Methods 74 (2015) 54–64

Contents lists available at ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Text as data: Using text-based features for proteins representation and for computational prediction of their characteristics

Hagit Shatkay [a,b,e,*], Scott Brady [c,e], Andrew Wong [d,e]

[a] Dept. of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA
[b] Delaware Biotechnology Institute, University of Delaware, Newark, DE 19711, USA
[c] School of Medicine, University of Toronto, Toronto, ON M5S 1A8, Canada
[d] Office of Personalized Genomics & Innovative Medicine, Mount Sinai Hospital, Toronto, ON M5G 1X5, Canada
[e] Computational Biology and Machine Learning Lab, School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada

ARTICLE INFO

Article history:
Received 12 April 2014
Received in revised form 21 September 2014
Accepted 21 October 2014
Available online 15 November 2014

Keywords:
Biomedical text mining
Machine learning
Text classification
Protein subcellular location
Protein function prediction
Protein annotation
Text mining
Protein representation
Protein location prediction

ABSTRACT

The current era of large-scale biology is characterized by a fast-paced growth in the number of sequenced genomes and, consequently, by a multitude of identified proteins whose function has yet to be determined. Simultaneously, any known or postulated information concerning genes and proteins is part of the ever-growing published scientific literature, which is expanding at a rate of over a million new publications per year. Computational tools that attempt to automatically predict and annotate protein characteristics, such as function and localization patterns, are being developed along with systems that aim to support the process via text mining. Most work on protein characterization focuses on features derived directly from protein sequence data. Protein-related work that does aim to utilize the literature typically concentrates on extracting specific facts (e.g., protein interactions) from text. In the past few years we have taken a different route, treating the literature as a source of text-based features, which can be employed just as sequence-based protein-features were used in earlier work, for predicting protein subcellular location and possibly also function. We discuss here in detail the overall approach, along with results from work we have done in this area demonstrating the value of this method and its potential use.

1. Introduction

The era of large-scale genome-based biology has been marked by an unprecedented number of sequenced genes and proteins, accompanied by a tremendous growth in the number of biomedical publications. High-throughput sequencing technology provides fast and relatively easy means to obtain the sequence information for a multitude of proteins. Naturally, traditional experimental methods for studying these proteins lag behind, resulting in a rapid increase in the number of proteins whose sequence is available but whose role within biological processes remains unknown. Much research is thus dedicated to characterizing proteins, identifying their structure, function, location and interactions, as well as to making such information available through public databases such as SwissProt and UniprotKB [59] or the Protein Data Bank [42].

As a lot of the information pertaining to genes and proteins is (and has been) published throughout the scientific literature, there is a surge of interest in biomedical text mining methods [52], aiming to accelerate the acquisition and the structuring of information obtained from unstructured text. Simultaneously, computational methods for predicting and deducing protein function, structure and location are also being developed. Here we discuss work that is in the intersection of these two directions, namely, the utilization of text as a component within computational methods for predicting protein subcellular location and function.

Computational methods for predicting proteins' characteristics typically utilize features derived from protein sequence, possibly along with structure or interaction networks [5,20,46]. For instance, the function-prediction systems GOtcha [30], OntoBLAST [63], and BLAST2GO [12] rely on sequence similarity, PHUNCTIONER [39] and ConFunc [60] use similarity between protein structures, while GeneMANIA [33] and an earlier system by Chua et al. [10] rely on protein-interaction networks. Similarly, quite a few location prediction systems use sequence motifs, sequence similarity, or more refined sequence-based features to predict the

* Corresponding author at: Dept. of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA.
E-mail address: shatkay@udel.edu (H. Shatkay).

http://dx.doi.org/10.1016/j.ymeth.2014.10.027
1046-2023/

subcellular location of proteins [e.g., [3,9,18,23,24,34,55]]. Notably, computational prediction of a protein's function or location, as discussed here, is often framed as a classification task. The class-labels are the possible functions or the organelles within the cell, and the goal is to take a protein – typically represented as a feature vector based on sequence properties – and assign it with a correct class-label.

An alternative approach to the sequence-based representation of proteins is text-based representation. The underlying idea is that if a passage of text is relevant to a protein, there is often information therein that can be used to help deduce a protein's class (i.e., its subcellular location or its bio-molecular role). In the context of protein characterization, text can be put to use through two distinct approaches: *Information Extraction* and *Text-based classification*. While we focus on the latter (i.e., classification), we briefly discuss the former. *Information extraction* systems aim to identify and extract phrases or terms within the text that explicitly describe the protein's characteristics. That is, rather than *predict* yet unknown information, extraction systems aim to *find out what has already been discovered and reported* in the literature about the protein in terms of function, process or location. AbXtract [1], which was one of the earliest extraction systems in the biomedical domain, aimed to identify and rank sentences discussing protein function based on statistical properties of words in the sentence. Craven and Kumlien [14] have used a hidden Markov model of sentence structure to *extract* protein subcellular location from documents discussing it. Several later systems have used extraction strategies to identify text passages discussing protein function. For instance, Pérez et al. [40] introduced a dictionary-based system that extracts keywords from the literature or from databases and associates them with GO categories; Other systems used pattern matching and sentence structure to retrieve sentences containing a protein along with Gene Ontology (GO) terms denoting function [8,26]. A recent function prediction system [56] identifies pairs of GO terms and proteins within abstracts, and uses them as part of an integrative similarity measure (kernel) employed in classifying proteins by function. Additional information extraction systems have been used in a variety of knowledge discovery tasks within the biomedical domain (see surveys, e.g., [11,25,52]).

In contrast to textual information extraction systems, *classification systems* represent genes and proteins using features that are derived from text sources – regardless of whether the text explicitly discusses the proteins' function/location. The idea underlying this approach, which is rooted in probabilistic information retrieval and language models [47–49], is that the language or, more explicitly, *the distribution of words* used within the text to discuss the protein (or the gene) can provide cues about its function, process or location. We can thus make use of sets of proteins whose characterization is already known, represent them based on text-features, and train machine-learning classifiers that can then assign class-labels to yet-unannotated proteins (where the latter are also represented using text-based features).

For instance, in an early work Raychaudhuri et al. [45] classified published abstracts into *biological process* GO categories (using 21 categories). Proteins that are mentioned in each abstract are then assigned the GO categories associated with the abstract. In our own early work on using text for characterizing gene's function, we have introduced the use of probabilistic topic models applied to PubMed abstracts for representing sets of genes sharing a common function [53]. Van Driel et al. [16] later use a similar idea for grouping and characterizing genes, by identifying similarities among the text describing their respective phenotypes, obtained from OMIM; Groth et al. [21,22] also approach phenotype-based study of genes by applying a clustering technique to the text-descriptions of phenotypes, and associating text and keywords within it with GO categories. A text-based classification system by Stapley et al. [57] used support vector machines to assign yeast proteins to subcellular locations; Nenadic et al. [36] used a similar approach to annotate proteins with one of *11 biological process* terms from the upper levels of the GO hierarchy. In both cases, proteins were represented as vectors of words occurring in abstracts that mentioned the protein's name. More recent work in the area of text-assisted functional annotation [37,58] examined the classification of biomedical abstracts (rather than of proteins) into functional categories, tagging the abstracts themselves with relevant GO codes.

Another source of text considered for use in automated characterization of proteins consists of the descriptive terminology (typically GO terms) appearing within protein annotations in public databases, such as SwissProt/UniProtKB. Eisenhaber and Bork's rule-based Meta_A(nnotator) [17] used functional annotation terms from the protein's SwissProt entry to deduce the protein's location. Nair and Rost [35] used text from the same source to associate proteins with selected functional keywords and develop the LOCkey classifier for predicting subcellular location. Utilizing only such functional keywords for protein representation greatly limited the coverage of the system to proteins already annotated with these keywords. Eskin and Agichtein [19] expanded on LOCkey by utilizing as part of the classification scheme more of the annotation terms associated with the proteins, as well as protein sequence features, albeit without demonstrating improved performance. More recent systems for protein subcellular location prediction such as Proteome Analyst [28] and YLoc [3], while relying primarily on sequence-based features for representing proteins, also employ text-features obtained from protein annotation (e.g., GO terms annotating the proteins) to aid in the prediction. Notably, having a prediction system use such features for protein-representation implies that the protein in question has already been manually curated and annotated, which limits the utility of the system to aid in de novo annotation of proteins that have not yet been characterized.

The methods we discuss throughout the rest of this paper aim to take advantage of the available published text for protein representation and classification, without relying on manually-curated annotation terms (such as GO terms assigned to the protein). We thus focus on text obtained from the published literature, specifically from PubMed abstracts [43], that can be associated with proteins and utilized by automated systems. These ideas have been put to use in specific systems we have developed to address the two tasks discussed before, namely protein subcellular location prediction [2,4] and protein function prediction [61]. Here we present a complete framework for using text as a basis for representing and characterizing proteins; moreover, the *function prediction* work and results discussed here employ *support vector machine classifiers (SVM)*, as opposed to *k*-nearest neighbor classifiers that were used before (the latter were reported in [61]). The approach and the methods, the results of applying them – and the lessons learned from these applications, are presented and discussed in detail throughout the following sections.

2. Methods: from proteins to text and back

To use text as a form of data for characterizing proteins, one must first identify a source of text pertaining to proteins, along with a strategy for associating each protein with its related text. Next, one needs to represent proteins as feature vectors based on the associated text, possibly making use of additional aspects of the protein (such as sequence-based information) in the representation. Once proteins are represented as feature vectors, machine-learning methods for training and testing classifiers can be applied and used for protein characterization. In this section we focus primarily on the first two steps, namely association of proteins

with text, and text-based protein representation. The classification task and the classifiers themselves are discussed at the end of this section.

2.1. Associating text with proteins

Theoretically speaking, any database discussing proteins (e.g., organism-specific resources) may contain text or documents pertaining to a subset of proteins. However, when developing a prediction system, it is essential that the resource selected as a text-source provides text association for a substantial number proteins, thereby allowing for the majority of the proteins to be represented. As such, we focus on two major resources: UniProtKB/SwissProt [59], which is a comprehensive database containing sequence data and available information about hundreds of thousands of proteins, and PubMed [43], which is a comprehensive online database containing the abstracts from more than twenty million published biomedical articles.

Given a set of proteins, we first identify the UniProtKB entries associated with them, either through their protein accession number or, if such identifiers are not provided, by sequence identity. To obtain abstracts pertaining to each protein, we extract the PubMed identifiers (PMIDs) provided as references from within the protein's respective UniProtKB entry. We then retrieve the corresponding abstracts that are all publicly available from PubMed.[1] By retrieving abstracts that are curated in UniProtKB we ensure that the abstracts are indeed relevant for the proteins and are of high scientific quality. An alternative approach could have been to scan through PubMed for articles that mention the protein's name or one of its synonyms. However, given the well-known difficulty and complex issues in correctly identifying protein names in the literature [29], as well as in identifying truly relevant articles for a subject matter, we adopt the strategy of harvesting abstract references from UniProtKB.

Notably, we use *abstracts* rather than full-text articles because abstracts are readily and publically available, whereas full-text articles are often not freely accessible. The use of abstracts as a text source has proven beneficial for curation in the context of protein characterization [15], and in our experience for predicting protein location [2]. Moreover, Shah et al. [51] showed that the abstract has the highest density of keywords out of all other sections in an article, and is therefore typically a good source of text features. We also point out that not all the abstracts obtained using the above procedure are strongly associated with any one protein or with any one biological function, location or other characteristics. For instance, an abstract may discuss a multitude of proteins and multiple functions. Such abstracts are typically flagged and discarded by our methods (as described in Section 3), because the terms occurring in them are usually not strongly indicative of any one characteristic (location, function or process) of a protein.

2.2. Selecting informative terms

Once the collection of published abstracts relevant to the set of proteins has been gathered, terms must be selected from the text as a basis for protein representation. This process, known as *feature selection*, is commonly used in a variety of text classification tasks. The goal is to select only those terms that are useful for distinguishing between items from different classes. In our case, the items are *proteins*, and the classes can be the different *subcellular locations*, or possibly the different *biological functions* or *processes* in which a protein participates. Feature selection reduces the computational cost of machine learning algorithms, and often improves classification accuracy [50,62]. Several feature selection methods are used in practice and comparative studies among them have been conducted [2,62]. Here we describe a method we have used in several contexts and has proven useful, which aims to select terms that are associated with a class with high statistical significance, as further described below. The selected terms are then used as text features to represent proteins, forming the basis for our classifiers.

To obtain the terms, we pre-process all the abstracts, extracting all individual words (unigrams) as well as pairs of consecutive words (bigrams). All words are stemmed through Porter stemming [41], which removes suffixes and retains the root form of words. The number of terms is further reduced by eliminating stop words such as pronouns, determiners and prepositions (e.g., '*the*', '*that*', '*or*') and removing common words occurring in more than 70% of the abstracts. We also remove rare and specific words, appearing in fewer than three abstracts. This process results in a set of *candidate terms*. We then apply the *Z-score* statistical test to identify *characteristic terms*, as described next.

We say that a term t is *characteristic* with respect to a class c, if t's probability to appear in abstracts associated with proteins belonging to class c, is statistically-significantly different from its probability to appear in abstracts associated with proteins of *all other* classes. The *Z-score* is calculated for each term t, indicating the statistical significance of the difference in t's occurrence-probability between classes. For a term t, and two different classes c and c', the *Z-score* is defined as:

$$Z^{t}_{c,c'} = \frac{\Pr(t|c) - \Pr(t|c')}{\sqrt{\dot{\Pr}\cdot(1-\dot{\Pr})\cdot\left(\frac{1}{|D_c|}+\frac{1}{|D_{c'}|}\right)}}, \text{ where } \dot{\Pr} = \frac{|D_c|\cdot\Pr(t|c) + |D_{c'}|\cdot\Pr(t|c')}{|D_c|+|D_{c'}|},$$

D_c denotes the set of abstracts associated with proteins whose class is c, and $\Pr(t|c)$ is the conditional probability of t to appear in abstracts that are associated with proteins whose class is c. For a class c, the latter conditional probability, $\Pr(t|c)$, is estimated through a maximum likelihood estimate derived from a training set of proteins whose classes (e.g., locations or functions) are already annotated. It is calculated by dividing the number of abstracts that are associated with proteins whose class is known to be c and *contain the term* t, by the total number of abstracts associated with proteins whose class is known to be c. Formally: $\Pr(t|c) \approx \frac{|d \in D_c \quad s.t. \quad t \in d|}{|D_c|}$, where d denotes an abstract. For a term t and a class c, if the absolute value of the *Z-score* is higher than a pre-set threshold with respect to *each of the other classes* c', t is considered to be *characteristic* for class c.

Fig. 1 illustrates the process of selecting characteristic terms in the context of protein subcellular localization, where classes correspond to organelles. Two classes are specifically shown, the *Nucleus (nuc)* and the *Endoplasmic reticulum (ER)*.

Table 1 shows examples of top characteristic terms associated with some of the molecular function categories, obtained by applying the method discussed above to the collection of abstracts discussed in Section 3.2. It also shows some of the characteristic terms obtained through the same process in the context of subcellular locations or organelles (Section 3.1 and [2]).

In each of the systems described in the following sections, the union of *all the characteristic terms* over all the classes considered by the system is used as the set of text features with which we represent proteins. We denote the resulting set of N characteristic terms by T_N.

It is worth noting that characteristic terms most strongly associated with a function or with a subcellular location, while indicative of the function/location and provide cues to what it might be, do not typically explicitly include its name. Table 1 illustrates the

[1] Proteins that have no UniProt entry, or no PMIDs listed in their entry, are referred to as textless and handled as discussed later.

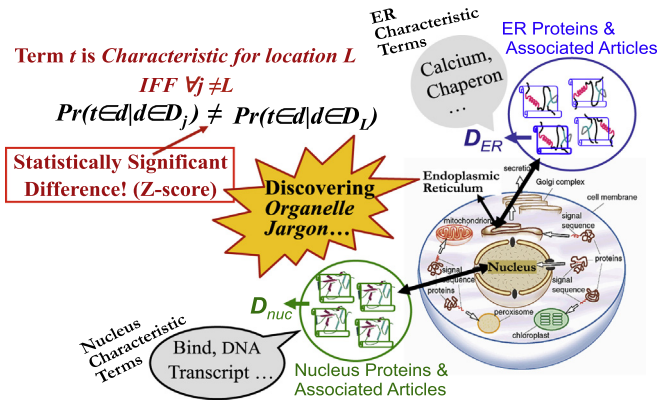

Fig. 1. Obtaining characteristic terms for proteins in eukaryotic cell organelles. The cell is shown on the bottom right (cell image [6]). Two organelles in particular are indicated: The *Nucleus* and the *Endoplasmic Reticulum (ER)*. Proteins already known to be associated with these organelles are obtained from UniProkB, along with references to their curated abstracts. The respective sets of abstracts are denoted D_{nuc} and D_{ER}. Terms are identified within the abstracts, and characteristic terms, whose probability to occur in documents associated with a particular organelle is significantly different from its probability to occur in the context of other organelles, are selected.

Table 1
Examples of stemmed characteristic terms associated with several molecular functions (top) and with several eukaryotic subcellular locations (bottom).

Molecular function	Example function-characteristic terms
Catalytic activity	Enzyme, oxidas, receptor, reductas, repair, require
Transporter activity	Anion, carrier, cation, channel, gate, potassium, uptake, voltage
Structural molecule activity	Actin, collagen, cytoskeleton, filament, matrix, muscle, myosin
Enzyme regulator activity	Cyclin depend, exchange factor, GTP, GTPase, inhibitor, kinase
Location	**Example location-characteristic terms**
Nucleus	Bind, base pair, chromatin, DNA
Mitochondria	Acyl coa, cytochrom, electron transport
Golgi apparatus	Acceptor, galactos, golgi, transferase
Endoplastic reticulum	Chaperon, disulfid isomeras, endoplasm

point. This observation supports the idea that we are *not performing information extraction* of location or function from the text, but rather executing an actually *predictive* task, of deducing location (or function) from the terms mentioned in the text. As a demonstration, in an earlier publication we have shown an example of location assignment to a protein based on a text-description that was completely sanitized of any location terms [2]. In a more extensive earlier study [54], we have tested and demonstrated the predictive ability of our approach by assigning location to proteins using only the text available from articles that pre-dated the time in which the proteins' location became known. (See [2,54] for further detail.)

2.3. Representing proteins as feature vectors

To represent each protein we use the well-known '*bag of words*' approach [32]. Each protein p is represented as an N-dimensional vector of term weights $\langle w^p_{t_1}, w^p_{t_i}, \ldots, w^p_{t_N} \rangle$, where $t_1, \ldots, t_N$ are the characteristic terms in the selected set, T_N, and $w^p_{t_i}$ is a weight reflecting the significance of term t_i within the set of abstracts D_p that is associated with protein p. The weight is calculated as the ratio between the number of occurrences of t_i within D_p and the total number of term occurrences of *all* the characteristic terms, t_j, from the set T_N in D_p:

$$w^p_{t_i} = \frac{\#\text{ of times } t_i \text{ appears in } D_p}{\sum_{t_j \in T_N} (\#\text{ of times } t_j \text{ appears in } D_p)}.$$

While proteins that are part of our training set are all associated with text, and can thus be represented as explained above, there may be query proteins that cannot be associated with text for either one of the following reasons:

- The protein does not have an entry in the UniProtKB/SwissProt database;
- There are no related articles recorded in its protein entry within the UniProtKB;
- The article(s) were removed from the text collection for reasons mentioned earlier (e.g., not being descriptive of any specific class of proteins).

We refer to such proteins as *textless*, and handle them by assigning the text features of homologous proteins to the *textless* protein, as briefly explained next. (See our earlier publications [2,4,61] for a more detailed description of the method, and [2] in particular for additional ways of handling textless proteins). The sequence of the textless protein is compared to sequences of proteins that do have associated abstracts using BLASTP; matches with *e-value* lower than 10 and *sequence identity* higher than 40%[2] are sorted by *e-value*, and the *three proteins with the lowest e-values* are selected as homologs. A weighted combination of the respective text-feature-vectors of the three homologs (denoted v_1, v_2, v_3) is designated as the textless protein's vector, $v_{textless}$. The weighting is based on the respective sequence identity, s_j, and is calculated as: $v_{textless} = (\sum_{j=1}^{3} s_j \cdot v_j / 3)$. We emphasize that this approach is very different from that of protein-annotation "*by similarity*" through re-use of the annotation of a protein's close homologs [10,34,60]. Notably, we use here the homologs just as a source of *text-based representation* for a textless protein; the representation is then used for independently categorizing the protein based on its text. This point is further discussed in Section 4.3.

2.4. The classifiers: training and testing

To train and test classifiers, we used datasets of proteins for which a reliable annotation of location, function, or process was already assigned according to UniProtKB. These datasets are further described in Section 3. As discussed earlier, for each protein we retrieved the PubMed abstracts referenced from its respective UniProtKB entry. Characteristic terms (as pertaining to each of the specific classification tasks addressed) were extracted and used as features for representing proteins. The resulting datasets of proteins were then used to train and test classifiers through multiple runs of *stratified 5-fold cross-validation*. Under this scheme, the dataset is partitioned at random into 5 disjoint subsets where in each subset the class instance distribution is the same as in the whole dataset. The classifier is trained and tested five times, where each time a different subset serves for testing while the other four subsets are used for training. To avoid biasing the results through any specific partition, we conduct five complete sets of cross-validation experiments, each experiment using a different 5-way partition, for a total of 25 runs.

The classifiers we employ here are all based on the LIBSVM [7] implementation of support vector machines (SVMs), all with a radial-basis-function kernel. LIBSVM enables soft, probabilistic categorization for n-class tasks through multiple *one-vs-one* runs; each classified item is assigned an n-dimensional vector denoting

[2] The 40% threshold is used due to a study by Brenner et al. suggesting that the sequence identity of a match should be greater than 40% for the matching sequences to be considered homologous.

the item's probability to belong to each of the n classes. For the classification of proteins according to location, we have also experimented with versions of Naïve Bayes classifiers, with similar results [27], but focus here on the system that uses SVM [2].

For the classification of proteins according to *function* or to *process* (using the respective GO categories) we have formerly used a k-nearest neighbor classifier (KNN, with $k = 10$) [13], as it is simple to implement and to modify. We have used that classifier in the CAFA (Critical Assessment of Function Annotation) challenge [44,61], and a detailed description of this classifier and its associated results are provided there [61]. Here we introduce new results (Section 3.2) obtained using an SVM classifier. For the sake of completeness and further insight into the results, we include in our tables results from both the SVM and the from the KNN classifier.

Viewing protein subcellular localization as a classification task is rather intuitive. Each of the *subcellular organelles* or substructures can be viewed as *a class*, and the proteins need to be assigned class tags reflecting the organelles to which they localize. Similarly, posing function prediction as classification implies that the GO categories comprising the *biological process* and *molecular function* sub-ontologies are the class labels. However, in contrast to protein location prediction that involves a relatively small number of organelles as classes (typically, at most 13), there are about *20,000* distinct *biological process* and *9000 molecular function* GO categories. The vast majority of these do not have a sufficient number of proteins (or associated abstracts) that can be used to train a classifier. For instance, the GO term '*dihydrofolic acid binding*' has only two associated proteins, while '*platelet activating factor metabolism*', has only one. Such dearth of data hinders both statistically significant feature selection and reliable training of classifiers.

We therefore experimented with several strategies for reducing the number of categories. One approach was to select only the 20 GO categories that have the highest number of associated proteins (regardless of their level in the GO hierarchy). However, these GO categories are relatively refined and deeply nested in the hierarchy; tracing to their parents up to the second-from-the-top level of the GO hierarchy, shows that out of the 17 high-level categories in the *molecular function* sub-ontology, only three (namely, '*binding*', '*structural molecular activity*', and '*electron carrier activity*') are represented. For the *biological process* sub-ontology, the selected categories represent only five out of the 29 categories at the second level of the GO hierarchy. Thus, we employ a different strategy that does not limit as much the diversity of protein functions our classifier can assign.

To ensure that diversity in high-level protein functions is better accounted for, we use as function classes all the GO categories at a specific relatively-high level in the GO hierarchy. Initially, we tried using the third level from the top, but found that a majority of the classes still did not have a sufficient number of associated proteins (fewer than 10 proteins for most of the categories). Therefore, we use as function classes only the *GO categories at the second level of the GO hierarchy* (one level away from the root node), where the GO sub-categories descending from each node are merged. There are 29 distinct *biological process* categories and 17 distinct *molecular function* categories at this level. However, 12 of these categories had fewer than 15 proteins each and were therefore removed (along with the total of 52 proteins that were associated with them) from our dataset. This leaves a final total of 10 *molecular function* categories and 24 *biological process* categories, as shown in Table 2.

3. Experimental setting and results

We discuss here two sets of experiments we have conducted. The *first,* presented in Section 3.1, is concerned with predicting the subcellular location of eukaryotic proteins into the main subcellular compartments of the eukaryotic cell (see [2,54] for additional details). The *second* set of experiments, discussed in Section 3.2, is concerned with the task presented by the CAFA challenge, namely predicting proteins' function and process, where both types of categories are as denoted within the GO hierarchy. While the classifier we have used in the CAFA challenge [61] was a KNN classifier, the classifiers discussed and presented here are all based on LIBSVM (as is our subcellular-location classifier).

3.1. Protein subcellular location prediction

3.1.1. Experimental setting

The text-based classifier for subcellular location prediction, to which we refer as *EpiLoc*, was trained and tested, as we have

Table 2
The 10 function categories and the 24 process categories that are used as classes.

Molecular function		Biological process	
GO ID	GO category	GO ID	GO category
GO:0005488	Binding	GO:0065007	Biological regulation
GO:0003824	Catalytic activity	GO:0032502	Developmental process
GO:0030528	Transcription regulator activity	GO:0009987	Cellular process
GO:0005215	Transporter activity	GO:0050896	Response to stimulus
GO:0060089	Molecular transducer activity	GO:0008152	Metabolic process
GO:0030234	Enzyme regulator activity	GO:0051234	Establishment of localization
GO:0005198	Structural molecular activity	GO:0016043	Cellular component organization
GO:0016247	Channel regulator activity	GO:0023052	Signaling
GO:0009055	Electron carrier activity	GO:0032501	Multi-cellular organismal process
GO:0045182	Translation regulator activity	GO:0022414	Reproductive process
		GO:0051704	Multi-organism process
		GO:0040011	Locomotion
		GO:0040007	Growth
		GO:0051179	Localization
		GO:0022610	Biological adhesion
		GO:0008283	Cell proliferation
		GO:0000003	Reproduction
		GO:0002376	Immune system process
		GO:0016265	Death
		GO:0071554	Cell wall organization or biogenesis
		GO:0048511	Rhythmic process
		GO:0023046	Signaling process
		GO:0044085	Cellular component biogenesis
		GO:0043473	Pigmentation

Table 3
Overall prediction performance of EpiLoc on the TargetP, PLOC and MultiLoc datasets, compared to that reported by the respective systems on the same dataset. For the MultiLoc dataset HomoLoc's performance is also shown. Performance is shown in terms of Average Sensitivity and Overall Accuracy. For TargetP, the non-Plant results are summarized only over the three organelles (*Mitochondria, Secretory Pathway* and *Other*), while the Plant results are shown summarized over the four organelles that also include Chloroplast. Highest performance values are indicated in boldface. Standard deviations are shown in parentheses (except for when unavailable in the original system paper).

Dataset	System	Avg. sensitivity	Overall accuracy
TargetP Plant	TargetP	0.856 (n/a)	0.853 (±0.035)
	EpiLoc	**0.883 (±0.001)**	**0.862 (±0.004)**
TargetP non-Plant	TargetP	0.907 (n/a)	0.900 (±0.007)
	EpiLoc	**0.908 (±0.003)**	**0.901 (±0.006)**
PLOC	PLOC	0.579 (± 0.021)	**0.796 (± 0.009)**
	EpiLoc	**0.773 (±0.0012)**	0.743 (±0.002)
MultiLoc	MultiLoc	0.741 (± 0.025)	0.746 (± 0.01)
	EpiLoc	0.818 (±0.005)	0.792 (±0.008)
	HomoLoc	**0.822 (± 0.005)**	**0.812 (± 0.010)**

described before [2], through extensive cross-validation studies. The latter were conducted on three different datasets, previously used by other location-prediction systems whose predictions were based on features derived directly from protein sequence, namely TargetP [18], PLOC [38], and MultiLoc [24], while using only the proteins in these sets that are associated with text (i.e., not textless). As the vast majority of the proteins do have associated text, the number of proteins is still substantial (as shown below), and enables a meaningful comparison. The performance of EpiLoc was then compared to that reported by these three systems. We provide below some more of the details to make this presentation of the topic self-contained.

The TargetP dataset [18] consists of *3123* eukaryotic proteins with associated text, known to be from four plant subcellular locations: *chloroplast* (*ch*, 123 proteins), *mitochondria* (*mi*, 465 proteins), *Secretory Pathway* organelles (*SP*, 921 proteins) and *Other* (*OT*, 1614 proteins), or three non-plant locations (*mi*, *SP*, and *OT*). The *Secretory Pathway* class includes proteins from all locations participating in this pathway, namely: *endoplasmic reticulum (er), extracellular space (ex), Golgi apparatus (go), lysosome (ly), plasma membrane (pm),* and *vacuole (va)*, while the *OT (Other)* class includes *cytoplasmic (cy)* and *nuclear (nu)* proteins. The *PLOC* dataset [38] contains *6503* eukaryotic proteins whose location was available in SwissProt Release 39.0. Proteins whose sequence identity with another protein in the set was greater than 80% were removed. In addition to the explicit locations listed above, the PLOC set also includes proteins from the *cytoskeleton* (*cs*), and the *peroxisome* (*pe*). The respective numbers of protein per location are: *ch (528 proteins), cs (37), cy (1113), er (101), ex (805), go (34), ly (85), mi (625), nu (1833), pm (1184), va (44),* and *pe (114)*. This set is the largest of the three, because it includes proteins whose subcellular location annotation in SwissProt indicated uncertainty through the words *potential*, *probable,* or *by similarity*. The *MultiLoc* set consists of *5345* eukaryotic proteins with associated text obtained from SwissProt release 42.0. Similar to the PLOC database, proteins whose sequence identity with another protein was greater than 80% were excluded. However, unlike PLOC, this set *excludes* proteins whose subcellular location annotation is uncertain, indicated by the terms *by similarity, potential,* or *probable* in the location annotation. The set of locations covered is the same as that covered by *PLOC* except for the *cytoskeleton* (*cs*). The distribution of proteins per location is given as part of Table 4.

To evaluate EpiLoc's performance we obtained the text associated with the proteins in each of the above datasets, identified a set of about 2000 characteristic terms as described in Section 2.2, and represented the proteins as weighted term vectors as described in Section 2.3. For each of the three datasets, we performed five complete studies of 5-fold cross-validation runs each, (25 cross-validation runs in total per dataset), where each of the five studies is done on a different five-way partition, thus ensuring the robustness of the results. Notably, the selection of characteristic terms for representing proteins is re-executed in each cross-validation run, *using only the training set for choosing informative terms*. This is critical in experiments involving feature selection, ensuring that the *feature selection itself utilizes only the training data* and does not rely on any of the test data.

To test the system's ability to handle textless proteins using the homology-based approach (see Section 2.3), we ran a set of cross-validation experiments over the MultiLoc dataset in which we removed the text associated with proteins in each of the test subsets. Each protein in the test set was then represented using the averaged text-based vector-representation of its homologs, without including the text associated with the protein itself. The system thus handling textless proteins is referred to throughout the rest of the paper as *HomoLoc* [2].

The evaluation metrics were the same ones as reported for the previous location prediction systems [18,24,38], namely sensitivity (*Sens*), specificity (*Spec*),[3] and Matthew's correlation coefficient (*MCC*) [31]. These are formally defined as:

$$Sens = \frac{TP}{TP+FN},\quad Spec = \frac{TP}{TP+FP},\quad \text{and}\quad MCC = \frac{TP\cdot TN - FP\cdot FN}{\sqrt{(TP+FN)\cdot(TP+FP)\cdot(TN+FN)\cdot(TN+FP)}},$$

where *TP, TN, FP,* and *FN* represent the number of true positives, true negatives, false positives, and false negatives, respectively, with respect to a given location, where a *positive* is the assignment of a protein to a location and a *negative* means *not assigning* the protein to the location.

The *sensitivity* measure as defined above is also referred to as *Recall,* while the specificity measure –when defined this way – is also referred to as *Precision*. In the context of protein subcellular localization, we use the terms *Sensitivity* and *Specificity* as defined above as opposed to the standard terms of *Recall* and *Precision*, for compatibility with the terminology used by many earlier systems. In the context of function prediction, as discussed in Section 3.2, we do use the *Recall* and *Precision* terminology. We also calculate for each dataset the *average sensitivity*, *Avg Sens*, over all locations, as well as the *overall accuracy*, $Acc = C/N$, where C is the total number of correctly classified proteins in the dataset and N is the total number of proteins in the dataset.

3.1.2. Results for protein subcellular prediction

Detailed results from our experiments of using text for protein subcellular location are provided in our earlier publications [54,2,4]. We therefore present here a brief summary of the results, as shown in Table 3. The table shows the average sensitivity (*Avg. Sens.*) and overall accuracy (*Overall Acc*) of the various systems over the respective datasets. For further illustration and for supporting the discussion in the context of this manuscript, Table 4 (adapted from [2]), shows location-specific results focusing on the *MultiLoc* dataset and highlighting the performance of the homology-based text classifier, *HomoLoc*. Further discussion of the results is provided in Section 4.

3.2. Protein function/process prediction

3.2.1. Experimental Setting

For our experiments in training and testing a text-based classifier for predicting proteins' function and process, we first

[3] Notably, Specificity is typically defined as $TN/(TN+FP)$; we use here the dual definition $TP/(TN+FP)$, because it was the one used in all preceding localization systems, and we wanted to retain a consistent notation with those in our comparison.

Table 4
(adapted from [2]). Prediction performance of MultiLoc (taken from [24]), the text-based classifier *EpiLoc*, and the version of the text-based classifier that assigns text to a protein based on the text associated with its homologs, *HomoLoc*. Results are shown on the Animal proteins of the MultiLoc dataset. Highest values are shown in boldface. The number of proteins associated with each location is shown in the second column on the left. Locations with fewer than 200 proteins, where the text classifiers show particular improvement over the sequence-based classifier are also shown in boldface.

Location	# of Proteins	MultiLoc dataset (Animal)		
		MultiLoc	EpiLoc	HomoLoc
		(Sens Spec MCC)		
go	**140**	0.71 0.43 0.53	0.88 0.62 0.73	**0.90 0.72 0.80**
ly	**98**	0.69 0.36 0.48	**0.86** 0.39 0.57	0.85 **0.49 0.63**
er	821	0.68 0.56 0.60	0.74 0.59 0.65	**0.77 0.67 0.71**
pe	**135**	0.71 0.31 0.44	0.90 **0.77 0.82**	0.80 0.69 0.74
mi	443	**0.88** 0.82 **0.83**	0.82 0.82 0.80	0.79 **0.84** 0.80
ex	821	0.79 **0.83** 0.77	0.80 0.82 0.77	**0.83 0.83 0.79**
cy	1290	0.67 **0.85** 0.68	0.68 0.79 0.65	**0.72** 0.80 **0.67**
pm	1173	0.73 0.90 0.76	0.85 0.90 0.84	**0.89 0.91 0.87**
nu	685	0.82 0.73 0.73	0.84 0.81 0.80	**0.87 0.84 0.83**
Overall Acc		0.746 (± 0.01)	0.792 (±0.008)	**0.812 (± 0.010)**
Avg. Sens.		0.741 (± 0.025)	0.818 (±0.005)	**0.822 (± 0.005)**

constructed a dataset of proteins that have a reliably assigned GO annotation of molecular function or biological process according to UniProtKB/SwissProt, as well as at least one reference to a PubMed abstract associated with their UniProtKB entry (see [61] for a more detailed description of the dataset). For each protein, we retrieved the PubMed abstracts referenced from its respective UniProtKB entry as a source of text features. As we aim to identify text-features that well-characterize each GO category corresponding to a function or a process, proteins annotated with three or more GO categories from the second level of the GO hierarchy are excluded. Furthermore, similar to the MultiLoc dataset of localized proteins, to ensure high-certainty functional annotation of the proteins, we excluded from the dataset proteins for which the evidence code associated with the GO annotation indicates reliance on computational methods (such as ISS: Inferred from Sequence/Structural similarity, ISA: Inferred from Sequence Alignment, etc. For a complete list of included and excluded evidence codes see [61], Table 1 therein).

Out of about 267,500 proteins – annotated with 4547 Molecular Function GO terms in UniProtKB/SwissProt, only 22,958 proteins are reliably annotated, leaving only 2859 of the Molecular Function GO terms reliably assigned; about 650 proteins annotated by three or more different functions or having no associated abstract are removed from the set, leaving 22,309 proteins tagged by 2641 GO function terms. Similarly, out of about 266,800 proteins annotated with 7204 Biological Process GO terms, only 23,919 are reliably annotated, and accordingly only 6733 of the associated GO terms are reliably assigned; of these proteins, about 1200 annotated by three or more different processes are removed from the set, leaving 21,764 proteins tagged by 4474 GO process terms.

The dataset thus consists of 36,536 proteins in total, of which 22,309 are reliably annotated by molecular function GO categories and 21,764 are reliably annotated by biological process categories. Representing proteins using text-features requires harvesting for each protein the PubMed abstracts associated with its UniProtKB entry. Recall that we aim to obtain terms that are highly predictive of potential function in order to represent proteins. As a single abstract may be referenced by multiple entries – possibly corresponding to proteins of different functions – in UniProtKB, abstracts associated with more than three proteins that have different functions are excluded from the set. The resulting text corpus holds a total of 68,337 abstracts, covering all the proteins in the dataset. This dataset was used to train and test our text-based classifiers through five complete studies of 5-fold cross-validation (as also described in Section 3.1). As was the case with the protein subcellular location experiments, to ensure robustness and statistical significance of the results, each study uses a different random partition into 5 disjoint subsets, thus resulting in a total of 25 train/test runs altogether.

As described in Section 2.2, we identified terms that are characteristic for each functional category. We then use the union of such terms from all functional classes to represent proteins and classify them. As explained in Section 3.1.1 above, the actual term selection step is done only *after* the set of proteins is partitioned into training and test sets; characteristic terms per function are selected based only on abstracts associated with proteins within the *training* set – never based on the test set. This process results in the selection of 521 *characteristic terms* for representing proteins in the context of *molecular function* classification and a total of 831 terms for protein representation in the context of *biological process*.

In the experiments we have conducted, our system performance was compared against two baseline classifiers. One of them, *Base-Prior*, simply derives the protein class label from the prior distribution of classes in the training set, through Monte Carlo sampling. That is, if the training dataset has 30% of its proteins annotated with the function '*transporter activity*' while 20% are annotated with '*enzyme regulator activity*', then to assign a label to protein p, Monte Carlo sampling is conducted from a distribution in which the label '*transporter activity*' has a 30% chance to be drawn and the label '*enzyme regulator activity*' has a 20% chance to be drawn (with other labels have their own chance to be drawn, for a total of 100%); the drawn label is then assigned to p.

The other baseline classifier, *Base-Seq*, is a k-nearest-neighbor classifier that uses *sequence similarity* as its classification criterion. To assign a function class to a protein p, it uses BLAST (with default parameters) to search for k proteins with a similar sequence to p's within the training set (in our experiments k was set to 10), and assigns p a function class that is shared by at least three of its k closest training proteins. Additional evaluation of a text-based classifier based on similar principles, has also taken place as part of the CAFA challenge and was discussed in that context, along with further details about the baseline classifiers [44,61].

As in the case of location-prediction, we have conducted additional preliminary experiments examining the ability of the function-prediction system to handle textless proteins, by representing proteins that had no associated abstracts listed in their UniProtKB entries (*textless*), using the homology-based strategy discussed in Section 2.3. As quite a few functional categories did not have textless proteins associated with them the experiment is limited in size.

The evaluation metrics employed are the same as those used in the location prediction task discussed earlier:

Table 5
Classification performance of the text-based classifiers *Text-SVM and Text-KNN*, over *molecular function* classes compared with *Base-Prior* and *Base-Seq*. The column *# of Proteins* shows the total number of proteins that are associated with each class in our dataset. The columns P, R, F, and M correspond to the classifier's Precision, Recall, *F*-measure, and MCC respectively, per class. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class. Highest values are shown in boldface.

Molecular function	# of Proteins	Text-SVM				Text-KNN				Base-Prior				Base-Seq			
		P	R	F	M	P	R	F	M	P	R	F	M	P	R	F	M
GO:0005488 Binding	13400	**0.70**	0.73	0.71	**0.23**	0.65	**0.88**	**0.75**	0.15	0.63	0.64	0.63	0.00	0.67	0.75	0.71	0.12
GO:0003824 Catalytic activity	3679	0.47	**0.46**	**0.46**	**0.36**	**0.52**	0.23	0.32	0.24	0.16	0.15	0.15	0.00	0.38	0.29	0.33	0.23
GO:0030528 Transcription regulator activity	1595	0.43	**0.53**	**0.47**	**0.43**	0.44	0.24	0.31	0.29	0.07	0.07	0.07	0.00	**0.49**	0.37	0.42	0.38
GO:0005215 Transporter activity	978	**0.50**	**0.55**	**0.52**	**0.50**	**0.59**	0.38	0.46	0.45	0.04	0.04	0.04	0.00	0.50	0.43	0.46	0.44
GO:0060089 Molecular transducer activity	922	**0**.38	**0.33**	**0.35**	**0.33**	**0.39**	0.16	0.22	0.25	0.04	0.04	0.04	0.00	0.26	0.27	0.27	0.23
GO:0030234 Enzyme regulator activity	606	0.33	**0.10**	**0.16**	**0.17**	**0.43**	0.05	0.08	0.15	0.03	0.03	0.03	0.01	0.16	0.09	0.12	0.11
GO:0005198 Structural molecular activity	418	**0.11**	0.01	0.02	0.03	0.04	0.01	0.01	0.00	0.02	0.02	0.02	0.00	**0.11**	**0.11**	**0.11**	0.09
GO:0016247 Channel regulator activity	72	0.50	0.06	0.11	0.18	**0.60**	**0.24**	**0.35**	**0.60**	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00
GO:0009055 Electron carrier activity	68	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0045182 Translator regulator activity	26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 6
Classification performance of the text-based classifiers *Text-SVM and Text-KNN*, over *biological process* classes compared with *Base-Prior* and *Base-Seq*. The column *# of Proteins* shows the total number of proteins that are associated with each class in our dataset. The columns P, R, F, and M correspond to the classifier's Precision, Recall, *F*-measure, and MCC respectively, per class. Precision and Recall values of 0 for a class indicate that all the proteins belonging to that class are misclassified into another class. Highest values are shown in boldface.

Biological Process	# of Proteins	Text-SVM				Text-KNN				Base-Prior				Base-Seq			
		P	R	F	M	P	R	F	M	P	R	F	M	P	R	F	M
GO:0065007 Biological regulation	4532	0.24	0.50	0.33	0.09	0.23	**0.52**	0.31	0.07	0.20	0.24	0.22	0.00	**0.32**	0.48	**0.38**	**0.15**
GO:0032502 Developmental process	4173	**0.27**	**0.35**	**0.31**	**0.21**	0.22	0.19	0.20	0.10	0.12	0.17	0.14	0.00	0.22	0.24	0.23	0.14
GO:0009987 Cellular process	2237	**0.26**	**0.43**	**0.33**	**0.16**	0.24	0.29	0.26	0.10	0.17	0.14	0.15	0.00	**0.26**	0.27	0.27	0.12
GO:0050896 Response to stimulus	2225	**0.31**	**0.22**	**0.26**	**0.19**	**0.25**	**0.16**	**0.19**	**0.12**	0.10	0.10	0.10	0.00	0.16	0.09	0.11	0.04
GO:0008152 Metabolic process	2073	**0.37**	0.07	0.11	0.13	0.23	0.14	0.17	0.13	0.08	0.06	0.07	0.00	0.28	**0.34**	**0.31**	**0.24**
GO:0051234 Establishment of localization	1505	0.39	0.26	0.31	0.29	0.32	0.20	0.25	0.21	0.05	0.05	0.05	0.00	**0.44**	**0.45**	**0.45**	**0.40**
GO:0016043 Cellular Component organization	1431	**0.17**	0.00	0.01	0.02	0.13	0.05	0.07	0.04	0.06	0.05	0.06	0.00	0.15	**0.12**	**0.13**	**0.09**
GO:0023052 Signaling	1206	0.22	0.11	0.15	0.13	0.18	0.11	0.14	0.13	0.05	0.04	0.04	0.00	**0.30**	**0.28**	**0.29**	**0.24**
GO:0032501 Multi-cellular organismal process	757	0.17	0.00	0.01	0.02	0.12	0.02	0.04	0.03	0.04	0.03	0.04	0.00	**0.24**	**0.11**	**0.16**	**0.15**
GO:0022414 Reproductive process	432	**0.57**	0.09	0.15	0.22	0.51	**0.15**	**0.24**	**0.34**	0.02	0.02	0.02	0.00	0.14	0.03	0.05	0.06
GO:0051704 muLti-organism process	340	0.20	0.01	0.01	0.04	**0.29**	**0.09**	**0.14**	**0.17**	0.01	0.01	0.01	0.00	0.09	0.04	0.05	0.05
GO:0040011 Locomotion	212	0.00	0.00	0.00	0.00	0.13	0.01	0.01	0.00	0.01	0.01	0.01	0.00	**0.84**	**0.05**	**0.09**	**0.28**
GO:0040007 Growth	206	0.00	0.00	0.00	0.00	**0.01**	**0.01**	**0.01**	0.00	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00
GO:0051179 Localization	189	0.00	0.00	0.00	0.00	**0.03**	**0.01**	**0.01**	0.00	0.01	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00
GO:0022610 Biological adhesion	160	**0.08**	0.01	0.02	**0.03**	0.07	**0.02**	**0.03**	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00
GO:0008283 Cell proliferation	147	0.00	0.00	0.00	0.00	**0.01**	**0.01**	**0.01**	0.00	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00
GO:0000003 Reproduction	120	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00
GO:0002376 Immune system response	93	**0.25**	**0.04**	**0.06**	**0.09**	0.06	0.03	0.04	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0016265 Death	80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00
GO:0071554 Cell wall organization	57	0.25	0.03	0.05	0.09	**0.38**	**0.08**	**0.13**	**0.21**	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0048511 Rhythmic process	54	0.00	0.00	0.00	0.00	**0.31**	**0.06**	**0.10**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0023046 Signaling process	44	0.00	0.00	0.00	0.00	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0044085 Cellular component biogenesis	20	0.00	0.00	0.00	0.00	**0.01**	**0.01**	**0.01**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GO:0043473 Pigmentation	16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

$$Recall = \frac{TP}{TP+FN},\quad Precision = \frac{TP}{TP+FP},\quad \text{and}\quad MCC = \frac{TP\cdot TN - FP\cdot FN}{\sqrt{(TP+FN)\cdot(TP+FP)\cdot(TN+FN)\cdot(TN+FP)}},$$

We also report the standard *F-measure*, which is the harmonic average of precision and recall [52] defined as: $F = \frac{2\cdot Recall\cdot Precision}{Recall + Precision}$.

3.2.2. Results for protein function/process prediction

Tables 5 and 6 show the results obtained from running the text-based SVM classifier over the protein datasets described in Section 3.2.1, along with results obtained from the baseline classifiers and those we have reported before obtained from the text-KNN classifier.

Table 7 shows results from classification of *textless proteins* that are annotated with *molecular function* classes, where the proteins were represented using text associated with their homologs (see Section 2.3). We note that some of the evaluated classes have a very small sample size of fewer than 10 textless proteins available and therefore, the evaluation results for these classes are not statistically meaningful. We only consider here classes with more than 2

Table 7
Performance of the SVM text-based classifier over *molecular function* classes, for proteins that have no associated text (textless). The columns P, R, and F show the classifier's Precision, Recall and F-measure, respectively, over individual classes that have at least 5 textless proteins. The rightmost part of the table shows for comparison the cross-validation results obtained for the same classes when testing/training over proteins that do have text (as shown in Table 5).

Molecular function	# Textless proteins	Text-SVM (Textless)			Text-SVM (Cross-validation)		
		P	R	F	P	R	F
GO:0005488 Binding	58	**0.88**	**0.76**	**0.81**	**0.70**	0.73	0.71
GO:0003824 Catalytic activity	9	0.25	0.44	0.32	0.47	**0.46**	**0.46**
GO:0005215 Transporter activity	5	**0.50**	0.20	0.29	**0.50**	**0.55**	**0.52**
GO:0060089 Molecular transducer activity	7	**0.38**	**0.43**	**0.40**	**0.38**	0.33	0.35

textless proteins. We also include, for comparison, the performance obtained for these classes through cross-validation for proteins that *do have* associated text (taken from Table 5). The results for the evaluated classes over textless proteins show Precision and Recall values that are consistent with those presented in Table 5, which were obtained using proteins that do have associated text. Similar results were obtained for biological process classes (not shown).

4. Discussion

Throughout this section we discuss the performance of our text-features-based classification/prediction approach. We start by discussing it in the context of protein subcellular location prediction (Section 4.1), then examine protein function prediction (Section 4.2), and finally, in Section 4.3, the applicability to *textless* proteins.

4.1. Protein subcellular location prediction using text-features

The results obtained by our location-prediction system clearly demonstrate the utility of our approach. In the first set of experiments discussed in Section 3.1, the text representation of proteins have shown to provide an effective means for characterizing protein location. The results shown in Table 3 (as well as the more detailed results not included here, see [2]) show that the text-based classifier, EpiLoc, performs at a level that is at least similar to (and often better than) that reported for sequence-based classifiers.

Compared to TargetP (Table 3, top) EpiLoc's *overall accuracy* and *average sensitivity* are slightly higher. On the PLOC dataset (Table 3, middle) PLOC's overall accuracy is higher than EpiLoc's, while EpiLoc's average sensitivity is much higher than PLOC's. The reader is referred to [2] for a more detailed location-specific comparative evaluation.

A more refined evaluation is included here in Table 4 for the MultiLoc dataset, which also includes the evaluation of HomoLoc for handling textless proteins, as further discussed in Section 4.3. EpiLoc's performance in terms of overall accuracy, average sensitivity, and almost all location-specific scores is *higher* compared to that of MultiLoc. In most cases the differences are statistically significant. The improvement is particularly noteworthy and significant *for subcellular locations with small datasets* such as the Golgi (*Go*), the Peroxisome (*Pe*) and the Lysosome (*Ly*), each with fewer than 150 proteins in the datasets. It appears that having the text as an extra data source in these cases is particularly beneficial.

4.2. Function prediction using text-features

The results presented in Tables 5 and 6 reflect preliminary experiments in which text is used as a feature-source for representation and functional classification of proteins. They show that the text-based classifiers (based either on SVM or on KNN) perform significantly better ($p < 0.05$, 2-sample t-test) than the simple baseline classifier, *Base-Prior*, which uses the class-distribution in the training set to guide its class assignment. An exception is the *molecular function* class '*structural molecular activity*' (GO:0005198, Table 5). The poor classification performance for this class can be explained by the fact that out of 418 proteins in this class, 218 are annotated as both '*structural molecular activity*' and '*binding*'. Thus, proteins associated with *'structural molecular activity'* have a similar weighted term-vector representation to proteins in the '*binding*' class. As Table 7 shows, proteins in the *'binding'* class outnumber those labeled as *'structural molecular activity'* by a factor of about 30. Hence, both the KNN and the SVM text-classifiers are likely to label *'structural molecular activity'* proteins as *'binding'*. Consequently, most of the proteins belonging to the *'structural molecular activity'* class are only classified as *'binding'*, resulting in a lower *Precision* and even lower *Recall* for the '*structural molecular activity'* class.

As for the *Base-Seq* classifier, which uses sequence similarity, the text-based classifiers have statistically significantly higher precision ($p < 0.05$) for half of the *molecular function* classes (Table 5), and the SVM classifier also shows higher *Recall*, *F-measure*, *MCC* for the majority of the classes. The three classes at the bottom of Table 5 each has fewer than 100 associated proteins; we note that both text-based classifiers make correct predictions for only one of these three classes, namely, *'channel regulator activity'* (GO:0016247), where the KNN classifier performs much better than the SVM. However, the *Base-Seq* classifier makes *no correct predictions* for *any* of the proteins in these classes. All classifiers perform poorly on these smallest classes, misclassifying most of the test proteins in these classes by assigning them to the majority class, *'binding'*. For assigning proteins to *biological process* classes (Table 6), the text-based SVM classifier (*Text-SVM)* has an overall accuracy of 0.26, similar to that of the *Base-Seq* classifier, which is 0.28. When considering larger classes (12 top classes out of 24, with *over 210* proteins), for 7 of them the *Base-Seq* has the highest values according to all performance measures, while for the others – one (or both) of the text-based classifiers performs better. For the vast majority of these larger classes, the SVM text classifier outperforms the KNN classifier.

Focusing on the smaller *biological process* classes, i.e., those with *fewer than 210* associated proteins, *Base-Seq* does not make any correct predictions, as it mis-assigns these proteins to the larger classes. In contrast, the text-based classifiers correctly assign process classes to at least some of these proteins, specifically in the classes *'immune system response'* (GO:0002376), *'cell wall organization or biogenesis'* (GO:0071554), and *'rhythmic process'* (GO:0048511).

Typically, *all classifiers* perform poorly over most of the *small classes* (GO classes with fewer than 100 proteins). As we have observed and discussed in detail in the context of CAFA [61], and briefly summarize here, the explanation for the phenomenon of poor text-based classification is that small classes often have a relatively low number of associated abstracts. This leads to a skewed term-distribution within the small abstract-set when compared to

the rest of the database, and, in turn, to a relatively large set of terms whose occurrence rate within the small abstract set is statistically significantly different from their occurrence rate in the rest of the dataset. Such a set of terms often includes generic concepts, like *human, male* or *library*. When such common terms are used in the representation, proteins from various classes end up having high weights in the common-term positions within their representing vector, which leads to misclassification. As part of our future work we shall reconsider and adjust the weighting scheme, which we believe will help address this issue and improve performance.

Another significant point is that the performance of all the classifiers when applied to *molecular function* is much better than when applied to *biological process* classification. This can be explained by the fact that proteins can take part in a *broad range of different biological processes*, while sharing similar molecular function and chemical properties. As such, while the classifiers can identify proteins that are likely to share a similar behavior (either based on sequence or based on similar associated text), and consequently may even share a similar molecular function, the biological processes of these proteins may still vary, and will not be correctly deduced based on any of the similarity measures used by our classifiers. Better performance is observed for *molecular function* classes because such functions are quite specific, and proteins sharing a common *molecular function* more often share other characteristics (both in terms of sequence similarity and in terms of language used to describe them). See [61] for further details concerning this point.

4.3. Handling textless proteins

We have presented the issue of proteins that may not have associated curated PubMed articles, or who may have generic associated articles, and as such are rendered "*textless*". One of the strategies we have devised and used for handling such textless proteins, as described in Section 2.3, is to identify their closest homologs that do have associated text, and use a weighted average of the homologs' text-based vector representation in order to represent the textless proteins.

It is important to note that this approach is significantly different from that of annotating proteins by "borrowing" the annotation of their closest homologs [10,34,60]. In contrast to such homology-based methods, we *do not* deduce the annotation of our target protein from the annotation of its homologs. Rather, we use the homologs just *in order to obtain a possible expected text-based representation* for the protein; this representation is then used as an input to a classifier that categorizes the protein. This representation can be extended to include the protein's own sequence characteristics (see [4] for an example).

Tables 4 and 7 both clearly demonstrate the value of this approach. As shown in Table 4, HomoLoc improves on EpiLoc's performance in terms of overall accuracy, and matches it in terms of average sensitivity. Homoloc also shows the highest performance for many of the individual locations. This improvement in performance is most likely due to the large amount of text that HomoLoc associates with each protein. Utilizing the additional abstracts that are associated with three close homologs – as opposed to just those abstracts referenced from the protein's own SwissProt entry – gives rise to a larger set of potentially significant terms for representing each protein, leading to a more robust representation and classification.

Table 7 shows results obtained when predicting protein function for textless proteins, using their homologs in order to represent them. As the number of textless proteins in the dataset was limited, the results are mostly illustrative, and are only shown for classes that had more than 2 textless proteins. Notably, the *only function class that has a substantial number of textless proteins* is the *binding* function (GO:0005488; 58 textless proteins). We observe that for this class the *Precision, Recall* and *F-measure* are much higher than expected from the cross-validation averages for these classes.

As for the remaining classes, each having fewer than 10 textless proteins: The *molecular transducer activity* (GO:0060089; 7 textless proteins), shows the same precision as expected from the cross-validation averages, along with significantly higher Recall and F-measure; The *catalytic activity* class, (GO:003824; 9 texless proteins) shows lower Precision but about the same Recall as expected from the cross-validation, while the *transporter activity* class, (GO:0005215; 5 texless proteins) shows the opposite trend with lower Recall and about the same precision as expected from the cross-validation average.

The results obtained from HomoLoc (Table 4), which are statistically significant as they were obtained on thousands of proteins, as well as the results from the *binding* class with more than 50 textless proteins in Table 7, provide strong evidence that in the absence of curated text for a protein, reliable prediction can still be performed by representing the protein through the text of its homologs. We have devised additional methods that can handle textless proteins, particularly proteins for which there are no close homologs and about which very little is known. The details of these methods are beyond the scope of this paper, and they are discussed further in an earlier publication [2].

5. Conclusions

We have presented and discussed a method, which we have developed over the past few years, for utilizing the biomedical literature pertaining to proteins as a source of data that can be used for representing proteins. Essentially, we use distributional properties of terms in the text in order to identify certain terms as informative about protein characteristics, and associate these terms with sets of proteins bearing these characteristics. This approach is significantly different from much of the work done in biomedical text mining, which typically focuses on automated *extraction* of facts that are *already stated* in the text. We have also discussed an effective method that we have devised, based on homology, to assign a relevant text-based representation to proteins that may not have published literature associated with them.

Several aspects of the approach and its value are demonstrated through two applications with which we have experimented, namely, protein location prediction and protein function prediction. The location prediction system, EpiLoc, has been shown to predict the subcellular location of proteins as reliably as other state-of-the-art systems. Moreover, it has been extensively tested and also integrated with a sequence-based classifier resulting in a highly accurate system for location prediction [4,54].

The function prediction system is still in its preliminary stages. An early KNN classifier was developed as a first attempt to use a text-based classification method as part of the CAFA challenge for automated function prediction [44]; the results of using that classifier on the CAFA challenge data are reported elsewhere [61].

Here we introduced the use of an SVM classifier for text-based function prediction, which typically outperformed the KNN in the 5-fold cross-validation runs. Both text-based classifiers performed significantly better than a simple baseline classifier that was based on class-distribution. They also showed comparable, or even higher, *Precision* and *Recall* compared to another baseline classifier that uses sequence-similarity, for several *molecular function* and *biological process* classes. These results suggest that text features extracted from the biomedical literature contain information about protein function beyond that evident in features obtained from

protein sequences alone. As such, integrating text-based features with sequence-based – as well as with other types of features, will likely improve the performance of existing function prediction systems.

We have pointed out that the performance of the function classifiers deteriorated for smaller classes, and noticed that in these cases the current feature selection identifies rather generic characteristic terms, due to the limited amount of associated text. Thus, an important direction for future work is the study of possible alternative statistics for selecting text features, which may prove more effective in such cases. We also note that the function prediction system was limited to categories at the second level of the GO hierarchy, due to the insufficient number of proteins associated with the deeper levels of GO. Once we improve the feature selection protocol to be effective even when there is a dearth of proteins and abstracts associated with a class, we shall extend the system by using categories at the lower levels of the GO hierarchy as function classes.

Both in the cross-validation studies shown here, and in the CAFA challenge, the prediction performance for molecular function was much higher than that obtained for biological process, which suggests that text features work better for predicting the *molecular function* of proteins than *biological process*. It is possible that improved selection of characteristic terms, discussed above, will also improve the performance over biological process classes.

We have focused throughout our presentation almost exclusively on text-based classification. Indeed, we view text as an important source of available information. However, we stress that we *do not* "advocate" for the replacement of sequence-based predictions by text-based prediction. On the contrary, we believe that *integrating text with other sources of data can improve prediction as a whole*. We have demonstrated this point in the integrative system SherLoc [54] and its successor SherLoc2 [4]. In the context of function prediction, our cross-validation results and the evaluation results from the CAFA challenge suggest that the information obtained from text features and that obtained from sequence features complement each other. Thus, the text-based and the sequence-based classifiers have different strengths and weaknesses. We therefore strongly believe that an integration of text- and sequence-based classifiers can advance function prediction, as well as other aspects of protein characterization.

Acknowledgements

This work was supported by HS's NSERC Discovery Award #298292-2009, NSERC Discovery Accelerator Award #380478-2009, CFI New Opportunities Award 10437, and Ontario's Early Researcher Award #ER07-04-085.

References

[1] M. Andrade, A. Valencia, Bioinformatics 14 (7) (1998) 600–607.
[2] S. Brady, H. Shatkay, in: Proc. of the Pacific Symposium on Biocomputing (PSB'08), 2008, pp. 604–615.
[3] S. Briesemeister, J. Rahnenfuhrer, O. Kohlbacher, Bioinformatics 26 (9) (2010) 1232–1238.
[4] S. Briesemeister, T. Blum, S. Brady, Y. Lam, O. Kohlbacher, H. Shatkay, J. Proteome Res. 8 (11) (2009) 5363–5366.
[5] R. Casadio, P.L. Martelli, A. Pierleoni, Brief. Funct. Genomic. Proteomic. 7 (1) (2008) 63–73.
[6] Cell Image from: <http://www.nobelprize.org/nobel_prizes/medicine/laureates/1999/press.html>.
[7] C.C. Chang, C.J. Lin. LIBSVM: A library for support vector machines. <http://www.csie.ntu.edu.tw/~clin/libsvm/>, 2003.
[8] J. Chiang, H. Yu, Bioinformatics 19 (11) (2003) 1417–1422.
[9] K. Chou, Z. Wu, X. Xiao, PLoS One 6 (3) (2011) 18258.
[10] H.N. Chua, W.K. Sung, L. Wong, Bioinformatics 22 (13) (2006) 1623–1630.
[11] A. Cohen, Brief. Bioinform. 6 (1) (2005) 57–71.
[12] A. Conesa, S. Götz, J.M. García-Góme, J. Terol, M. Talón, M. Robles, Bioinformatics 21 (18) (2005) 3674–3676.
[13] T. Cover, P. Hart, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[14] M. Craven, J. Kumlien, in: Proc. of the Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'99), 1999, pp. 77–86.
[15] R. Denroche, R. Madupu, S. Yooseph, G. Sutton, H. Shatkay, In: Linking Literature, Information, and Knowledge for Biology, Lecture Notes in Computer Science, vol. 6004, Springer, 2010. pp. 33–42.
[16] M.A. van Driel, J. Bruggeman, G. Vriend, H.G. Brunner, J.A.M. Leunissen, Eur. J. Hum. Genet. 24 (2006) 535–542.
[17] F. Eisenhaber, P. Bork, Bioinformatics 15 (1999) 525–535.
[18] O. Emanuelsson, H. Nielsen, S. Brunak, G. von Heijne, J. Mol. Biol. 300 (2000) 1005–1016.
[19] E. Eskin, E. Agichtein, in: Proc. of the Pacific Symposium on Biocomputing (PSB'04), 2004, pp. 288–299.
[20] I. Friedberg, Brief. Bioinform. 7b (2006) 225–242.
[21] P. Groth, B. Weiss, H.D. Pohlenz, U. Leser, BMC Bioinformatics 9 (2008) 136.
[22] P. Groth, B. Weiss, H.D. Pohlenz, U. Leser, in: Silico Tools for Gene Discovery, Methods in Molecular Biology, vol. 760, Springer, 2011, pp. 159–173 (Chapter 10).
[23] P. Horton, K. Park, T. Obayashi, N. Fujita, H. Harada, C. Adams-Collier, K. Nakai, Nucleic Acids Res. 35 (Web Server issue) (2007) 585–587.
[24] A. Höglund, P. Dönnes, T. Blum, H.W. Adolph, O. Kohlbacher, Bioinformatics 22 (10) (2006) 1158–1165.
[25] L. Jensen, J. Saric, P. Bork, Nat. Rev. Genet. 7 (2006) 119–129.
[26] A. Koike, Y. Niwa, T. Takagi, Bioinformatics 21 (7) (2005) 1227–1236.
[27] Y.P. Lam, Comparing Naïve Bayes classifiers with Support Vector Machines for predicting protein subcellular location using text features (M.Sc. thesis), Queen's University, School of Computing, <https://qspace.library.queensu.ca/handle/1974/5920>, 2010.
[28] Z. Lu, D. Szafron, R. Greiner, P. Lu, D.S. Wishart, B. Poulin, J. Anvik, C. Macdonell, R. Eisner, Bioinformatics 20 (4) (2004) 547–556.
[29] Z. Lu, H.Y. Kao, C.H. Wei, et al., BMC Bioinformatics 12 (Suppl. 8) (2011) S9.
[30] D.M. Martin, M. Berriman, G.J. Barton, BMC Bioinformatics 5 (2004) 178.
[31] B.W. Matthews, Biochim. Biophys. Acta 405 (1975) 442–451.
[32] T.M. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.
[33] S. Mostafavi, R. Debajyoti, D. Warde-Farley, C. Grouious, Q. Morris, Genome Biol. 9 (Suppl. 1) (2008) S4.
[34] R. Nair, B. Rost, Protein Sci. 11 (2002) 2836–2847.
[35] R. Nair, B. Rost, Bioinformatics 18 (2002) S78–S86.
[36] G. Nenadic, S. Rice, I. Spasic, S. Ananiadou, B. Stapley, in: Proc. of the ACL Workshop on Natural Language Processing in Biomedicine, vol. 13, 2003, pp. 121–128.
[37] H. Pan, L. Zuo, V. Choudhary, Z. Zhang, S.H. Leow, F.T. Chong, Y. Huang, V.W.S. Ong, B. Mohanty, S.L. Tan, S.P.T. Krishnan, V. Bajic, Nucleic Acids Res. 32 (2008) 230–234.
[38] K.J. Park, M. Kanehisa, Bioinformatics 19 (2003) 1656–1663.
[39] F. Pazos, M. Sternberg, Proc. Natl. Acad. Sci. USA 101 (41) (2004) 14754–14759.
[40] A.J. Pérez, C. Perez-Iratxeta, P. Bork, G. Thode, M.A. Andrade, Bioinformatics 20 (13) (2004) 2084–2091.
[41] M. Porter, Program Electron. Libr. Inf. Syst. 40 (3) (2006) 211–218.
[42] The Protein Data Bank, <http://www.wwpdb.org/index.html>.
[43] Pubmed, NCBI/NLM/NIH, <http://www.ncbi.nlm.nih.gov/pubmed>.
[44] P. Radivojac, W.T. Clark, T.R. Oron, et al., Nat. Methods 10 (2013) 221–227.
[45] S. Raychaudhuri, J. Chang, P. Sutphin, R. Altman, Genome Res. 12 (2002) 203–214.
[46] R. Rentzsch, C. Orengo, Trends Biotechnol. 27 (4) (2009) 210–219.
[47] C.J. van Rijsbergen, J. Doc. 33 (2) (1977) 106–119.
[48] C.J. van Rijsbergen, Information Retrieval, Butterworth, London, 1979.
[49] G. Salton, Automatic Text Processing, Addison-Wesley, 1989.
[50] F. Sebastiani, ACM Comput. Surv. 34 (1999) 1–47.
[51] P.K. Shah, C. Perez-Iratxeta, P. Bork, M.A. Andrade, BMC Bioinformatics 4 (2003) 20.
[52] H. Shatkay, M. Craven, Mining the Biomedical Literature, MIT Press, 2012.
[53] H. Shatkay, S. Edwards, W.J. Wilbur, M. Boguski, in: Proc. of the Int. Conf. on Intelligent Systems for Molecular Biology (ISMB'2000), 2000, pp. 317–328.
[54] H. Shatkay, A. Höglund, S. Brady, T. Blum, P. Dönnes, O. Kohlbacher, Bioinformatics 23 (11) (2007) 1410–1417.
[55] H. Simha, H. Shatkay, Algorithms Mol. Biol. 9 (1) (2014) 8.
[56] A. Sokolov, C. Funk, K. Graim, K. Verspoor, A. Ben-Hur, BMC Bioinformatics 14 (Suppl. 3) (2013) S10.
[57] B.J. Stapley, L.A. Kelley, M.J.E. Sternberg, in: Proc. of the Pacific Symposium on Biocomputing (PSB), 2002, pp. 374–385.
[58] T. Theodosiou, L. Angelis, A. Vakali, G.N. Thomopoulos, Int. J. Med. Informatics 76 (8) (2007) 601–613.
[59] The UniProt Consortium, Nucleic Acids Res. 42 (2014) D191–D198.
[60] M. Wass, M. Sternberg, Bioinformatics 24 (6) (2008) 798–806.
[61] A. Wong, H. Shatkay, BMC Bioinformatics 14 (Suppl. 3) (2013) S14.
[62] Y. Yang, J.O. Pedersen, in: Proc. of the 14th International Conference on Machine Learning (ICML), 1997, pp. 412–420.
[63] G. Zehetner, Nucleic Acids Res. 31 (13) (2003) 3799–3803.