Neuroinformatics (2019) 17:361–371
https://doi.org/10.1007/s12021-018-9403-z

SOFTWARE ORIGINAL ARTICLE

Automated Metadata Suggestion During Repository Submission

Robert A. McDougal[1,2] · Isha Dalal[3] · Thomas M. Morse[1] · Gordon M. Shepherd[1]

Published online: 31 October 2018

Abstract

Knowledge discovery via an informatics resource is constrained by the completeness of the resource, both in terms of the amount of data it contains and in terms of the metadata that exists to describe the data. Increasing completeness in one of these categories risks reducing completeness in the other because manually curating metadata is time consuming and is restricted by familiarity with both the data and the metadata annotation scheme. The diverse interests of a research community may drive a resource to have hundreds of metadata tags with few examples for each making it challenging for humans or machine learning algorithms to learn how to assign metadata tags properly. We demonstrate with ModelDB, a computational neuroscience model discovery resource, that using manually-curated regular-expression based rules can overcome this challenge by parsing existing texts from data providers during user data entry to suggest metadata annotations and prompt them to suggest other related metadata annotations rather than leaving the task to a curator. In the ModelDB implementation, analyzing the abstract identified 6.4 metadata tags per abstract at 79% precision. Using the full-text produced higher recall with low precision (41%), and the title alone produced few (1.3) metadata annotations per entry; we thus recommend data providers use their abstract during upload. Grouping the possible metadata annotations into categories (e.g. cell type, biological topic) revealed that precision and recall for the different text sources varies by category. Given this proof-of-concept, other bioinformatics resources can likewise improve the quality of their metadata by adopting our approach of prompting data uploaders with relevant metadata at the minimal cost of formalizing rules for each potential metadata annotation.

Keywords Metadata · Repository · Data sharing · Natural language processing

Introduction

Online neuroinformatics databases aggregate data to facilitate discovery of individual data items as well as relations between different data sets. Metadata, data that describes the data, provides searchable context that plays a key role in the discovery process. The assignment of relevant metadata was identified as the second (F2) principle of FAIR data sharing, after having a unique and persistent identifier (Wilkinson et al. 2016).

✉ Robert A. McDougal
robert.mcdougal@yale.edu

[1] Department of Neuroscience, Yale University, 333 Cedar Street, PO Box 208001, New Haven, CT 06520-8001, USA

[2] Yale Center for Medical Informatics, Yale University, 300 George Street, New Haven, CT 06510, USA

[3] Yale College, Yale University, New Haven, CT 06520, USA

Assigning rich and accurate metadata (FAIR principle R1) in a repository of user-submitted data is non-trivial. The data producer is the person most familiar with the data, but they are not in general experts on any given repository's metadata scheme. Asking them to suggest metadata annotations from a set of hundreds of potential values imposes a barrier to data submission, and many scientists are already hesitant to share data (Ascoli 2015). Entry by the curator is less effective since the curator is less familiar with the model. Worse, curator-entry does not scale. Howe et al. (2008) proposes a middle path – community curation – that avoids the scaling issue but potentially suffers from unfamiliarity with both the data and the metadata scheme.

Automated or semi-automated curation is an obvious solution, but requires choosing the right tool for the task. Modern statistical text-mining approaches have shown promise when the data is in manuscripts on topics in specific biological domains; for example, French et al. 2015 and Richardet et al. 2015 both describe tools for mining neuroanatomy from publications. Topic modeling (e.g. Wallach 2006) identifies the

main topics of a paper, but cannot identify the single-mention factoids (e.g. brain region X is connected to brain region Y) or metadata (a given model cell includes a sodium channel) of interest to many data repositories. Indeed, many such repositories face additional challenges to employing fully automated annotation, such as covering a field that is simultaneously broad (e.g. all of computational neuroscience) requiring many different metadata tags, but also shallow (too few entries on one or more topics for robust training of classifiers).

We demonstrate with ModelDB (modeldb.yale.edu; McDougal et al. 2017) that a rule-based system can rapidly suggest metadata annotations during the process of submission from a large set of possible metadata tags. ModelDB is a computational neuroscience repository with the source code for over 1300 models of nerve cells and neural circuits; the data archived in ModelDB (the model) is required to be used in a publication but it is generally not included in full in the publication. No metadata is required from the submitter beyond their name and the citation of an associated paper. Submitters are, however, encouraged to supply metadata annotations describing the model in each of ten broad categories (what types of cells are modeled, what biological topics are investigated, etc). We have been told by some submitters that providing such annotations makes model sharing less appealing due to the extra effort required and due to being unsure of what information to provide. To address these challenges, a rule-based metadata predictor has been integrated into the ModelDB model submission process as an optional but recommended step. Model entry now proceeds by collecting basic information (contact information and citation), requesting the abstract, confirming or denying predicted metadata tags (supplied in under a second and thus not breaking workflow), and optionally manually adding more metadata tags. Although ModelDB provides suggested metadata tags, the submitter remains in control to authorize associating each tag and to specify additional metadata. We note that although our rules are necessarily domain-specific, the general strategy of interactively suggesting relevant metadata annotations during data submission could readily be adopted by other repositories, giving their curators a richer set of metadata on which to build.

Methods

Corpus Creation

We gathered plain-text versions of all abstracts (1164 in total) of published papers associated with models in ModelDB as of March 5, 2016. For those papers not having abstracts, an excerpt of the first page was used instead. The unit of data in ModelDB is a model, but since model codes are typically not formally published themselves, all metadata prediction is done based on the text of an associated publication.

Regular-Expression Rule Creation

As many metadata tags were associated with low numbers of model entries (e.g. only 4 models were tagged as being about "circadian rhythms"), we used a semi-manual rule-building process instead of using machine-learning.

Abstracts were fully manually reviewed in sequential order of ModelDB accession number to identify novel *n*-grams (sequences of words) of length 1–5 that are associated with existing ModelDB metadata tags. No longer phrases recurred between different abstracts. An *n*-gram was considered to be *associated* with a metadata tag if it represents the same concept or if it represents a concept that implies with high likelihood a ModelDB metadata tag concept as determined manually by the authors based on tag definitions and existing usage within ModelDB. For example, a model simulating the action of tetrodotoxin (TTX) likely simulates sodium currents as TTX is a sodium channel blocker. Such models would be identified using the regular expression "tetrodoxin|ttx" and annotated with "I Sodium". Approximately half of these abstracts were reviewed without reference to the existing associated ModelDB metadata tags. A Python script was used to generate HTML representations of each abstract highlighting text matched by a previously identified *n*-gram pattern to accelerate the abstract review by avoiding duplicative effort. This process was repeated until identification of novel *n*-gram patterns from a given abstract review was rare, at which point a separate process described below was employed to review the remaining abstracts. This threshold, assessed manually by authors ID and RAM, occurred around ModelDB accession number 60,000 by which point 236 abstracts associated with 193 models had been reviewed.

Identified *n*-grams were encoded in generalized, regular expression-like forms uniting spelling variants, number, tense, and abbreviations. Our regular expression-like representation followed the Python convention for regular expressions except we treat the $ character as a separator for multiple patterns required within nearby text (for this study, the patterns all had to occur within 5 contiguous words), and we treat all lowercase patterns as case-insensitive with the rest as case-sensitive. For example, the regular expression "(ca2?|calcium) (activated|activation of|dependent) (k|potassium)" detects several variant phrasings indicating the presence of calcium activated potassium currents. Matching multiple nearby patterns makes the system robust to changes in order; for example, "geniculate nucleus$interneurone?s?" matches both "interneurons of the geniculate nucleus" and "geniculate nucleus interneurones." Given that in practice nearby terms were expected to handle word order variants and filler words, and given that our patterns had as many as five words in them, we

chose a cutoff threshold of five contiguous words, matching the length of the longest repeated phrases found in different abstracts.

The generalized regular expression rules were stored as keys in a JavaScript Object Notation (JSON; Crockford 2006) file where the associated values are lists of object identifiers (primary keys) for the corresponding ModelDB metadata tags. During development of the ruleset, the names of the metadata were used instead of the corresponding identifiers for ease of verification. A Python script later automatically replaced these with the identifiers. Each regular expression-like encoding and the pairings with ModelDB metadata was reviewed by two of the authors (ID and RAM). As the patterns were being identified, a Python script was used to check for duplicate keys. All such duplicate entries were merged.

To complete the development of our rule set, we focused on identifying predictors for the 356 metadata tags not associated with any of the first 193 models in ModelDB and on the 387 tags associated with fewer than 5 models. Note that tags appearing late are not necessarily rare and may reflect the use of new technologies (e.g. Python) or new domains becoming tractable to model (e.g. Parkinson's). A document was generated for manual review with the abstracts and the associated priority tags. Authors ID and RAM manually reviewed this document to identify text patterns within the abstracts that implied the metadata tags under review, and then entered the text patterns into the JSON rule set as before.

Text Analysis

Text analysis is performed by a Python script running on the ModelDB server. The script tokenizes the text to be analyzed, using the regular expression [a-zA-Z0–9]+. It generates a processed text containing all the tokens with no punctuation and list of all nearby token sets (here taken to be all possible lists of 5 consecutive tokens, but in principle other rules could be used such as grouping by sentence). The analysis script then loops over every rule. No further processing is done for rules that, if satisfied, would not provide any new metadata terms. For the remaining rules, pattern components (portions of the pattern string split at $ symbols; see the definition of $ in the previous section) are tested as described below against the processed text using re.search. If any pattern component is not found in the processed full text, then the rule does not apply. If there is only one pattern component (i.e. no $ in the pattern) and it is found in the processed text, then the rule applies. If there are multiple pattern components and they are all found in the processed text, then each of the nearby token sets is tested to see if they satisfy every pattern component. If any do, then the rule applies. If the rule applies, then the set of predicted metadata tags is augmented by the tags predicted by the rule. Short-circuit evaluation is used at each stage; e.g. if one pattern component is not found in the processed text, then no further components are tested. Performance was measured on a 2.8 GHz i7 MacBook Pro from mid-2015 running macOS Sierra with 16 GB of memory.

Testing a text against a pattern component begins by augmenting both with leading and trailing spaces to ensure the matches align with word boundaries. Pattern components containing only lowercase characters are prefaced with (?i) which indicates case-insensitive search. A standard Python regular expression search (re.search) is then used to determine if the text matches the pattern component.

Comparisons to ModelDB are based on a snapshot from August 2016, shortly before the first version of the metadata predictor tool was completed.

An overview of the analysis process is shown in Fig. 1.

Online Tool

We integrated the metadata predictor tool into the model submission process by adding a Bootstrap (getbootstrap.com) styled button labeled "Let us find ModelDB keywords for you!" just below the required data fields. Clicking on the button opens a Bootstrap dialog that prompts for the abstract's text; once the "submit" button is selected, jQuery (jquery.com) performs a post request to query the analysis script on the server. The analysis script returns a JSON-encoded list of predictions, where each prediction is itself a list containing the name of the predicted metadata term (e.G. *hippocampus* CA1 Pyramidal Cell), an identifier for the metadata category (e.g. 25 to indicate a cell type), and the object identifier for the metadata (in our example, 258 indicates a Hippocampus CA1 Pyramidal Cell). As only the last piece of data is stored in the JSON rule set, the rest is looked up using a cached set of data that is automatically updated every time the database is updated. These names are used to populate a set of checkboxes in a new Bootstrap dialog that prompt the user to confirm each prediction. The categories, identifiers, and names for confirmed metadata predictions are used to populate ModelDB's existing metadata entry fields, which may then be manually populated with additional information.

The process of model submission using the metadata predictor is illustrated in Fig. 2. The submitter clicks the prominent "submit model" link on the ModelDB homepage, enters minimal required information (contact information, model files, and an access code), clicks a button, enters their abstract or other text, is presented with a list of suggested metadata, selects which metadata suggestions to accept, the selected metadata populates lists grouped by type (e.g. cell types vs genes vs …), and additional metadata may be entered into each list. This contrasts with the prior approach where no examples were shown and the user is faced with 10 blank lists to populate.

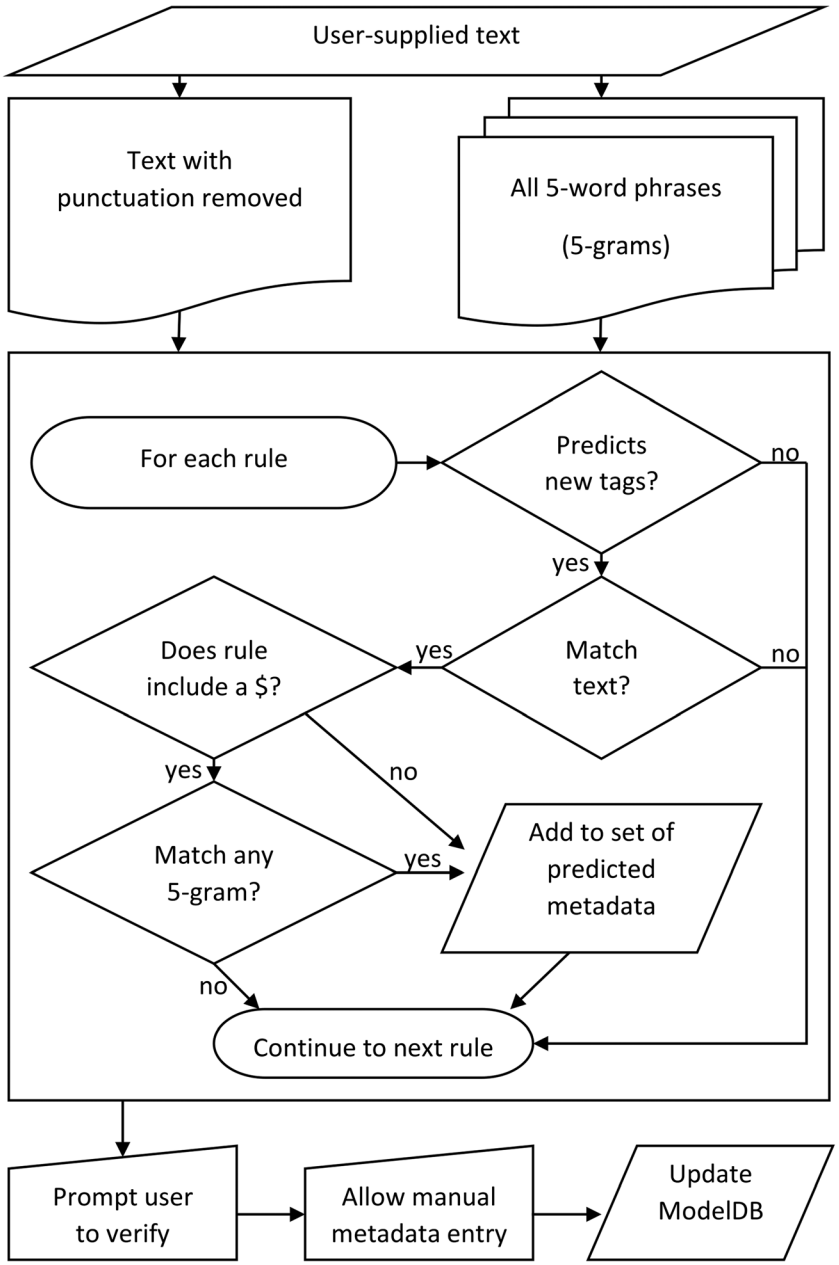

Fig. 1 Schematic of the text analysis process. When text is submitted as part of the model curation process, the punctuation is removed and the text is broken into 5-word phrases. For performance reasons, only rules that would predict new metadata are tested and those are first tested against the full-text and only later against individual phrases if the full-text matches. In production, predicted metadata is then human reviewed and augmented

Results

Validation of Initial Rule Set

The initial manually derived ruleset consisted of 1220 regular-expression-based patterns which predict 598 distinct ModelDB metadata tags (Hippocampus CA1 Pyramidal Neurons, Alzheimer's, …). 597 of the patterns used regular expressions; the rest look for an exact phrase. The patterns predicted on average 1.177 ± 0.436 tags; for example the pattern (central pattern generator|cpg)s? predicts three ModelDB tags: "Activity Patterns", "Temporal Pattern Generation", and "Oscillations". The tags predicted by the most rules were 3 ion channel currents ("I" denotes current) – "I Potassium" (76 rules), "I Calcium" (39), "I Sodium" (32) – and "Network" (25). The distribution of the rule predictions by ModelDB metadata categories is shown in Table 1. The metadata category most commonly used in ModelDB, model concept – the biological topic investigated by a model (e.g. "Alzheimer's", "Spike Timing Dependent Plasticity", etc), is also the category with the most predictor rules. Cell type, the category with the second highest number of predictor rules, is also the category with the most options. The distribution of predictor rules by metadata category thus loosely reflected existing usage and category complexity within ModelDB.

Initial validation of the rule set was performed by applying the algorithm to every abstract in our corpus. The results were compared with the existing ModelDB metadata, and 75

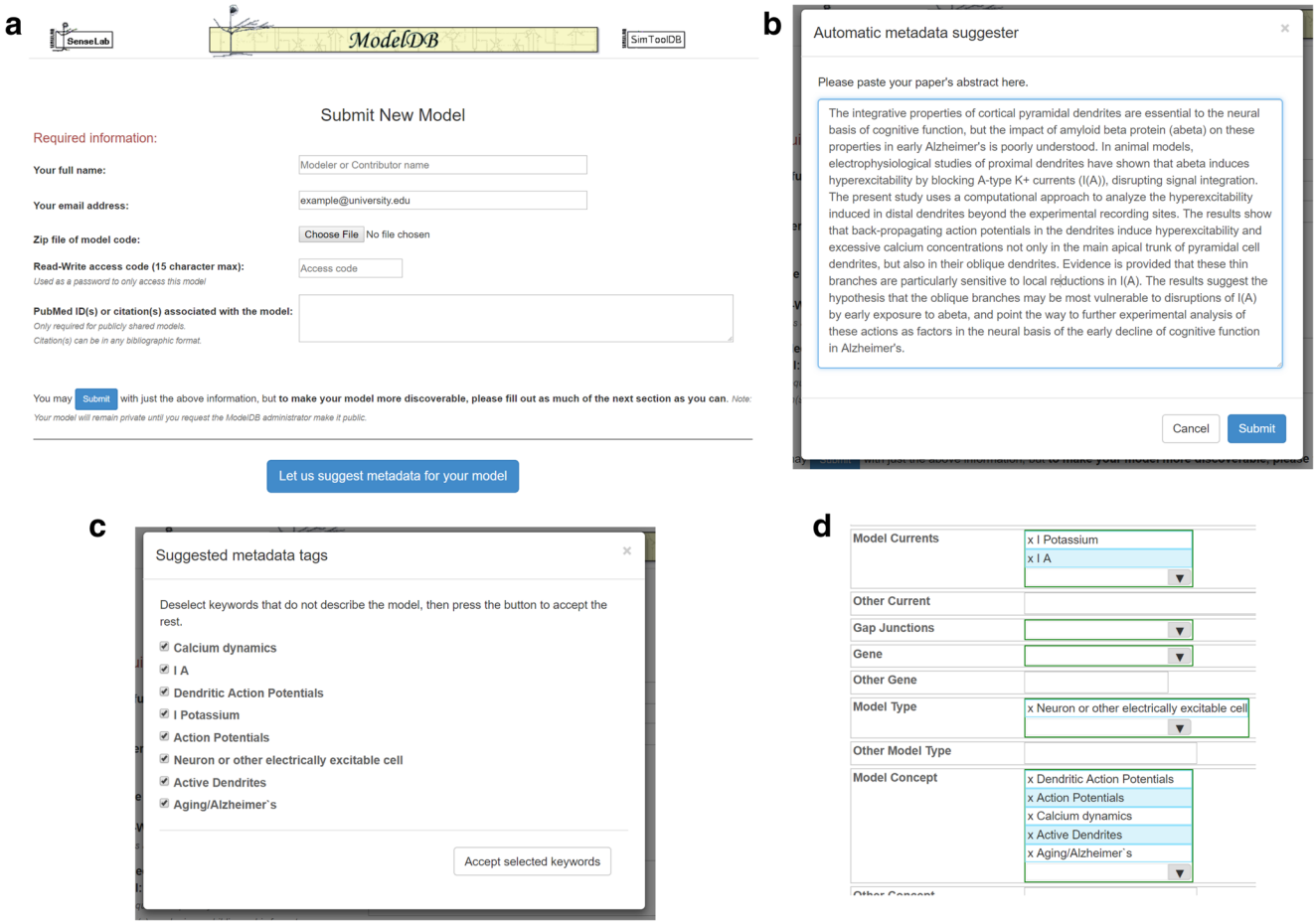

Fig. 2 The process of submitting a model to ModelDB begins by clicking the blue "submit model" button on the ModelDB homepage (modeldb.yale.edu), not shown. **a** A new page opens, prompting for contact information, an access code, the model code, and a citation. **b** Clicking the "let us suggest metadata for your model" button opens a dialog where the submitter may enter text describing their model, typically copy-pasting an associated paper's abstract. **c** A new dialog opens, suggesting possibly relevant metadata annotations, each of which can be confirmed or denied independently. **d** Accepted metadata suggestions appear in categorized manual metadata entry lists, which can be used to remove or add additional metadata tags. Clicking submit on this page enters the model into the system as a private model. When the model author is satisfied with the state of their model entry, they can contact the curator to make the model entry public (not shown). The abstract text used here is from Morse et al. 2010 and is used with permission

Table 1 Distribution of rule predictions by ModelDB metadata categories compared to the number of valid metadata values and the number of times a piece of metadata of that category is associated with a ModelDB entry

Category	Predictors	Metadata options	ModelDB usage (% models)
Model type	50	14	1320 (94.8%)
Cell type	360	188	1252 (69.7%)
Channel	268	53	3136 (58.6%)
Transmitter	42	21	361 (20.3%)
Receptor	62	55	756 (28.4%)
Gap junction	2	1	51 (4.6%)
Gene	53	45	151 (5.2%)
Region	96	33	426 (34.5%)
Concept	394	135	3217 (94.5%)
Simulation environment	109	98	1371 (99.7%)

Each prediction is counted separately; that is, a rule that predicts 3 model concepts contributes a count of 3. Percentages in the last column indicate the percentage of models that have metadata of the specified category. Metadata options and usage as of mid-2016

instances of novel predicted metadata tags (i.e. those not matching pre-existing metadata) were selected at random. Of those 75, a review by RAM found 54 (72%) were correct, a further 10 (13.3%) subject to interpretation or plausible but beyond what is stated in the abstract alone, and only 11 (14.7%) incorrect.

For this manuscript, we consider a metadata prediction to be *correct* if a manual review by one of the authors (TMM or RAM) deemed it to accurately describe the accompanying model. We consider a metadata prediction to be *contextual* if it describes the biological context the model is used in, but not a property of the model itself. Such contextual metadata predictions occur because we are analyzing a paper associated with a computational model instead of the model itself, which is expressed mathematically or in code with limited information about the biology it represents. A metadata prediction was considered *incorrect* if it was neither correct or contextual, as defined above.

After examining the incorrect predictions, the rules were adjusted in a way that eliminated four incorrect predictions: a typo in the rules that related the word "topographic" with the Topographica simulator was fixed and 4 rules were deemed too broad and removed (e.g. a mention of wakefulness does not imply the model is about sleep). With these changes, the ruleset was reduced to 1216 rules which are used for the remainder of the analysis.

Comparing Predictive Power by Text Source

To identify an appropriate choice of text to provide to the analysis tool, ten models from ModelDB were chosen at random by the computer; they are listed in Table 2. For each selected model, the analysis tool separately processed the title, abstract, and full-text. The full-text used included the abstract and figure captions but did not include the title or references.

Table 2 The ten models and accompanying papers whose titles, abstracts, and full-text were used for analyzing the metadata predictor algorithm

ModelDB ID	Paper
3812	Anderson et al. 1999
36,834	Heinz et al. 2001
97,903	Cornelisse et al. 2007
112,834	Wolf et al. 2005
116,386	Prescott et al. 2008
138,379	Neymotin et al. 2011
139,655	Kim et al. 2011
147,929	Rishikesh and Venkatesh 2003
151,949	Sousa et al. 2014
168,861	Garcia-Grajales et al. 2015

The models were selected randomly by a computer

Analysis of the 10 titles completed in a total of 1.65 s, of the 10 abstracts in a total of 1.87 s, and of the 10 full-texts in a total of 9.88 s. (Stated runtimes are the best of 5 runs to minimize the contributions of background system processes.) Thus analysis of any length of text associated with a publication is on average below the estimated 1 s threshold required for uninterrupted "flow of thought" (Nielsen 1993).

Using the full-text provided the most and the title provided the least predicted metadata tags both with respect to all tags and with respect to the correct tags; the full-text also had the highest false positive rate, and the title had the lowest (Fig. 3). All of the metadata predicted based on paper title was deemed to be correct, however only an average of 1.3 metadata tags (range: 0 to 5) were predicted per paper using this method. Using the abstract predicted an average of 6.2 metadata tags (range: 1 to 14) per paper, of which 79% were correct (49 out of 62). Using the full-text predicted an average of 28.8 metadata tags (range: 3 to 61), of which 41% were correct (118 out of 288). While full-text predicted more than 4.5x as many metadata tags as the abstract, it only made less than 2.5x as many correct predictions. A manual review of the metadata predictions from full-text before entering them into the database would thus require rejecting on average 17.0 tags per model, while manual review of abstract-based metadata predictions would only require rejecting on average 1.3 tags per model.

The prior manual curation exceeded the number of predicted metadata tags from the abstract for all but two of the ten papers (Cornelisse et al. 2007 and Kim et al. 2011), but provided fewer metadata tags than were predicted by the full text for all but Heinz et al. 2001. For six of the papers the number of correctly predicted pieces of metadata from the full text exceeded the number of pieces of metadata in ModelDB, with a seventh paper exceeding the ModelDB level if contextual predictions are included. Many of the correctly predicted metadata were novel, even in the case of metadata predicted from the abstract, where 56.5% of the predicted metadata was not previously in ModelDB. Of those, 62.8% were correct, 22.8% borderline, and 14.3% incorrect (the numbers do not add up to 100% due to rounding). Applying our algorithm to the abstracts correctly identified 2.2 metadata tags per model that were not previously identified by manual curation; while incorrect and borderline metadata tags on this set were in the minority, they occurred on average more than once per model.

To determine if some categories of metadata could be predicted more accurately than others, we compared prediction precision for each of the three text sets (title, abstract, full-text) for nine of the ten metadata categories shown in Table 1: Model Type, Brain Region/Organism, Cell Type, Channel, Receptor, Gene, Transmitter, Simulation Environment, and Model Concept. Gap junctions are omitted from this analysis as gap junctions were not predicted to be present for any of the ten models reviewed. As previously noted, all of the 13

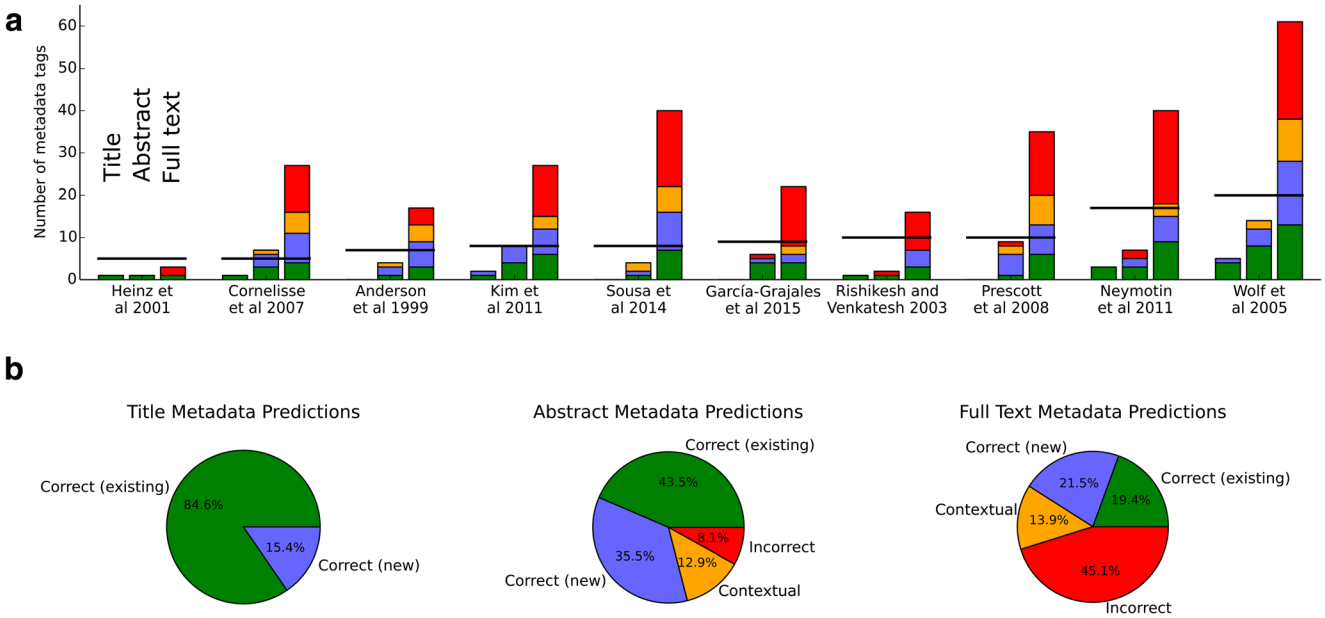

Fig. 3 Metadata prediction analysis comparing predictions from titles, abstracts, and full text for ten different randomly selected models from ModelDB. **a** For each model, the left, middle, and right columns represent metadata predicted based on the paper title, abstract, and full-text, respectively. Green indicates correctly predicted metadata already in ModelDB, blue indicates novel, correctly predicted metadata, orange indicates metadata describing the context of the problem but not necessarily for the model, and red indicates predicted metadata tags that are not appropriate for the model. Horizontal lines indicate the number of metadata tags previously entered into ModelDB for the given model. **b** Aggregated data across all examined models comparing the metadata prediction precision based on title vs abstract vs full text

metadata predictions from the titles were correct; these predictions were confined to four categories: receptors, model concepts, cell types, and brain regions. The majority of metadata predicted by the algorithm from the abstracts in 7 of the 8 categories for which metadata was predicted was correct; the only exception was the single predicted transmitter, which was incorrect. No genes were predicted. Only for channels ($n = 44$), genes ($n = 2$), cell types ($n = 20$), and simulation environment ($n = 12$) were at least half of the metadata tags predicted from the full-text correct. That is, most predictions in most categories of metadata were incorrect when analyzing full-text while most predictions in most categories of metadata were correct when analyzing abstracts.

For each text source, model concepts were the most common category of predicted metadata tags (title: 8/13, abstract: 31/62, full-text: 119/288); as shown in Table 1, model concepts were also the most common category of manually curated metadata tags in ModelDB. For the abstracts, 21 of those predictions were unambiguously correct; although analyzing the full-text added 88 more model concept predictions, it only added 13 correct predictions. Five brain regions were correctly identified from the abstracts; only one extra brain region was correctly identified when using the full-text at the cost of an additional 19 incorrect brain region identifications. By contrast, using the full-text increased the number of correctly identified simulation environments by a factor of 3 (from 2 to 6) while keeping the error rate no more than 50%. A comparison of the metadata identification precision from abstracts vs full-text is shown in Fig. 4. In brief, this shows that for our test set, the predictive utility would have been improved if certain categories of metadata (e.g. brain regions) were only predicted based on the abstract while others (e.g. channels and simulator environments) were predicted based on the full-text.

Analysis of Abstract-Based Predictions

For the remainder of our analysis, we used predictions based on the abstract text, as titles were deemed to predict too few metadata tags, and full text was deemed to have an unacceptably high false positive rate.

We ran the predictor algorithm on all 1164 abstracts in our data set. Using the algorithm as described in the methods section on the MacBook Pro, this process took 219.7 s to complete (188.7 ± 268.2 ms per abstract). Testing every rule regardless of whether or not it would lead to previously predicted metadata tags increased execution time to 232.4 s (199.7 ± 12.7 ms per abstract). Always testing the nearby token sets (in this paper, all 5-g) instead of testing with the full-text first increased execution time to 726.9 s (624.5 ± 155.3 ms per abstract). Without either optimization, execution time was 734.5 s (631.0 ± 162.6 ms per abstract). All individual abstract analysis times are mean ± standard deviation. Thus, our two performance optimizations reduced abstract analysis time by approximately 70% relative to the naïve algorithm.

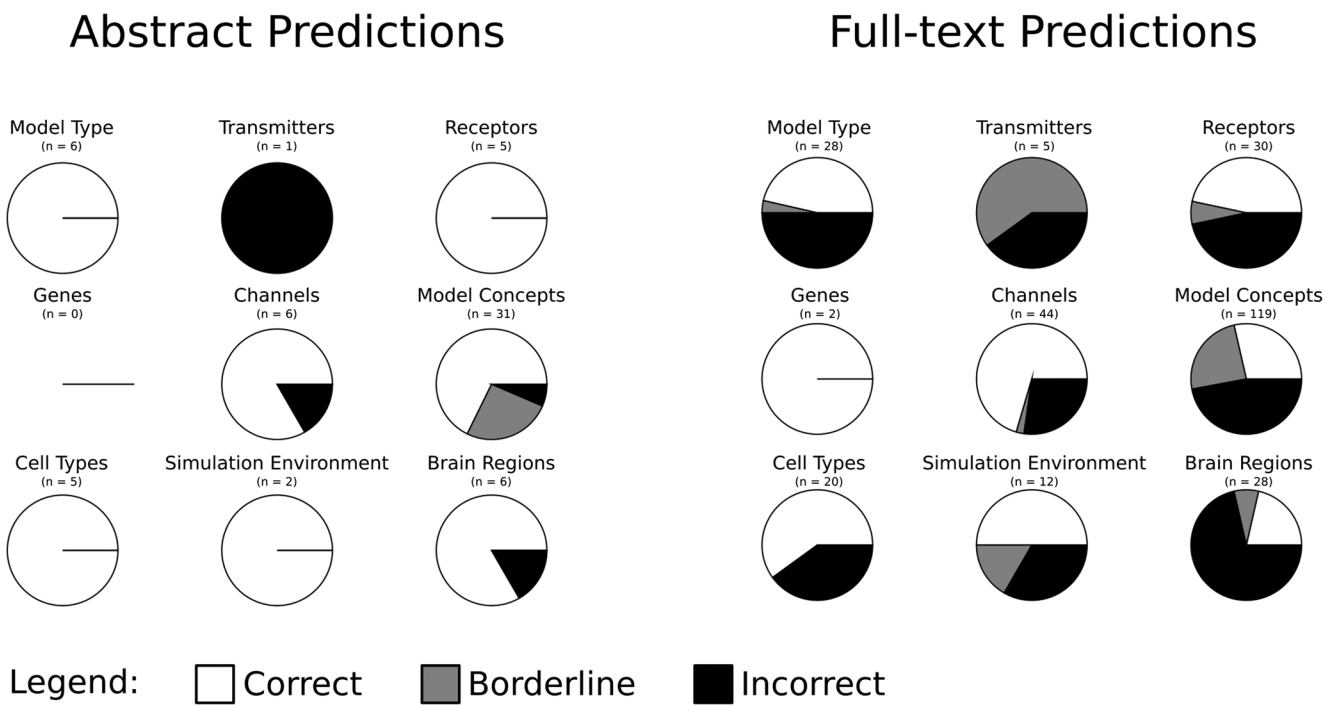

Fig. 4 Comparison of the prediction success rate for metadata predicted based on abstract vs full-text for ten papers for nine different categories of metadata. Prediction precision based on paper title is not shown because all such predictions were correct, however only four categories of metadata had any predicted metadata tags: receptors (2 predictions), model concepts (8), cell types (2), and brain regions (1)

On average, the metadata predictor predicted 6.41 ± 3.60 metadata tags per abstract, with 3.63 ± 2.53 of those not previously identified by manual curation. Twenty-five abstracts had 10 or more new metadata tags predicted. The distribution of numbers of novel predicted metadata tags is shown in Fig. 5. The predictor identified at least one and as many as 18 metadata tags not previously manually assigned for more than 90% of abstracts for ModelDB entries.

To assess the precision (fraction of suggestions that are accurate), recall (percent of all valid metadata tags that the suggestor suggests), and F-measure (the harmonic mean of the above), one of the authors who was not involved with the generation of the rule set (TMM, who has curated ModelDB for the majority of its existence) examined the metadata assigned to 39 models. Of these 39 models, 17 had no associated submitter-supplied metadata of the types considered here, illustrating the need for a way of promoting metadata entry. For this set of papers, on average our tool suggested 3.72 ± 2.24 metadata tags as compared to 5.13 ± 8.49 user-provided metadata annotations. Precision was assessed based on the expert curators assessment of the relevance of the tag, and recall was assessed against the final set of ModelDB metadata which included author provided metadata, correct predictor suggested, and curator added metadata. Given these definitions, our tool showed precision of 68.9 ± 27.6% and recall of 40.3 ± 19.9%, giving an F-measure of 0.480.

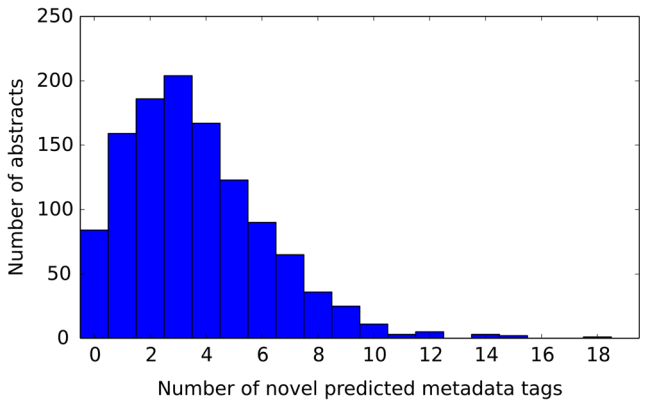

Fig. 5 Distribution of numbers of metadata tags not already found in ModelDB predicted by running the metadata predictor tool on all the ModelDB abstracts

The predictor algorithm emphasizes different types of metadata than manual curation has historically identified, enriching the distribution of metadata tags. In particular, model concepts comprise 43.5% of all predicted metadata, but only 27.1% of manually curated metadata. Simulation environment metadata (e.g. NEURON) comprises 11.2% of manually curated metadata and was associated with all but one model, but the algorithm only predicted a simulation environment for 27 of the 1164 abstracts, comprising 1.2% of all predicted metadata tags. A comparison of the distributions of metadata by category is shown in Fig. 6.

We found negligible correlation ($r = 0.2$) between the average number of times that a tag is predicted per model and the

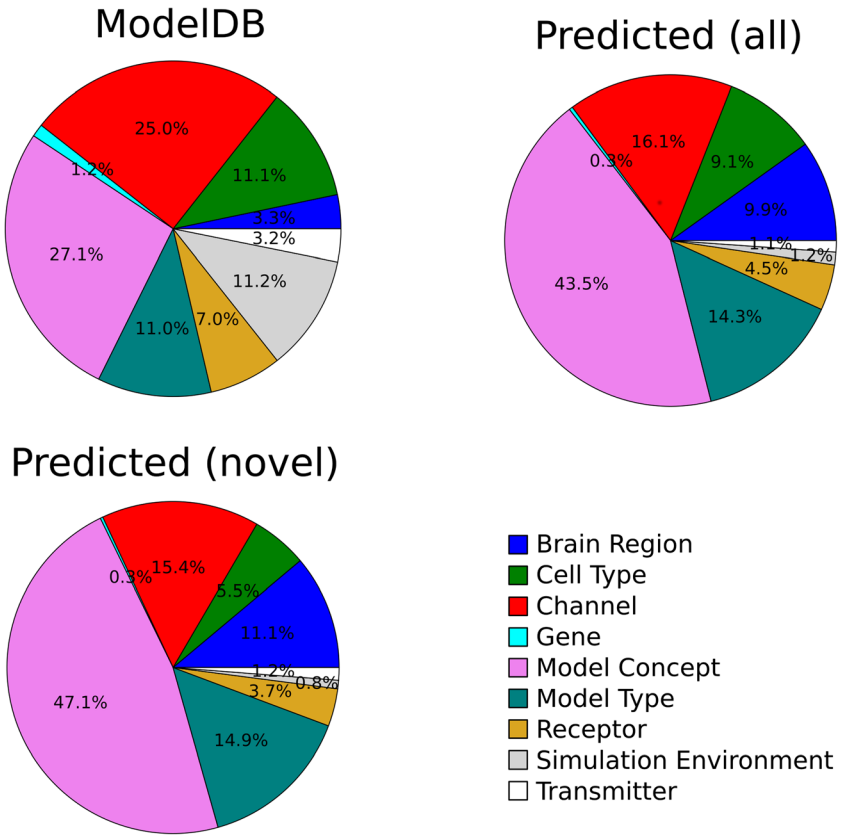

Fig. 6 Comparison of the distribution of metadata categories between the contents of ModelDB, all predicted metadata tags, and novel metadata tags. The categories start at 0 radius and are arranged alphabetically in the counterclockwise direction. Metadata is counted per paper abstract not per model entry

frequency with which those predictions coincide with existing ModelDB metadata entries on those models. There was a weak correlation ($r = 0.5$) between the frequency a tag is predicted vs the frequency it appears in ModelDB. Thus while metadata tags frequently used in manual curation tend to coincide with frequently predicted tags from the algorithm, the number of predictive rules triggered for a particular metadata tag in a given model is not indicative of whether or not the model will have been manually tagged with that piece of metadata.

Discussion

Richly annotating shared data with metadata is a core principle of FAIR data sharing (Wilkinson et al. 2016). This is a challenging task for repositories like ModelDB, as the submitters who created the data and are therefore the best able to describe it are typically unfamiliar with the details of the metadata ontology. The burden of annotation has thus fallen heavily on curation via manual or automatic (e.g. Van Auken et al. 2009) means, where the creator's expertise is no longer available. We describe a novel approach, bringing the data creator into the metadata annotation process by providing them with suggestions specific to their data that can be used directly and provide prompting to enter additional, related metadata. This approach is general and can be adapted by other informatics repositories.

Our approach prompts the data submitter to provide text related to the data they are providing as part of the submission process. For ModelDB, we suggest they provide the text of the abstract of a publication using the model. There is an unavoidable mismatch as the description of the published research as a whole is not necessarily a description of the model itself; nonetheless, publications are typically the only sources for information about a modeling work. A rule-based system then suggests arbitrarily many of 598 (in our case) potential metadata tags in a fraction of a second. Submitters then accept or reject the metadata annotations. Accepted annotations are listed in categorized metadata lists (e.g. there is a list for all cell types and a list for all genes), prompting the submitter to further annotate the model with additional metadata.

Submitters are, of course, free to provide any text they wish. Biological topics (e.g. Alzheimer's; called "model concepts" in ModelDB) are readily predicted from abstracts. Simulator environments (e.g. NEURON, MCell, etc) are typically not mentioned in abstracts and are therefore usually not predictable from abstracts. Submitters may list extra terms explicitly in the text field and exploit the rule set's ability to map terms to the repository ontology. Alternatively, they may provide larger sections of their manuscript. Moving from title to abstract to full text, we found using more text improved

recall (i.e. more correct metadata tags were suggested) with the tradeoff that it decreased precision (i.e. a smaller percentage of suggestions were correct). Preliminary tests suggest that providing some other sections (e.g. methods) can also provide metadata suggestions with a low false positive-rate. Extracting different categories of metadata from different portions of the text shows potential for increasing recall while preserving precision. By leaving the text used up to the submitter, they are free to chose their tolerance for false-positive suggestions. In any case, false-positive suggestions need not impair the quality of metadata annotations as the submitter is free to simply not accept those suggestions.

Text-mining has long been recognized as a useful tool for extracting information from the neuroscience literature. Crasto et al. (2003) used semi-automatic text-mining to extract experimentally-derived information about ion channel distribution to populate NeuronDB (senselab.med.yale.edu/neurondb; Mirsky et al. 1998). The WhiteText project (reviewed in French et al. 2015) develops automated techniques to identify brain regions in abstracts to extract information about connectivity between the regions. Ambert and Cohen (2012) and Richardet et al. (2015) identified the same type of information through machine learning techniques, and Tirupattur et al. (2011) used abstracts to build association graphs between neuroscience terms. These previous projects focused on identifying isolated facts within an article of a specific type (e.g. that two brain regions are connected); our current work, by contrast, suggests a wide variety of types of metadata describing many facets of a model.

To support suggesting from such a wide variety of potential metadata based on a real-world repository where existing metadata annotations are incomplete, not all metadata tags have formal definitions (for example, as of September 2, 2017, 35 out of 147 ModelDB "model concepts" were undefined), and many are used on only a few entries (Parkinson's is one of the most commonly modeled diseases in ModelDB, but even that has only 40 models) we applied rule based models. (See Cohen and Hunter 2008 for a brief discussion of text mining via rule-based vs machine learning approaches.) We achieved acceptable precision with only 1216 regular-expression based rules for 598 metadata terms, or 2.03 rules per term. Although we employed some optimizations to accelerate generation of the rule set, the process fundamentally consisted of reviewing the abstracts for all existing ModelDB entries to identify text and variants of text that supported the existing metadata annotations for the corresponding model. By using human-generated rules, we can exploit human expertise to examine all the abstracts without overfitting. No advanced expertise in text-mining is required, making the process easily adoptable by other repositories, although we did find it important to have a second person review the rules.

Quantifying the success of metadata suggestion in this context presents a couple of specific challenges. The primary measures, like for any text-mining project, are precision (fraction of suggestions that are correct) and recall (completeness). Complicating precision: not everything in a paper abstract describes the associated model, so finding a term that maps to an ontology term does not mean that it is an appropriate annotation for the model. Determining the border between what is part of the model and part of the context is sometimes subjective. Complicating recall: comparing to a human curated list of metadata annotations is limited by the curator's ability to identify which of 598 terms are relevant. Furthermore, not every relevant fact about a model is described in a corresponding paper's abstract, making such annotations impossible regardless of the algorithm.

In future work, more sophisticated strategies using more information could be combined synergistically with the rule set to improve metadata suggestion precision and recall. For example, the abstract for Beech and Barnes (1989), makes multiple references to the M-current, but only to describe similarities between that and a novel potassium current they identified and modeled. The current algorithm incorrectly predicts the model is about the M-current, but an algorithm that used sentence parsing could in principle recognize that the M-current was only mentioned as a comparison. We speculate that identifying context in this manner could improve the utility of a full text search by eliminating false predictions from terms mentioned only for comparisons to other work. Full text search could also be improved by gathering different categories of metadata from different sections of the text. For example, Fig. 4 suggests that analyzing the full text produces about 7x more predicted channels than analyzing the abstract with a true positive rate of around 75%; by contrast most correctly identified brain regions were predicted from the abstract text alone, and expanding beyond the abstract mostly only added false positives. Different rules could be employed for different sections of the text; e.g. multiple pieces of evidence might be required to suggest metadata tags that are not predicted from the abstract alone to eliminate false positives from comparisons mentioned only once. Copyright laws may limit the ability of some authors to share text for analysis.

The text of the model code, especially comments and variable names, could be analyzed in addition to the abstract, and the code could be compared to that of other models on ModelDB; e.g. similar ion channel specification files may describe the same biological channel. The ability to predict metadata from source code is limited in practice by the need for modelers to substitute data from other organisms to replace missing data about the kinetics in their organisms of interest (De Schutter 2014). A paper's author list also provides information that could be used to refine predictions as many authors tend to focus on a few research areas; this is not done at present because a citation is not currently required until the submitter is ready to make their model public.

Automatically suggesting relevant metadata tags to the data submitter during data entry into a repository offers the potential to help neuroinformatics resources process the increasing amounts of data generated by individual labs and large neuroscience initiatives. By assisting with metadata identification, these tools can help ensure newly entered data is discoverable for use in future research.

Information Sharing Statement

The abstract analysis code is available under a modified BSD license on ModelDB at http://modeldb.yale.edu/195555.

Acknowledgments This study was funded by the NIH grant R01 DC009977. We thank N Ted Carnevale for valuable feedback on the manuscript.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

Ambert, K. H., & Cohen, A. M. (2012). Text-mining and neuroscience. *International Review of Neurobiology, 103*, 109–132.

Anderson, J. C., Binzegger, T., Kahana, O., Martin, K. A. C., & Segev, I. (1999). Dendritic asymmetry cannot account for directional responses of neurons in visual cortex. *Nature Neuroscience, 2*(9), 820–824.

Ascoli, G. A. (2015). Sharing neuron data: carrots, sticks, and digital records. *PLoS Biology, 13*(10), e1002275.

Beech, D. J., & Barnes, S. (1989). Characterization of a voltage-gated K+ channel that accelerates the rod response to dim light. *Neuron, 3*, 573–581.

Cohen, K. B., & Hunter, L. (2008). Getting started in text mining. *PLoS Computational Biology, 4*(1), e20.

Cornelisse, L. N., van Elburg, R. A. J., Meredith, R. M., Yuste, R., & Mansvelder, H. D. (2007). High speed two-photon imaging of calcium dynamics in dendritic spines: consequences for spine calcium kinetics and buffer capacity. *PLoS One, 2*(10), e1073.

Crasto, C. J., Marenco, L. N., Migliore, M., Mao, B., Nadkarni, P. M., Miller, P., & Shepherd, G. M. (2003). Text mining neuroscience journal articles to populate neuroscience databases. *Neuroinformatics, 1*(3), 215–237.

Crockford, D. (2006). The application/json media type for JavaScript object notation (JSON). Available: https://www.ietf.org/rfc/rfc4627.txt. Accessed 1 Aug 2018.

De Schutter, E. (2014). The dangers of plug-and-play simulation using shared models. *Neuroinformatics, 12*(2), 227.

French, L., Liu, P., Marais, O., Koreman, T., Tseng, L., Lai, A., & Pavlidis, P. (2015). Text mining for neuroanatomy using WhiteText with an updated corpus and a new web application. *Frontiers in Neuroinformatics, 9*, 13.

Garcia-Grajales, J. A., Rucabado, G., Garcia-Dopico, A., Pena, J. M., & Jerusalem, A. (2015). Neurite, a finite difference large scale parallel program for the simulation of electrical signal propagation in neurites under mechanical loading. *PLoS One, 10*(2), e0116532.

Heinz, M. G., Zhang, X., Bruce, I. C., & Carney, L. H. (2001). Auditory nerve model for predicting performance limits of normal and impaired listeners. *Acoustics Research Letters Online, 2*(3), 91–96.

Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., St Pierre, S., & Twigger, S. (2008). Big data: the future of biocuration. *Nature, 455*(7209), 47–50.

Kim, M., Park, A. J., Havekes, R., Chay, A., Guercio, L. A., Oliveira, R. F., Abel, T., & Blackwell, K. T. (2011). Colocalization of protein kinase A with adenylyl cyclase enhances protein kinase A activity during induction of long-lasting long-term-potentiation. *PLoS Computational Biology, 7*, e1002084.

McDougal, R. A., Morse, T. M., Carnevale, T., Marenco, L., Wang, R., Migliore, M., Miller, P. L., Shepherd, G. M., & Hines, M. L. (2017). Twenty years of ModelDB and beyond: building essential modeling tools for the future of neuroscience. *Journal of Computational Neuroscience, 42*(1), 1–10.

Mirsky, J. S., Nadkarni, P. M., Healy, M. D., Miller, P. L., & Shepherd, G. M. (1998). Database tools for integrating and searching membrane property data correlated with neuronal morphology. *Journal of Neuroscience Methods, 82*, 105–121.

Morse, T., Carnevale, N. T., Mutalik, P., Migliore, M., & Shepherd, G.M. (2010). Abnormal excitability of oblique dendrites implicated in early Alzheimer's: a computational study. *Frontiers in Neural Circuits, 4*, 16. https://doi.org/10.3389/fncir.2010.00016.

Neymotin, S. A., Lee, H., Park, E., Fenton, A. A., & Lytton, W. W. (2011). Emergence of physiological oscillation frequencies in a computer model of neocortex. *Frontiers in Computational Neuroscience, 5*, 19–75.

Nielsen, J. (1993). *Usability Engineering*. Academic Press: Boston, MA.

Prescott, S. A., Ratte, S., De Koninck, Y., & Sejnowski, T. J. (2008). Pyramidal neurons switch from integrators in vitro to resonators under in vivo-like conditions. *Journal of Neurophysiology, 100*(6), 3030–3042.

Richardet, R., Chappelier, J. C., Telefont, M., & Hill, S. (2015). Large-scale extraction of brain connectivity from the neuroscientific literature. *Bioinformatics, 31*(10), 1640–1647.

Rishikesh, N., & Venkatesh, Y. V. (2003). A computational model for the development of simple-cell receptive fields spanning the regimes before and after eye-opening. *Neurocomputing, 50*, 125–158.

Sousa, M., Szucs, P., Lima, D., & Aguiar, P. (2014). The pronociceptive dorsal reticular nucleus contains mostly tonic neurons and shows a high prevalence of spontaneous activity in block preparation. *Journal of Neurophysiology, 111*(7), 1507–1518.

Tirupattur, N., Lapish, C.C., & Mukhopadhyay, S. (2011). Text mining for neuroscience. In *American Institute of Physics Conference Series 1371*, 118–127. https://doi.org/10.1063/1.3596634.

Van Auken, K., Jaffery, J., Chan, J., Müller, H. M., & Sternberg, P. W. (2009). Semi-automated curation of protein subcellular localization: a text mining-based approach to gene ontology (GO) cellular component curation. *BMC Bioinformatics, 10*(1), 228.

Wallach, H. M. (2006). Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd international conference on machine learning*, 977–984. ACM: Chicago. https://doi.org/10.1145/1143844.1143967.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., & Bouwman, J., (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data, 3*.

Wolf, J. A., Moyer, J. T., Lazarewicz, M. T., Contreras, D., Benoit-Marand, M., O'Donnell, P., & Finkel, L. H. (2005). NMDA-AMPA ratio impacts state transitions and entrainment to oscillations in a computational model of the nucleus accumbens medium spiny projection neuron. *The Journal of Neuroscience, 25*, 9080–9095.