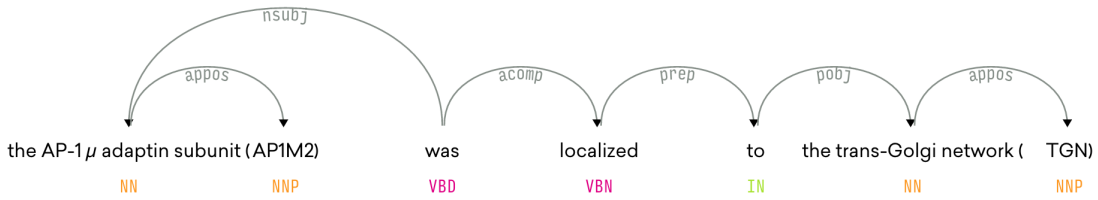


Table S1. Sample features after L1 regularization selection.

Feature name	**Explanation**
Total count of entities	Float (scaled [0, 1]), count of entities (proteins + localizations + organisms) in sentence-based instance.
Is Protein Marker	(Binary) test whether the marked protein text was equal to a list of manually defined protein markers: GFP, CYH2, ALG2, MSB2, KSS1, KRE11, SER2.
PROTEIN localize	Binary, Linear Dependency (LD) feature. Test if the sentence-based instance contains two consecutive tokens (2-gram), where the first token is part of an protein name and the following has lemma "localize" (e.g. localized).
SOURCE localization at	Binary, LD. Test if the first entity (source) in the sentence-based instance (either protein or location; likely in this case, a protein), is followed by the lemmas "localization" and "at".
SOURCE proliferation *TARGET*	Binary, Parsing Dependency (PD). Test if the source entity is connected to the second entity (target) by the lemma "proliferation" in the dependency parse tree.
in the *TARGET*	Binary, LD. Test if the lemmas "in" and "the" follow the target entity (likely a location).
SOURCE VERB DET	Binary, LD. Test if the source entity is followed by two tokens, the first a VERB, the next a determiner (DET)
VERB NOUN TARGET	Binary, PD. Test if, in the dependency parse tree, a VERB token connects a NOUN token that connects the target entity.
NSUBJ DOBJ PREP	Binary, PD. Test if, in the dependency parse tree, three tokens are connected with the dependencies NSUBJ (nominal subject) to (direct object) to PREP (prepositional modifier).

Table S2. Examples of novel GO annotations text-mined by *LocText*.

Protein	Localization	Text Source
P61586 human	GO:0005634 (nucleus)	Recent studies have revealed the localization of **RhoA** protein in the cell **nucleus**, in addition to its distribution in the cytosol and cell membrane. *PMID 26622605, 2015*
P09936 human	GO:0008021 (synaptic vesicle)	Both **synaptic vesicle markers** co-localized with the neuronal marker **PGP 9.5** and exhibited granular accumulation patterns in the **human** and rat ENS. *PMID 24025431, 2013*
P56817 human	GO:0005764 (lysosome)	Here, we report that lack of ubiquitination at Lys-501 (BACE1K501R) does not affect the rate of endocytosis but produces **BACE1** stabilization and accumulation of **BACE1** in early and late endosomes/**lysosomes** as well as at the cell membrane. *PMID 23109336, 2012*
P01149 yeast	GO:0005773 (vacuole)	[...] we provided evidence for the existence of an endocytic intermediate(s) from the **yeast Saccharomyces cerevisiae** that is responsible for the transport of the **pheromone alpha-factor** from the plasma membrane to the **vacuole**. PMID 8314797, 1993
P13134 yeast	GO:0005768 (endosome)	However, **Kex2** localization is not static, and its itinerary apparently involves transiting out of the late Golgi and cycling back from post-Golgi **endosomal** compartments during its lifetime. We tested whether the endocytic pathway could deliver small molecules to **Kex2** from the extracellular medium. Here we report that intramolecularly quenched fluorogenic substrates taken up into intact **yeast** revealed fluorescence due to specific cleavage by **Kex2** protease in **endosomal** compartments. *PMID 10393104, 1999*
Q9FE59 cress	GO:0009705 (plant-type vacuole membrane)	We demonstrate that this motif can reroute other proteins, such as INT4, **SUCROSE TRANSPORTER2 (SUC2)**, or SWEET1, to the **tonoplast** and that the position of the motif relative to the transmembrane helix is critical. *PMID 22253225, 2012*
O82533 cress	GO:0005829 (cytosol)	Here, we report the identification of a **second nuclear-encoded FtsZ-type protein** from **Arabidopsis** that does not contain a chloroplast targeting sequence or other obvious sorting signals and is not imported into isolated chloroplasts, which strongly suggests that it is localized in the **cytosol**. *PMID 9836740, 1998*

Fig. S2. List of all finally selected features. Python-readable list of descriptive feature names.

```
[
    "SentenceFeatureGenerator::1.1_counts_individual_int_individual_e_1_[0]",  # 0
    "SentenceFeatureGenerator::1.1_counts_individual_int_individual_e_3_[0]",  # 1
    "SentenceFeatureGenerator::3_order_[0]",  # 2
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_1_<NOUN>_[0]",  # 3
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<[SOURCE ~~ PUNCT>_[0]",  # 4
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PUNCT ~~ VERB>_[0]",  # 5
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<NOUN ~~ NOUN>_[0]",  # 6
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ NOUN ~~ PUNCT>_[0]",  # 7
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ PUNCT ~~ ADJ>_[0]",  # 8
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ PUNCT ~~ NOUN>_[0]",  # 9
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ NOUN ~~ NOUN>_[0]",  # 10
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<as>_[0]",  # 11
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<an>_[0]",  # 12
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_1_<ADP>_[0]",  # 13
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_1_<NUM>_[0]",  # 14
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_1_<DET>_[0]",  # 15
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<[SOURCE ~~ NOUN>_[0]",  # 16
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PUNCT ~~ DET>_[0]",  # 17
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_1_<VERB>_[0]",  # 18
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_1_<NOUN>_[0]",  # 19
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<nsubj>_[0]",  # 20
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<prep>_[0]",  # 21
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<NOUN ~~ NOUN>_[0]",  # 22
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<[SOURCE ~~ VERB ~~ ADP>_[0]",  # 23
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<VERB ~~ ADP ~~ NOUN>_[0]",  # 24
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<ADP ~~ NOUN ~~ NOUN>_[0]",  # 25
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<prep ~~ pobj ~~ appos>_[0]",  #
26
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_1_<PROPN>_[0]",  # 27
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_1 ~~ e_1>_[0]",  # 28
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<DET ~~ ADJ>_[0]",  # 29
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<NOUN ~~ PROPN>_[0]",  # 30
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PUNCT ~~ CONJ>_[0]",  # 31
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ CONJ ~~ DET>_[0]",  # 32
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<CONJ ~~ DET ~~ TARGET]>_[0]",  # 33
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<[SOURCE ~~ VERB>_[0]",  # 34
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ PUNCT ~~ TARGET]>_[0]",  # 35
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<amod ~~ appos>_[0]",  # 36
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<[SOURCE ~~ NOUN ~~ TARGET]>_[0]",  # 37
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ form ~~ form>_[0]",  # 38
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<,>_[0]",  # 39
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<[SOURCE ~~ ,>_[0]",  # 40
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<, ~~ TARGET]>_[0]",  # 41
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<to>_[0]",  # 42
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<exit>_[0]",  # 43
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<advcl>_[0]",  # 44
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<VERB ~~ VERB>_[0]",  # 45
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<pobj ~~ prep>_[0]",  # 46
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<prep ~~ advcl>_[0]",  # 47
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<in>_[0]",  # 48
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<ADP ~~ NOUN>_[0]",  # 49
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<ADP ~~ ADJ>_[0]",  # 50
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<[SOURCE ~~ VERB ~~ VERB>_[0]",  # 51
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ VERB ~~ ADP>_[0]",  # 52
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<dobj>_[0]",  # 53
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<nsubj ~~ advcl>_[0]",  # 54
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<dobj ~~ amod>_[0]",  # 55
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<VERB ~~ NOUN ~~ TARGET]>_[0]",  # 56
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<mutant>_[0]",  # 57
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_1_<SYM>_[0]",  # 58
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<, ~~ the>_[0]",  # 59
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<DET ~~ PROPN>_[0]",  # 60
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<the ~~ e_1 ~~ e_1>_[0]",  # 61
```

```
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<accumulate>_[0]",  # 62
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<VERB ~~ ADJ>_[0]",  # 63
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<ADP ~~ TARGET]>_[0]",  # 64
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<acl ~~ prep>_[0]",  # 65
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<[SOURCE ~~ NOUN ~~ ADP>_[0]",  # 66
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<NOUN ~~ ADP ~~ NOUN>_[0]",  # 67
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<ADP ~~ NOUN ~~ ADP>_[0]",  # 68
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<NOUN ~~ ADP ~~ VERB>_[0]",  # 69
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<pobj ~~ prep ~~ pobj>_[0]",  #
70
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<acl ~~ prep ~~ pobj>_[0]",  # 71
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<cell>_[0]",  # 72
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<with>_[0]",  # 73
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<the ~~ e_2>_[0]",  # 74
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_2 ~~ of>_[0]",  # 75
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_1 ~~ and>_[0]",  # 76
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<with ~~ the>_[0]",  # 77
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ ADP ~~ NOUN>_[0]",  # 78
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<ADP ~~ NOUN ~~ VERB>_[0]",  # 79
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ PROPN ~~ NOUN>_[0]",  # 80
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ ADP ~~ DET>_[0]",  # 81
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<nsubjpass>_[0]",  # 82
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ DET ~~ TARGET]>_[0]",  # 83
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<of>_[0]",  # 84
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<cell>_[0]",  # 85
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<[SOURCE ~~ of>_[0]",  # 86
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<NOUN ~~ VERB>_[0]",  # 87
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<prep ~~ pobj ~~ acl>_[0]",  # 88
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<dobj ~~ acl>_[0]",  # 89
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<acl ~~ pobj>_[0]",  # 90
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<amod ~~ prep>_[0]",  # 91
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<amod ~~ prep ~~ pobj>_[0]",  #
92
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<[SOURCE ~~ in>_[0]",  # 93
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<pobj ~~ compound>_[0]",  # 94
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<be>_[0]",  # 95
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<class>_[0]",  # 96
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<relate>_[0]",  # 97
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<on ~~ TARGET]>_[0]",  # 98
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<ADP ~~ TARGET]>_[0]",  # 99
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<[SOURCE ~~ VERB ~~ DET>_[0]",  # 100
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<be>_[0]",  # 101
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<NOUN ~~ NOUN ~~ VERB>_[0]",  # 102
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<NOUN ~~ CONJ>_[0]",  # 103
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<CONJ ~~ VERB>_[0]",  # 104
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ NOUN ~~ PUNCT>_[0]",  # 105
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<CONJ ~~ VERB ~~ VERB>_[0]",  # 106
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ ADP ~~ ADJ>_[0]",  # 107
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<nummod>_[0]",  # 108
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<VERB ~~ VERB ~~ ADP>_[0]",  # 109
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<expression>_[0]",  # 110
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ ADP ~~ TARGET]>_[0]",  # 111
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<) ~~ be>_[0]",  # 112
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PROPN ~~ PUNCT>_[0]",  # 113
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ ADJ ~~ NOUN>_[0]",  # 114
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ ADP ~~ VERB>_[0]",  # 115
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<from>_[0]",  # 116
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<acomp>_[0]",  # 117
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<advmod>_[0]",  # 118
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<pcomp ~~ prep>_[0]",  # 119
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<nsubj ~~ acomp ~~ prep>_[0]",  #
120
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<with ~~ e_1>_[0]",  # 121
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<ADP ~~ PROPN>_[0]",  # 122
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PROPN ~~ ADP>_[0]",  # 123
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<VERB ~~ PROPN>_[0]",  # 124
```

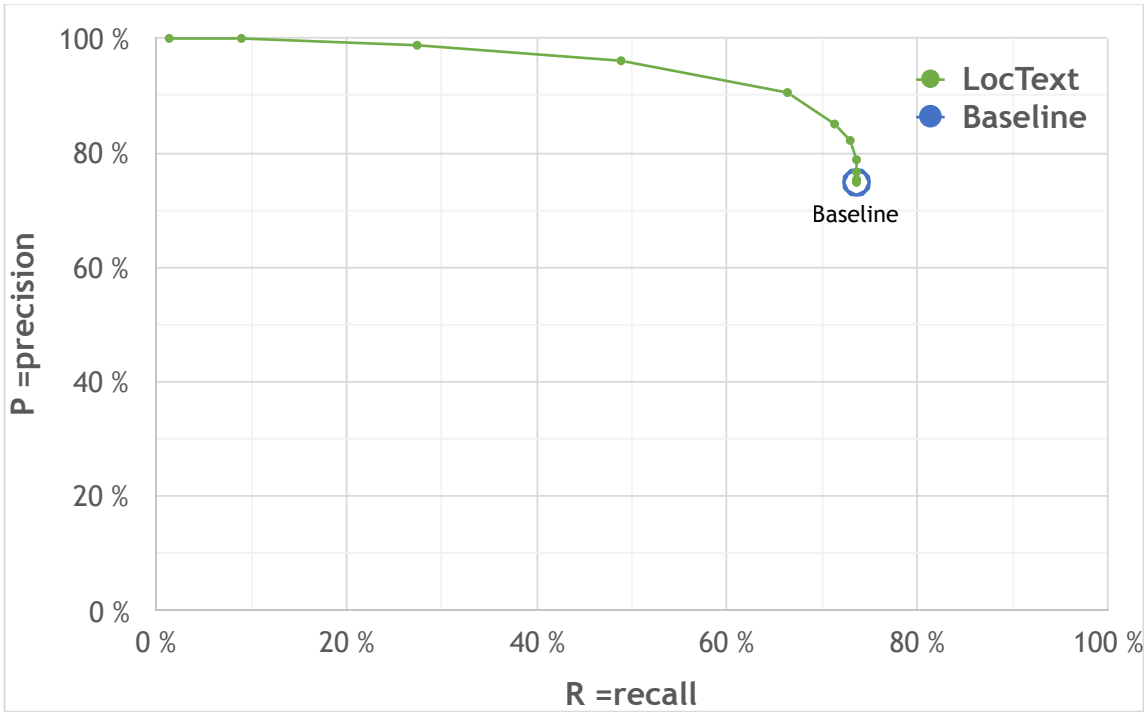
***LocText*: relation extraction of protein localizations to assist database curation**

```
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<[SOURCE ~~ colocalizes ~~ with>_[0]",  # 125
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NUM ~~ ADP ~~ DET>_[0]",  # 126
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ NOUN ~~ TARGET]>_[0]",  # 127
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<[SOURCE ~~ and ~~ e_1>_[0]",  # 128
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<[SOURCE ~~ PROPN>_[0]",  # 129
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<conj ~~ pobj>_[0]",  # 130
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<conj ~~ conj>_[0]",  # 131
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<[SOURCE ~~ in>_[0]",  # 132
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<[SOURCE ~~ NOUN ~~ TARGET]>_[0]",  # 133
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<transporter>_[0]",  # 134
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<that>_[0]",  # 135
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_1 ~~ ,>_[0]",  # 136
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<ADV ~~ VERB>_[0]",  # 137
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<e_1 ~~ e_1 ~~ ,>_[0]",  # 138
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<e_1 ~~ ( ~~ e_1>_[0]",  # 139
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ NOUN ~~ NUM>_[0]",  # 140
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ ADV ~~ VERB>_[0]",  # 141
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<NOUN ~~ PROPN>_[0]",  # 142
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<appos ~~ appos>_[0]",  # 143
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<[SOURCE ~~ PUNCT ~~ NOUN>_[0]",  # 144
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<both>_[0]",  # 145
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<e_3>_[0]",  # 146
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<localize ~~ to>_[0]",  # 147
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<of ~~ e_3>_[0]",  # 148
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PUNCT ~~ ADP>_[0]",  # 149
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<ADP ~~ PROPN ~~ NUM>_[0]",  # 150
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<pobj ~~ amod>_[0]",  # 151
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<prep ~~ pobj ~~ amod>_[0]",  #
152
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ NOUN ~~ PART>_[0]",  # 153
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<prep ~~ nsubj>_[0]",  # 154
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<ccomp ~~ dobj>_[0]",  # 155
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<pobj ~~ prep ~~ nsubj>_[0]",  #
156
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<with ~~ TARGET]>_[0]",  # 157
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<nmod>_[0]",  # 158
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_1_<agent>_[0]",  # 159
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_2_<ADV ~~ VERB>_[0]",  # 160
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<of ~~ a>_[0]",  # 161
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PROPN ~~ VERB>_[0]",  # 162
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<involve ~~ in>_[0]",  # 163
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<protein>_[0]",  # 164
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ NOUN ~~ ADP>_[0]",  # 165
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<protein>_[0]",  # 166
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<of ~~ activity>_[0]",  # 167
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<and ~~ the>_[0]",  # 168
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ PUNCT ~~ CONJ>_[0]",  # 169
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ NOUN ~~ NOUN>_[0]",  # 170
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<[SOURCE ~~ protein>_[0]",  # 171
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ VERB ~~ NOUN>_[0]",  # 172
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<conj ~~ nsubj>_[0]",  # 173
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_3 ~~ TARGET]>_[0]",  # 174
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<nsubj ~~ dobj>_[0]",  # 175
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ ADP ~~ NOUN>_[0]",  # 176
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<tag>_[0]",  # 177
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<advmod ~~ dobj>_[0]",  # 178
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<advmod ~~ dobj ~~
compound>_[0]",  # 179
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<gene>_[0]",  # 180
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<[SOURCE ~~ by>_[0]",  # 181
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ NOUN ~~ PROPN>_[0]",  # 182
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ PROPN ~~ PROPN>_[0]",  # 183
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<activate>_[0]",  # 184
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<require>_[0]",  # 185
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<biogenesis>_[0]",  # 186
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<modulates>_[0]",  # 187
```

```
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<glycosylation>_[0]",  # 188
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ ADJ ~~ NOUN>_[0]",  # 189
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<expression ~~ TARGET]>_[0]",  # 190
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<and ~~ TARGET]>_[0]",  # 191
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ CONJ ~~ TARGET]>_[0]",  # 192
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<NOUN ~~ NOUN ~~ ADP>_[0]",  # 193
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<DET ~~ ADV>_[0]",  # 194
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ VERB ~~ PART>_[0]",  # 195
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<be ~~ protein>_[0]",  # 196
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<induce>_[0]",  # 197
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_2 ~~ and>_[0]",  # 198
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<and ~~ inhibit>_[0]",  # 199
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<retention ~~ inhibit>_[0]",  # 200
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<ADP ~~ ADP ~~ TARGET]>_[0]",  # 201
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<nsubj ~~ dobj ~~ compound>_[0]",
# 202
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<compound ~~ conj>_[0]",  # 203
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<dobj ~~ advcl>_[0]",  # 204
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<pobj ~~ prep ~~ dobj>_[0]",  #
205
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<function ~~ TARGET]>_[0]",  # 206
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<VERB ~~ ADJ ~~ TARGET]>_[0]",  # 207
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<nmod ~~ pobj>_[0]",  # 208
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<ADP ~~ PROPN ~~ TARGET]>_[0]",  # 209
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<[SOURCE ~~ be ~~ require>_[0]",  # 210
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ ADV ~~ PUNCT>_[0]",  # 211
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<function>_[0]",  # 212
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<of ~~ e_1>_[0]",  # 213
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<- ~~ independent ~~ function>_[0]",  # 214
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ function ~~ require>_[0]",  # 215
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<of ~~ function>_[0]",  # 216
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ of ~~ function>_[0]",  # 217
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<pobj ~~ prep ~~ nsubjpass>_[0]",
# 218
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<pobj ~~ nmod>_[0]",  # 219
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<prep ~~ pobj ~~ nmod>_[0]",  #
220
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ ADP ~~ VERB>_[0]",  # 221
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<signal>_[0]",  # 222
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<[SOURCE ~~ signal>_[0]",  # 223
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<amod ~~ nsubj>_[0]",  # 224
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<show>_[0]",  # 225
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<dobj ~~ ccomp>_[0]",  # 226
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<ubiquitination>_[0]",  # 227
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<ubiquitination>_[0]",  # 228
    "DependencyFeatureGenerator::23_PD_pos_N_gram_PD_3_<[SOURCE ~~ ADP ~~ TARGET]>_[0]",  # 229
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<activation>_[0]",  # 230
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ VERB ~~ PROPN>_[0]",  # 231
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NUM ~~ NOUN ~~ ADP>_[0]",  # 232
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<region>_[0]",  # 233
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<ADJ ~~ ADP ~~ PROPN>_[0]",  # 234
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<nsubj ~~ advcl ~~
nsubjpass>_[0]",  # 235
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<e_1 ~~ , ~~ the>_[0]",  # 236
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<advcl ~~ prep>_[0]",  # 237
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<[SOURCE ~~ from>_[0]",  # 238
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<into ~~ TARGET]>_[0]",  # 239
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<ADP ~~ ADJ ~~ TARGET]>_[0]",  # 240
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<ADV ~~ TARGET]>_[0]",  # 241
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<this>_[0]",  # 242
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<DET ~~ ADJ ~~ ADJ>_[0]",  # 243
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<ADP ~~ PROPN ~~ PUNCT>_[0]",  # 244
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<compromise>_[0]",  # 245
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<fuse>_[0]",  # 246
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<acl ~~ acomp>_[0]",  # 247
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NOUN ~~ PUNCT ~~ PUNCT>_[0]",  # 248
```

```
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<vesicle>_[0]",  # 249
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<dispensable>_[0]",  # 250
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<be ~~ dispensable>_[0]",  # 251
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<dispensable ~~ for>_[0]",  # 252
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<be ~~ dispensable ~~ for>_[0]",  # 253
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<e_1 ~~ to>_[0]",  # 254
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<location>_[0]",  # 255
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<[SOURCE ~~ location>_[0]",  # 256
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<[SOURCE ~~ location>_[0]",  # 257
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<, ~~ to>_[0]",  # 258
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<, ~~ while>_[0]",  # 259
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_2_<appos ~~ compound>_[0]",  # 260
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<PUNCT ~~ VERB ~~ TARGET]>_[0]",  # 261
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<important>_[0]",  # 262
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<morphology>_[0]",  # 263
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<pobj ~~ amod ~~ npadvmod>_[0]",
# 264
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<defect>_[0]",  # 265
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<cohesin>_[0]",  # 266
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<assembly>_[0]",  # 267
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ determine ~~ localize>_[0]",  #
268
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<[SOURCE ~~ TARGET]>_[0]",  # 269
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<significantly>_[0]",  # 270
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<be ~~ significantly>_[0]",  # 271
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<at ~~ have>_[0]",  # 272
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ at ~~ have>_[0]",  # 273
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ recruitment ~~ of>_[0]",  # 274
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<prep ~~ nsubj ~~ prep>_[0]",  #
275
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<[SOURCE ~~ domain ~~ of>_[0]",  # 276
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<[SOURCE ~~ domain ~~ of>_[0]",  # 277
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<reconfiguring>_[0]",  # 278
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<division ~~ TARGET]>_[0]",  # 279
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<abundance ~~ TARGET]>_[0]",  # 280
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<eliminate>_[0]",  # 281
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<from ~~ through>_[0]",  # 282
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<from ~~ through ~~ TARGET]>_[0]",  # 283
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_1_<stress>_[0]",  # 284
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<[SOURCE ~~ stress>_[0]",  # 285
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_1_<stress>_[0]",  # 286
    "DependencyFeatureGenerator::26_PD_undirected_edges_N_gram_PD_3_<appos ~~ nsubj ~~ dobj>_[0]",  #
287
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_3_<localization ~~ disrupt ~~ by>_[0]",  # 288
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<e_1 ~~ to ~~ TARGET]>_[0]",  # 289
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<that ~~ u>_[0]",  # 290
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_2_<require ~~ e_1>_[0]",  # 291
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_2_<PRON ~~ NUM>_[0]",  # 292
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<show ~~ that ~~ u>_[0]",  # 293
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<first ~~ require ~~ e_1>_[0]",  # 294
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<ADP ~~ PRON ~~ NUM>_[0]",  # 295
    "DependencyFeatureGenerator::19_LD_pos_N_gram_LD_3_<NUM ~~ ADV ~~ VERB>_[0]",  # 296
    "DependencyFeatureGenerator::22_PD_bow_N_gram_PD_2_<show ~~ require>_[0]",  # 297
    "DependencyFeatureGenerator::18_LD_bow_N_gram_LD_3_<[SOURCE ~~ to ~~ mediate>_[0]",  # 298
    "IsSpecificProteinType::40_is_marker_[0]",  # 299
    "LocalizationRelationsRatios::50_corpus_unnormalized_total_background_loc_rels_ratios_[0]",  #
300
    "LocalizationRelationsRatios::58_SwissProt_normalized_exists_relation_[0]",  # 301
]
```

Fig. S3. PR-curve analysis. *LocText* vs. *Baseline*, using manually-annotated entities. The maximum recall for both methods is 74%. The *Baseline* is shown as a single point (no decision value). A two-sample two-tailed t-test was performed to determine whether the methods' difference in F-Measure, F(*LocText*)=79%±3 vs. F(*Baseline*)=74%±3, was significant. The t-statistic was significant at the 99% confidence level, t(3998)=28.04, p=3.99e-165.



4.3 References

Ashburner, M. et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium". In: *Nat Genet* 25.1, pp. 25–9. ISSN: 1061-4036 (Print) 1061-4036 (Linking). DOI: 10.1038/75556. URL: https://www.ncbi.nlm.nih.gov/pubmed/10802651.

Boutet, E. et al. (2016). "UniProtKB/Swiss-Prot, the Manually Annotated Section of the UniProt KnowledgeBase: How to Use the Entry View". In: *Methods Mol Biol* 1374, pp. 23–54. ISSN: 1940-6029 (Electronic) 1064-3745 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/26519399.

Cejuela, J. M. et al. (2014). "tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles". In: *Database (Oxford)* 2014.0, bau033. ISSN: 1758-0463 (Electronic) 1758-0463 (Linking). DOI: 10.1093/database/bau033. URL: https://www.ncbi.nlm.nih.gov/pubmed/24715220.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018. URL: http://dx.doi.org/10.1007/BF00994018.

Goldberg, Tatyana et al. (2015). "Linked annotations: a middle ground for manual curation of biomedical databases and text corpora". In: *BMC Proceedings* 9.Suppl 5, A4–A4. ISSN: 1753-6561. DOI: 10.1186/1753-6561-9-S5-A4. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4582921/.

Goldberg, T. et al. (2014). "LocTree3 prediction of localization". In: *Nucleic Acids Res* 42.Web Server issue, W350–5. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gku396. URL: https://www.ncbi.nlm.nih.gov/pubmed/24848019.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. 655813: Morgan Kaufmann Publishers Inc., pp. 282–289.

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Ng, Andrew Y. (2004). "Feature selection, L1 vs. L2 regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*. 1015435: ACM, p. 78. DOI: 10.1145/1015330.1015435.

Sayers, E. W. et al. (2010). "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Res* 38.Database issue, pp. D5–16. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkp967. URL: https://www.ncbi.nlm.nih.gov/pubmed/19910364.

Sherry, S. T. et al. (2001). "dbSNP: the NCBI database of genetic variation". In: *Nucleic Acids Res* 29.1, pp. 308–11. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/11125122.

Simpson, J. C. and R. Pepperkok (2003). "Localizing the proteome". In: *Genome Biol* 4.12, p. 240. ISSN: 1474-760X (Electronic) 1474-7596 (Linking). DOI: 10.1186/gb-2003-4-12-240. URL: https://www.ncbi.nlm.nih.gov/pubmed/14659010.

The UniProt Consortium (2017). "UniProt: the universal protein knowledgebase". In: *Nucleic Acids Res* 45.D1, pp. D158–D169. ISSN: 1362-4962 (Electronic) 0305-1048 (Linking). DOI: 10.1093/nar/gkw1099. URL: https://www.ncbi.nlm.nih.gov/pubmed/27899622.

UniProt, Consortium (2015). "UniProt: a hub for protein information". In: *Nucleic Acids Res.* 43.Database issue, pp. D204–12. ISSN: 0305-1048. DOI: 10.1093/nar/gku989. URL: http://dx.doi.org/10.1093/nar/gku989%20http://www.ncbi.nlm.nih.gov/pubmed/25348405%20http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4384041%20http://nar.oxfordjournals.org/cgi/pmidlookup?view=long&pmid=25348405.

Chapter 5

Conclusions

In this work, I discussed techniques of text mining specific to biomedicine, and studied three newly-developed methods that were proven useful in practice to help us navigate the deluge of literature data. The three methods shared the same characteristic: they combined automatic annotations made by machines together with the careful expertise of users.

With this semi-automatic approach, I (and my co-authors) were able to (peer-reviewed): 1) Create one of the largest corpora of labeled data for NLP to date, with 451 annotated full-text articles. Professional curators at FlyBase, the premier database of the model organism *Drosophila melanogaster* (common fruit fly), supervised and/or performed the annotations, and all these were readily useful and added to the database. 2) Demonstrate with two independent curators, that the semi-automatic annotation assisted by the here designed, interactive web interface, *tagtog*, was up to 2 times faster than manual annotation alone. 3) Create the largest resource of genetic mutations as described in scientific publications. 4) Develop a new method, *nala*, that superseded the results of all existing solutions in finding mutation descriptions, and discovered up to 33% more, previously unhidden genetic variations. 5) Develop a new method, *LocText*, that is highly accurate (65%-85%) in discovering novel, functional annotations of protein subcellular localization from PubMed, the search engine for biomedical literature. We showed that our system could assist database curators, and yield over one hundred new annotations, per employee per work day. 6) Text-mine and assert 46 protein localization annotations, which were previously unknown to the highly-accurate UniProtKB/Swiss-Prot database, the universal protein knowledgebase. – We made all our methods and datasets available: *tagtog* `http://tagtog.net`, *nala* `http://github.com/Rostlab/nala`, and *LocText* `http://github.com/Rostlab/LocText`.

Our era of never-ending information demands automatic solutions that work at scale. It is my belief, that this automatic power is best reaped when is intertwined with the expertise of humans. For this work, at least, the guidance and supervision inputted by users were essential for our discoveries. Likewise, in this work, the automatic machinery of our text mining methods complemented, not substituted, the labor of experts, *viz.* database curators.